

Manual do usuário

Amazon CodeCatalyst



Amazon CodeCatalyst: Manual do usuário

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

As marcas comerciais e imagens de marcas 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

.....	xxiii
O que é o Amazon CodeCatalyst?	1
O que posso fazer com o CodeCatalyst?	1
Como posso começar a usar o CodeCatalyst?	2
Saiba mais sobre o CodeCatalyst	2
Como migrar do CodeCatalyst	3
Migrar seus repositórios	3
Migrando seu CodeCatalyst repositório para um repositório GitLab	3
Migrando seu CodeCatalyst repositório para um repositório GitHub	3
Migração genérica para outro provedor de repositório	3
Extraindo seus dados de CodeCatalyst	5
Baixar artefatos	5
Baixar os anexos do problema	6
Baixar o código-fonte de uma ação	6
Excluindo seus dados do CodeCatalyst	7
Solicite que a equipe de serviço exclua dados em seu nome	7
Exclua seu CodeCatalyst espaço	7
Excluir um projeto	8
Excluir um repositório de origem	9
Excluir um esquema personalizado	10
Excluir os anexos do problema	10
Excluindo um arquivo em ambientes de desenvolvimento acessados pela Amazon CodeCatalyst	11
Excluir um Ambiente de Desenvolvimento para o espaço	11
Excluir conexões de conta	11
Remover uma conta de um CodeCatalyst espaço no AWS Management Console	12
Excluir um segredo	13
Exclusão de uma equipe	14
Exclusão de uma frota provisionada	15
Exclusão de um repositório de pacotes	15
Conceitos	17
AWS Espaços de ID do construtor em CodeCatalyst	18
Espaços que oferecem suporte à federação de identidades em CodeCatalyst	18
Projetos	18

Esquemas	18
Conexões de conta	19
Conexões VPC	19
AWS ID do construtor	20
Perfis de usuário em CodeCatalyst	20
Repositórios de origem	20
Confirmações	21
Ambientes de Desenvolvimento	22
Fluxos de trabalho	22
Ações	22
Problemas	23
Tokens de acesso pessoal (PATs)	23
Conexões pessoais	23
Perfis	24
Configuração e login no CodeCatalyst	25
Criar um espaço e perfil de desenvolvimento (começando sem um convite)	27
Criar um espaço e perfis do IAM	28
Aceitando um convite e criando um AWS Builder ID	34
Aceitando um convite e criando um AWS Builder ID	35
Login com um ID do builder AWS	36
Dispositivos confiáveis	37
Fazer login com SSO	37
Visualização de todos os espaços e projetos de um usuário	37
Visualização e gerenciamento de perfis do CodeCatalyst	38
Visualização dos seus perfis do CodeCatalyst	39
Visualização do perfil do CodeCatalyst de outro usuário	39
Atualização de um perfil	40
Alteração de uma senha do CodeCatalyst associada a um ID do builder AWS	41
Configurando para usar o AWS CLI com CodeCatalyst	42
Tutoriais de conceitos básicos	44
Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis	45
Pré-requisitos	47
Etapa 1: criar o projeto de aplicação web moderna de três níveis	48
Etapa 2: convidar alguém para o projeto	49
Etapa 3: criar problemas para colaborar e monitorar o trabalho	50
Etapa 4: visualizar o repositório de origem	51

Etapa 5: criar um Ambiente de Desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código	52
Etapa 6: visualizar o fluxo de trabalho que cria a aplicação moderna	54
Etapa 7: pedir que outras pessoas revisem suas alterações	57
Etapa 8: encerrar o problema	61
Limpar recursos.	61
Referência	62
Tutorial: começo com um projeto vazio	64
Pré-requisitos	64
Criar um projeto vazio	65
Criar um repositório de origem	65
Criar um fluxo de trabalho para criar, testar e implantar uma alteração de código	67
Convidar alguém para o projeto	67
Criar problemas para colaborar e monitorar o trabalho	68
Tutorial: Usar os recursos de IA generativa	69
Pré-requisitos	70
Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade	71
Adicionar um resumo gerado automaticamente ao criar uma solicitação pull	75
Criar um resumo dos comentários deixados sobre alterações de código em uma solicitação pull	78
Criar um problema e atribuir ao Amazon Q	79
Criar um problema e ter tarefas recomendadas pelo Amazon Q para ele	87
Limpar os recursos	88
Organização de recursos com espaços	89
Criar um espaço	91
Editar um espaço	94
Excluir um espaço	94
Monitoramento de atividades para usuários e recursos em um espaço	96
Permitindo acesso a AWS recursos com conexão Contas da AWS	96
Adicionando um Conta da AWS a um espaço	97
Adicionar perfis do IAM às conexões da conta	101
Adição da conexão da conta e dos perfis do IAM ao seu ambiente de implantação	103
Visualização de conexões de conta	104
Excluindo conexões de conta (em CodeCatalyst)	104
Configurar uma conta de faturamento para um espaço	106

Configuração de perfis do IAM para contas conectadas	106
Perfil do CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i>	107
Perfil do AWSRoleForCodeCatalystSupport	108
Criação de uma função do IAM e uso da política de CodeCatalyst confiança	109
Concessão de permissões de espaço aos usuários	110
Visualização dos membros em um espaço	111
Convite de um usuário diretamente para um espaço	112
Cancelamento de um convite para um espaço	113
Alteração do perfil de um membro do espaço	114
Remoção de um membro do espaço	115
Remoção ou alteração do perfil de um usuário com o perfil de administrador do espaço	116
Permitir o acesso ao espaço usando equipes	117
Criação de uma equipe	118
Visualizar uma equipe	120
Conceder perfis de espaço para uma equipe	121
Concessão de perfis de projeto para uma equipe no nível do espaço	121
Adição de um usuário diretamente a uma equipe	122
Remover um usuário de uma equipe diretamente	123
Adição de um grupo de SSO a uma equipe	124
Exclusão de uma equipe	124
Permissão do acesso ao espaço para recursos da máquina	125
Visualização do acesso ao espaço para recursos da máquina	126
Desativação do acesso ao espaço para recursos da máquina	126
Ativação do acesso ao espaço para recursos da máquina	127
Administração de Ambientes de Desenvolvimento para um espaço	128
Visualização de Ambientes de Desenvolvimento para seu espaço	129
Editar um Ambiente de Desenvolvimento para o espaço	130
Interromper um Ambiente de Desenvolvimento para o espaço	130
Excluir um Ambiente de Desenvolvimento para o espaço	131
Cotas para espaços	132
Organizar o trabalho com projetos	134
Criação de um projeto	135
Criação de um projeto vazio	135
Criar um projeto com um repositório de terceiros vinculado	136
Criar um projeto com um esquema	141

Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas	144
Práticas recomendadas para usar esquemas em projetos	146
Adição de recursos e tarefas aos projetos criados	146
Obtenção de uma lista de projetos	147
Visualização de tarefas do projeto e ambientes de desenvolvimento	148
Visualização de todos os projetos em um espaço	148
Visualização das configurações do projeto	149
Mudando para um projeto diferente em CodeCatalyst	149
Excluir um projeto	150
Concessão de permissões de projeto aos usuários	150
Obtenção de uma lista dos membros e perfis no projeto	150
Convite de um usuário para um projeto	152
Cancelamento de um convite	153
Remoção de um usuário do projeto	153
Aceitação ou recusa de um convite para um projeto	154
Permissão do acesso ao projeto usando equipes	155
Adição de uma equipe a um projeto	155
Concessão de perfis de projeto para uma equipe	156
Remoção de um perfil de projeto para uma equipe	156
Permissão do acesso ao projeto para recursos da máquina	157
Visualizar o acesso ao projeto para recursos da máquina	158
Desativação do acesso ao projeto para recursos da máquina	158
Ativação do acesso ao projeto para recursos da máquina	159
Cotas para projetos	160
Enviar notificações	160
Como as notificações funcionam?	161
Conceitos básicos das notificações do Slack	163
Envio de notificações do Slack e por e-mail	167
Configurar projetos com esquemas	172
Criar um projeto com um esquema	173
Adicionar um esquema a um projeto para integrar recursos	174
Desassociar um esquema de um projeto	177
Alteração das versões do esquema em um projeto	177
Edição de uma descrição para um esquema em um projeto	180
Trabalho com o gerenciamento do ciclo de vida como usuário do esquema	181

Usar o gerenciamento do ciclo de vida em projetos existentes	182
Usar o gerenciamento do ciclo de vida em vários esquemas em um projeto	182
Trabalho com conflitos em solicitações pull do ciclo de vida	182
Cancelamento das alterações no gerenciamento do ciclo de vida	182
Como ignorar o gerenciamento do ciclo de vida de um esquema em um projeto	183
Criação de um projeto abrangente com esquemas	184
Esquemas disponíveis	184
Descoberta de informações sobre o esquema do projeto	188
Padronização de projetos com esquemas personalizados	188
Conceitos de esquemas personalizados	189
Conceitos básicos de esquemas personalizados	193
Tutorial: Criar e atualizar uma aplicação React	199
Converter repositórios de origem em esquemas personalizados	206
Trabalho com o gerenciamento do ciclo de vida como autor do esquema	208
Desenvolvimento de um esquema personalizado para satisfazer os requisitos do projeto	215
Publicação de um esquema personalizado em um espaço	254
Definição de permissões de publicação para um esquema personalizado	258
Adicionar um esquema personalizado a um catálogo de esquemas de espaço	259
Alterar as versões do catálogo para um esquema personalizado	260
Visualização de detalhes, versões e projetos de um esquema personalizado	261
Remoção de um esquema personalizado de um catálogo de esquemas de espaço	262
Exclusão de um esquema ou uma versão personalizada publicada	263
Como adicionar dependências, lidar com incompatibilidades e atualizar ferramentas e componentes	265
Contribuir	268
Cotas para esquemas	268
Armazenamento e colaboração no código com repositórios de origem	269
Conceitos de repositório de origem	271
Projetos	18
Repositórios de origem	271
Ambientes de Desenvolvimento	22
Tokens de acesso pessoal (PATs)	23
Ramificações	273
Ramificações padrão	273
Confirmações	21
Solicitações pull	274

Revisões	274
Fluxos de trabalho	22
Configurar	275
Instale o Git	276
Criar um token de acesso pessoal	276
Conceitos básicos dos repositórios de origem	277
Criar um projeto com um esquema	277
Visualizar os repositórios de um projeto	280
Criar um Ambiente de Desenvolvimento	281
Criar uma solicitação pull	283
Mesclar uma solicitação pull	286
Visualizar o código implantado	287
Limpar os recursos	287
Armazenamento do código-fonte em repositórios	288
Criar um repositório de origem	289
Clonar um repositório Git existente em um repositório de origem	291
Vincular um repositório de origem	294
Visualizar um repositório de origem	297
Editar as configurações de um repositório de origem	299
Clonar um repositório de origem	300
Exclusão de um repositório de origem	302
Organizar o trabalho de código-fonte com ramificações	303
Criar uma ramificação	304
Gerenciar a ramificação padrão	305
Gerenciar as ações permitidas com regras de ramificação	307
Comandos do Git para ramificações	310
Visualização de ramificações e detalhes	311
Exclusão de uma ramificação	312
Gerenciar arquivos de código-fonte	313
Criar ou adicionar um arquivo	314
Visualizar um arquivo	316
Visualizar o histórico de alterações em um arquivo	317
Edição de um arquivo	318
Renomear ou excluir um arquivo	319
Revisar o código com solicitações pull	319
Criar uma solicitação pull	321

Visualizar solicitações pull	326
Gerenciamento de requisitos para mesclar com regras de aprovação	328
Revisão de uma solicitação pull	330
Atualizar uma solicitação pull	333
Mesclar uma solicitação pull	335
Encerrar uma solicitação pull	339
Noções básicas das alterações no código-fonte com confirmações	339
Visualização de confirmações em uma ramificação	340
Alterando a forma como os commits são exibidos (CodeCatalystconsole)	341
Cotas para repositórios de origem	342
Escrever e modificar código com ambientes de desenvolvimento	347
Criar um Ambiente de Desenvolvimento	348
Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento	349
Criando um ambiente de desenvolvimento em CodeCatalyst	350
Criação de um Ambiente de Desenvolvimento em um IDE	353
Interromper um Ambiente de Desenvolvimento	353
Retomar um Ambiente de Desenvolvimento	355
Edição de um Ambiente de Desenvolvimento	357
Excluir um Ambiente de Desenvolvimento	357
Estabelecer conexão com um Ambiente de Desenvolvimento usando SSH	358
Configuração e uso de um devfile	360
Edição de um devfile	362
Recursos do Devfile suportados pelo CodeCatalyst	363
Exemplo de um devfile para um Ambiente de Desenvolvimento	364
Solução de problemas de um repositório devfile usando o modo de recuperação	365
Especificar imagens de devfile universais	365
Comandos Devfile	370
Eventos do Devfile	371
Componentes do Devfile	372
Associar uma conexão VPC a um Ambiente de Desenvolvimento	372
Cotas de ambientes de desenvolvimento	374
Publicar e compartilhar pacotes de software	375
Conceitos dos pacotes	376
Pacotes	376
Namespaces de pacote	377

Versões do pacote	377
Ativos	377
Repositórios de pacotes	377
Repositórios upstream	378
Repositórios de gateway	378
Configuração e uso de repositórios de pacotes	379
Criação de um repositório de pacotes	379
Conexão a um repositório de pacotes	380
Exclusão de um repositório de pacotes	380
Configuração e uso de repositórios upstream	381
Adição de um repositório upstream	382
Edição da ordem de pesquisa dos repositórios upstream	383
Solicitar uma versão do pacote com repositórios upstream	383
Remoção de um repositório upstream	387
Conexão a repositórios externos públicos	387
Repositórios de pacotes externos compatíveis e repositórios de gateway	388
Publicação e modificação de pacotes	389
Publicação de pacotes	390
Visualização de detalhes da versão de pacote	391
Exclusão de uma versão de pacote	392
Atualização do status da versão do pacote	392
Editar controles de origem do pacote	394
Uso de npm	400
Configuração e uso de npm	400
Tratamento de tags npm	410
Usando o Maven	411
Configuração e uso do Gradle Groovy	412
Configuração e uso de mvn	421
Publicação de pacotes com curl	429
Usar somas de verificação e instantâneos do Maven	431
Usando NuGet	433
Usando CodeCatalyst com o Visual Studio	433
Configuração e uso de nuget ou dotnet	436
NuGet normalização do nome do pacote, versão e nome do ativo	439
NuGet compatibilidade	440
Usar o Python	441

Configurar pip e instalação de pacotes Python	441
Configuração do Twine e publicação de pacotes Python	445
Normalização do nome do pacote Python	447
Compatibilidade com Python	447
Cotas para pacotes	448
Compilação, teste e implantação com fluxos de trabalho	449
Sobre o arquivo de definição do fluxo de trabalho	449
Usando os editores visual e YAML do CodeCatalyst console	451
Descoberta de fluxos de trabalho	453
Visualização de detalhes da execução do fluxo de trabalho	454
Próximas etapas	455
Conceitos de fluxos de trabalho	455
Fluxos de trabalho	455
Arquivos de definição de fluxo de trabalho	456
Ações	456
Grupos de ações	456
Artefatos	456
Computação	457
Ambientes do	457
Portões	457
Relatórios	457
Execuções	458
Fontes	458
Variáveis	458
Gatilhos do fluxo de trabalho	458
Conceitos básicos de fluxos de trabalho	459
Pré-requisitos	460
Etapa 1: criar e configurar o fluxo de trabalho	460
Etapa 2: salvar o fluxo de trabalho com uma confirmação	462
Etapa 3: visualizar os resultados da execução	463
(Opcional) Etapa 4: limpar	463
Criação com fluxos de trabalho	463
Como faço para criar uma aplicação?	464
Benefícios da ação de criação	465
Alternativas à ação de criação	465
Adição da ação de criação	465

Visualização dos resultados da ação de criação	467
Tutorial: Fazer upload de artefatos no Amazon S3	468
YAML: ações de criação e de teste	477
Teste com fluxos de trabalho	506
Tipos de relatório de qualidade	506
Adição da ação de teste	509
Visualização dos resultados da ação de teste	511
Omissão dos testes com falha	511
Integrando com universal-test-runner	512
Configuração de relatórios de qualidade	514
Melhores práticas de teste	521
Propriedades SARIF	525
Implantar com fluxos de trabalho	534
Como faço para implantar uma aplicação?	534
Lista de ações de implantação	535
Benefícios das ações de implantação	536
Alternativas para ações de implantação	536
Implantação no Amazon ECS	537
Implantação no Amazon EKS	590
Implantando uma pilha CloudFormation	639
Implantação de um aplicativo AWS CDK	692
Inicializando um aplicativo AWS CDK	718
Publicação no Amazon S3	735
Implantação em e Contas da AWS VPCs	750
Exibir o URL da aplicação	763
Remoção de um destino de implantação	768
Rastreamento do status de implantação por confirmação	769
Visualização de logs de implantação	771
Visualizar informações de implantação	772
Criação de um fluxo de trabalho	773
Executar um fluxo de trabalho	777
Iniciar execuções manualmente	778
Início de execuções automáticas usando gatilhos	779
Configurar gatilhos somente manuais	796
Parada de uma execução de fluxo de trabalho	798
Bloquear uma execução de fluxo de trabalho	798

Solicitar aprovações em execuções de fluxo de trabalho	802
Configurar o comportamento de enfileiramento das execuções	815
Armazenar arquivos em cache entre execuções	822
Visualização do status e detalhes de execução do fluxo de trabalho	826
Configuração de ações de fluxo de trabalho	831
Tipos de ação	831
Adição de uma ação	835
Remover uma ação	837
Desenvolver uma ação personalizada	838
Agrupar ações em grupos de ações	839
Sequenciar ações	841
Compartilhar artefatos e arquivos entre ações	846
Especificação da versão da ação a ser usada	860
Lista das versões de ação disponíveis	862
Visualizar o código-fonte de uma ação	863
Integração com GitHub ações	865
Configuração de imagens de computação e runtime	904
Tipos de computação	905
Frotas de computação	905
Propriedades da frota sob demanda	906
Propriedades da frota provisionada	907
Criar uma frota provisionada	909
Edição de uma frota provisionada	910
Exclusão de uma frota provisionada	910
Atribuir uma frota ou computação a uma ação	911
Compartilhamento de computação entre ações	913
Especificação de imagens de ambiente de runtime	919
Conectar repositórios de origem aos fluxos de trabalho	929
Especificar o repositório de origem de um arquivo de fluxo de trabalho	929
Especificar o repositório de origem da ação do fluxo de trabalho	930
Fazer referência a arquivos do repositório de origem	932
Variáveis - 'BranchName' e 'CommitId'	933
Conectar repositórios de pacotes a fluxos de trabalho	934
Tutorial: extrair de um repositório de pacotes	934
Especificação de repositórios de CodeCatalyst pacotes em fluxos de trabalho	951
Usar tokens de autorização em ações de fluxo de trabalho	953

Exemplos: repositórios de pacotes em fluxos de trabalho	955
Invocar uma função do Lambda	957
Quando usar essa ação	958
Imagem de tempo de execução usada pela AWS Lambda ação 'invocar'	958
Exemplo: invocar uma função do Lambda	958
Adicionando a ação 'AWS Lambda invocar'	960
Variáveis - 'AWS Lambda invocar'	962
YAML - ação 'AWS Lambda invocar'	963
Modificação de uma definição de tarefa do Amazon ECS	978
Quando usar essa ação	978
Como a ação “Renderizar definição de tarefa do Amazon ECS” funciona	979
Imagem de runtime usada pela ação “Renderizar definição de tarefa do Amazon ECS”	981
Exemplo: modificar uma definição de tarefa do Amazon ECS	981
Adicionar a ação “Renderizar definição de tarefa do Amazon ECS”	984
Visualização do arquivo de definição de tarefa atualizado	986
Variáveis: “Renderizar definição de tarefa do Amazon ECS”	988
YAML: “Renderizar definição de tarefa do Amazon ECS”	988
Uso de variáveis em fluxos de trabalho	997
Usar variáveis definidas pelo usuário	998
Usar variáveis predefinidas	1011
Mascarar dados usando segredos	1016
Criar um segredo	1016
Editar um segredo	1017
Usar um segredo	1018
Excluir um segredo	1019
Visualização do status de um fluxo de trabalho	1020
Cotas para fluxo de trabalho	1021
Estados de execução do fluxo de trabalho	1023
Estados do fluxo de trabalho	1024
Definição do YAML do fluxo de trabalho	1025
Exemplo de um arquivo de definição de fluxo de trabalho	1026
Diretrizes e convenções de sintaxe	1027
Propriedades de nível superior	1029
Acompanhar e organizar o trabalho com problemas	1042
Conceitos de problemas	1043
Problemas ativos	1043

Problemas arquivados	1044
Destinatário	1044
Campos personalizados	1044
Estimativa	1044
Problema	1044
Rótulo	1045
Prioridade	1045
Status e categorias de status	1045
Tarefas	1045
Visualizações	1045
Acompanhamento do trabalho com problemas	1046
Criar um problema	1046
Estimativa de um problema	1051
Editar e colaborar em problemas	1052
Descoberta e visualização de problemas	1061
Progresso em um problema	1064
Arquivamento de um problema	1065
Exportação de problemas	1066
Organização do trabalho com backlogs, rótulos e quadros	1067
Classificação do trabalho com rótulos	1067
Organização do trabalho com campos personalizados	1068
Acompanhamento do trabalho com status personalizados	1069
Configuração da estimativa do esforço do problema	1071
Ativação ou desativação de vários responsáveis	1072
Criação de uma visualização de problemas	1072
Cotas para problemas	1073
Configure identidade, permissões e acesso no CodeCatalyst	1075
Concessão de acesso com perfis de usuário	1076
Noções básicas sobre perfis de usuário em espaços e projetos	1076
Visualização das permissões disponíveis para cada perfil	1079
Visualização e alteração de perfis do usuário	1123
Conceder aos usuários acesso ao repositório com tokens de acesso pessoal	1125
Criando PATs	1125
Visualizando PATs	1127
Excluindo PATs	1129
.....	1130

Criação de conexões pessoais	1131
Excluir conexões pessoais	1132
Configure seu AWS Builder ID para fazer login com autenticação multifator (MFA)	1133
Como registrar um dispositivo para uso com autenticação multifator	1134
Aplicativos autenticadores	1136
Alteração dos dispositivos de MFA	1137
Segurança	1138
Privacidade de dados	1139
Proteção de dados	1140
CodeCatalyst e Identity and Access Management	1142
Validação de compatibilidade	1197
Resiliência	1198
Segurança da infraestrutura	1198
Análise de configuração e vulnerabilidade	1198
Seus dados e privacidade no Amazon CodeCatalyst	1199
Melhores práticas para ações de fluxo de trabalho	1199
Entendendo o modelo de CodeCatalyst confiança	1200
Monitoramento de eventos e chamadas de API usando o registro	1203
Monitoramento de chamadas de API em Contas da AWS usando o registro do AWS	
CloudTrail	1206
Acesso a eventos de logs usando o registro de eventos	1214
Cotas para identidade, permissão e acesso	1217
Solução de problemas	1218
Problemas de inscrição	1218
Problemas de login	1219
Problemas ao sair	1220
Recebo um erro de perfil não existe para um fluxo de trabalho com falha	1220
Recebo um erro de perfil para um fluxo de trabalho com falha	1221
Preciso atualizar o perfil do IAM no fluxo de trabalho do projeto	1221
Tenho uma solicitação de revisão para minha GitHub conta depois de criar uma conexão pessoal	1222
Como preencho um formulário de suporte?	1222
Adicionar funcionalidade a projetos com extensões	1223
Extensões de terceiros disponíveis	1224
Integrando GitHub repositórios em CodeCatalyst	1224
Integrando repositórios do Bitbucket em CodeCatalyst	1225

Integrando GitLab repositórios em CodeCatalyst	1226
Integrando problemas do Jira no CodeCatalyst	1227
Conceitos de extensões	1227
Extensões	1227
CodeCatalyst catálogo	1228
Conexão e vínculo	1228
Início rápido: instalação de extensões, conexão de provedores e vínculo de recursos	1228
Etapa 1: instalar uma extensão de terceiros do CodeCatalyst catálogo	1231
Etapa 2: Conecte seu provedor terceirizado ao seu CodeCatalyst espaço	1232
Etapa 3: vincular seus recursos de terceiros ao seu CodeCatalyst projeto	1235
Próximas etapas	1239
Instalação de uma extensão em um espaço	1239
Desinstalação de uma extensão em um espaço	1240
Conectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira	1241
Desconectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira	1245
Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira	1247
Vínculo de recursos de provedores de terceiros conectados	1250
Vincular um repositório de terceiros durante a criação do projeto	1256
Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira	1256
Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst	1258
Visualizando repositórios de terceiros no CodeCatalyst	1258
Pesquisando problemas com o Jira no CodeCatalyst	1260
Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros	1260
Adição de gatilhos para iniciar a execução do fluxo de trabalho	1261
Restrição do acesso ao IP com provedores de repositório de terceiros	1262
Endereços IP usados pela extensão de repositórios de terceiros	1263
Bloqueio de mesclagens de terceiros quando os fluxos de trabalho falham	1263
Vincular problemas do Jira com solicitações pull	1264
Visualizando CodeCatalyst eventos em edições do Jira	1266
Pesquisa do código, problemas, projetos e usuários	1267
Refinamento da sua consulta de pesquisa	1268

Refinamento por tipo	1268
Refinamento por campo	1269
Refinamento com operadores booleanos	1269
Refinamento por projeto	1270
Considerações ao trabalhar com a pesquisa	1270
Referência de campos pesquisáveis	1271
Solução de problemas	1277
Solução de problemas de acesso gerais	1277
Esqueci minha senha	1277
Alguns ou todos os Amazon CodeCatalyst não estão disponíveis	1278
Não consigo criar um projeto no CodeCatalyst	1278
Solução de problemas de suporte	1278
Recebo uma mensagem de erro ao acessar o Suporte para o Amazon CodeCatalyst	1278
Não consigo criar casos de suporte técnico para meu espaço	1279
Minha conta para casos de suporte não está mais conectada ao meu espaço no CodeCatalyst	1279
Não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) no Suporte para o Amazon CodeCatalyst	1280
Alguns ou todos os Amazon CodeCatalyst não estão disponíveis	1278
Não consigo criar um projeto no CodeCatalyst	1278
Não consigo acessar meu espaço BID como um novo usuário ou não posso ser adicionado como um novo usuário de SSO devido ao nome de usuário truncado	1281
Adicionar um usuário do SSO ao meu espaço federado como um novo usuário criou um usuário duplicado	1282
Quero enviar feedback no CodeCatalyst	1282
Solucionar problemas de repositórios de origem	1283
Atingi o máximo de armazenamento do meu espaço e são exibidos avisos ou erros	1283
Eu recebo um erro ao tentar clonar ou enviar para um repositório de origem do Amazon CodeCatalyst	1284
Eu recebo um erro ao tentar confirmar ou enviar para um repositório de origem do Amazon CodeCatalyst	1285
Preciso de um repositório de origem para meu projeto	1285
Meu repositório de origem é totalmente novo, mas contém uma confirmação	1286
Quero uma ramificação diferente como minha ramificação padrão	1286
Estou recebendo e-mails sobre atividades em solicitações pull	1286
Esqueci meu token de acesso pessoal (PAT)	1286

Uma solicitação pull não exibe as alterações esperadas	1287
Uma solicitação pull mostra o status Não mesclável	1287
Solução de problemas de projetos e esquemas	1288
API Java com dependências ausentes do esquema AWS Fargate para apache-maven-3.8.6	1288
O fluxo de trabalho moderno de três camadas do blueprint de aplicativos web	
OnPullRequestfalha com erro de permissões para a Amazon CodeGuru	1289
Ainda está tentando resolver seu problema?	1292
Solucionar problemas em Ambientes de Desenvolvimento	1292
A criação do meu Ambiente de Desenvolvimento não foi bem-sucedida devido a um problema com cotas	1293
Não consigo enviar alterações do meu Ambiente de Desenvolvimento para uma ramificação específica em um repositório	1294
Meu Ambiente de Desenvolvimento não foi retomado	1294
Meu Ambiente de Desenvolvimento foi desconectado	1294
Meu Ambiente de Desenvolvimento conectado à VPC falhou	1295
Não consigo encontrar em qual diretório meu projeto está	1295
Não consigo me conectar ao Ambiente de Desenvolvimento via SSH	1295
Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque minha configuração SSH local está ausente	1295
Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque estou com problemas com AWS Config para o perfil codecatalyst	1296
Não consigo criar um Ambiente de Desenvolvimento quando estou conectado ao CodeCatalyst usando uma conta de login único	1296
Solução de problemas de IDEs	1297
Solução de problemas de devfiles	1299
Solucionar problemas com fluxos de trabalho	1301
Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?	1302
Como faço para corrigir os erros “A definição do fluxo de trabalho tem <i>n</i> erros”?	1303
Como faço para corrigir os erros “Não é possível localizar credenciais” e “ExpiredToken”?	1305
Como faço para corrigir os erros “Não foi possível conectar-se ao servidor”?	1306
Por que os campos do CodeDeploy estão ausentes no editor visual?	1307
Como faço para corrigir erros de recursos do IAM?	1307
Como faço para corrigir erros de “instalação npm”?	1309
Por que vários fluxos de trabalho têm o mesmo nome?	1313
Posso armazenar meus arquivos de definição de fluxo de trabalho em outra pasta?	1314

Como adiciono ações em sequência ao meu fluxo de trabalho?	1314
Por que meu fluxo de trabalho é validado com sucesso, mas falha em runtime?	1314
A descoberta automática não descobre nenhum relatório da minha ação	1314
Minha ação falha em relatórios descobertos automaticamente depois que eu configuro os critérios de sucesso	1315
A descoberta automática gera relatórios que eu não quero	1316
A descoberta automática gera diversos pequenos relatórios para uma única estrutura de teste	1316
Os fluxos de trabalho listados em CI/CD não correspondem aos do repositório de origem .	1317
Não consigo criar ou atualizar fluxos de trabalho	1318
Solução de problemas	1318
Não consigo selecionar um destinatário para o meu problema	1318
Solucionar problemas com pesquisa	1319
Não consigo encontrar um usuário no meu projeto	1319
Não vejo o que estou procurando no meu projeto ou espaço	1319
O número de resultados de pesquisa continua mudando quando eu navego pelas páginas	1319
Minha consulta de pesquisa não está sendo concluída	1320
Solução de problemas de extensões	1320
Não consigo ver as alterações em um repositório de terceiros vinculado nem pesquisar os resultados dessas alterações	1320
Solução de problemas de contas associadas	1321
Minha solicitação de conexão da Conta da AWS recebe um erro de token inválido	1321
Meu fluxo de trabalho do projeto Amazon CodeCatalyst falha com um erro na conta configurada, no ambiente ou no perfil do IAM	1322
Preciso de uma conta, perfil e ambiente associados para criar um projeto	1323
Não consigo acessar a página de espaços do Amazon CodeCatalyst no Console de gerenciamento da AWS	1324
Quero uma conta diferente como minha conta de faturamento	1324
O fluxo de trabalho do meu projeto falha com um erro de nome de conexão	1324
Solução de problemas de AWS CLI e problemas de SDK	1325
Eu recebo uma mensagem de erro quando entro em aws codecatalyst em uma linha de comando ou terminal dizendo que é uma escolha inválida	1325
Eu recebo um erro de credenciais quando executo comandos aws codecatalyst	1325
Noções básicas sobre o status atual do serviço com o relatório de integridade do CodeCatalyst	1327
Conceitos do relatório de integridade do CodeCatalyst	1328

O incidente	1328
Status	1328
Recursos afetados	1328
Atualizado em	1328
Suporte para o Amazon CodeCatalyst	1329
Cobrança do Suporte para o Amazon CodeCatalyst	1329
Configuração do espaço para o Suporte para o Amazon CodeCatalyst	1332
Acesso ao suporte para o CodeCatalyst no Console de gerenciamento da AWS	1333
Criação de um caso de suporte do CodeCatalyst no CodeCatalyst	1334
Resolução de um caso de suporte no CodeCatalyst	1337
Reabertura de um caso de suporte no CodeCatalyst	1337
Cotas	1339
Histórico do documento	1341
AWS Glossário	1373

A Amazon não CodeCatalyst está mais aberta a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte [Como migrar do CodeCatalyst](#).

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.

O que é o Amazon CodeCatalyst?

O Amazon CodeCatalyst é um serviço integrado para equipes de desenvolvimento de software que adotam práticas contínuas de integração e implantação em seus processos de desenvolvimento de software. O CodeCatalyst coloca as ferramentas de que você precisa em um só lugar. Você pode planejar o trabalho, colaborar no código e compilar, testar e implantar aplicações com ferramentas de integração contínua/entrega contínua (CI/CD). Você também pode integrar recursos da AWS a seus projetos conectando suas Contas da AWS ao seu espaço do CodeCatalyst. Ao gerenciar todos os estágios e aspectos do ciclo de vida da aplicação em uma única ferramenta, você pode fornecer software com rapidez e confiança.

No CodeCatalyst, você cria um espaço para representar sua empresa, departamento ou grupo e, depois, cria projetos que contêm os recursos necessários para oferecer suporte às suas equipes e tarefas de desenvolvimento. Os recursos do CodeCatalyst são estruturados em projetos que estão dentro de espaços. Para ajudar as equipes a começarem rapidamente, o CodeCatalyst fornece esquemas de projetos baseados em linguagem ou ferramentas. Quando você cria um projeto a partir de um esquema de projeto, o projeto vem com recursos, como um repositório de origem com código de amostra, scripts de compilação, ações de implantação, servidores virtuais ou recursos sem servidor e muito mais.

O que posso fazer com o CodeCatalyst?

Você e sua equipe de desenvolvimento podem usar o CodeCatalyst para cada aspecto do desenvolvimento de software, desde o planejamento do trabalho até a implantação das aplicações. Use o CodeCatalyst para:

- Iterar e colaborar no código – Trabalhe de forma colaborativa com sua equipe no código com ramificações, mesclagens, solicitações pull e comentários em seus repositórios de código-fonte. Crie Ambientes de Desenvolvimento para trabalhar no código rapidamente sem precisar clonar ou configurar conexões com repositórios.
- Criar, testar e implantar a aplicação com fluxos de trabalho: configure fluxos de trabalho com ações de compilação, teste e implantação para lidar com a integração e entrega contínuas das aplicações. Você pode iniciar fluxos de trabalho manualmente ou configurá-los para serem iniciados automaticamente com base em eventos como envios de código ou criação ou fechamento de solicitações pull.

- Priorizar o trabalho da sua equipe com o rastreamento de problemas – Use problemas para criar listas de pendências e monitorar o status das tarefas em andamento com as placas. Criar e manter um acúmulo saudável de itens para sua equipe trabalhar é uma parte importante do desenvolvimento de software.
- Configurar o monitoramento e as notificações – Monitore a atividade da equipe e o status dos recursos e configure as notificações para se manter atualizado com as alterações importantes.

Como posso começar a usar o CodeCatalyst?

Se você não tiver um espaço ou quiser aprender a configurar e gerenciar um, recomendamos que você comece lendo o [Guia do administrador do Amazon CodeCatalyst](#).

Se você for novo no trabalho em um projeto ou espaço, recomendamos que comece:

- Revisando o [CodeCatalyst conceitos](#)
- [Criar um espaço](#)
- Criando seu primeiro projeto seguindo as etapas em [Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis](#)

Saiba mais sobre o CodeCatalyst

É possível saber mais sobre a funcionalidade do CodeCatalyst neste guia do usuário, bem como nos seguintes recursos:

- [AWS Artigos do blog DevOps sobre o Amazon CodeCatalyst](#)
- [Guia de referência da API do Amazon CodeCatalyst](#)
- [Guia do desenvolvedor do Amazon CodeCatalyst Action Development Kit](#)
- [Perguntas frequentes sobre o CodeCatalyst](#)
- [Depoimentos](#)

Como migrar do CodeCatalyst

Após uma análise cuidadosa, tomamos a decisão de fechar o acesso de novos clientes à Amazon CodeCatalyst, a partir de 7 de novembro de 2025. CodeCatalyst Os clientes existentes da Amazon podem continuar usando o serviço com os espaços existentes, mas não poderão criar novos espaços. AWS continua investindo em segurança e disponibilidade para a Amazon CodeCatalyst, mas não planejamos introduzir novos recursos.

Os clientes podem migrar manualmente seus dados da Amazon CodeCatalyst para outros provedores. Este documento descreve as formas básicas de migrar, extrair ou excluir dados dos consoles de AWS gerenciamento CodeCatalyst e. Recursos e dados criados em outros serviços AWS ou em 3P por meio do CodeCatalyst console precisarão ser excluídos por meio desses serviços para interromper o acúmulo de cobranças (se aplicável).

Os clientes podem considerar a migração para [GitLab o Duo com o Amazon Q](#), que anunciou a disponibilidade geral em 17 de abril de 2025. [Essa nova oferta é um produto integrado que reúne a GitLab DevSecOps plataforma com os recursos de IA generativa do Amazon Q](#). O Gitlab Duo com Amazon Q incorpora os recursos DevSecOps do agente Amazon Q diretamente GitLab na plataforma para acelerar tarefas complexas de várias etapas em todo o ciclo de vida do desenvolvimento de software.

Migrar seus repositórios

Migrando seu CodeCatalyst repositório para um repositório GitLab

[Usando o URL de pré-requisito em combinação com as credenciais do repositório HTTPS Git, siga as orientações na documentação sobre a importação do código-fonte GitLab de um repositório por URL.](#)

Migrando seu CodeCatalyst repositório para um repositório GitHub

[Usando o URL de pré-requisito em combinação com as credenciais do repositório HTTPS Git, siga as orientações na documentação sobre a importação do código-fonte GitHub.](#)

Migração genérica para outro provedor de repositório

1. Clone o repositório CodeCatalyst

Clone o CodeCatalyst repositório da Amazon em sua máquina local usando o Git. Se estiver usando HTTPS, você pode fazer isso executando o seguinte comando:

```
git clone --mirror https://your-aws-repository-url your-aws-repository
```

`your-aws-repository-url` Substitua pela URL do seu CodeCatalyst repositório Amazon.

Substitua `your-aws-repository` por um nome para esse repositório.

Exemplo:

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

2. Configurar um novo ponteiro de repositório remoto

Navegue até o diretório do seu CodeCatalyst repositório clonado da Amazon. Em seguida, adicione o URL do repositório do novo provedor de repositórios como remoto:

```
git remote add <provider name> <provider-repository-url>
```

Substitua `<provider name>` pelo nome do provedor que você escolher. (Exemplo: gitlab)

Substitua `< provider-repository-url >` pela URL do repositório do seu novo provedor de repositório.

3. Envie seu repositório local para o novo repositório remoto:

Isso enviará todas as ramificações e tags para o repositório do seu novo provedor de repositórios. O nome do provedor deve corresponder ao nome do provedor da etapa 2.

```
git push <provider name> --mirror
```

Observações:

- O repositório remoto deve estar vazio
- O repositório remoto pode ter ramificações protegidas que não permitem o envio forçado, dependendo do provedor. Se isso acontecer, você precisará navegar até seu novo provedor de repositórios e desativar as proteções de ramificação para permitir o envio forçado.

4. Verificar a migração

Depois que o push for concluído, verifique se todos os arquivos, ramificações e tags foram migrados com sucesso para o novo provedor de repositórios. Para fazer isso, navegue no seu repositório on-line ou clone-o em outro local e verifique-o localmente.

5. Atualizar controle remoto URLs (opcional)

Se você planeja continuar trabalhando com o repositório migrado localmente, talvez queira atualizar a URL remota para apontar para o repositório do novo provedor em vez da Amazon. CodeCatalyst Faça isso usando o seguinte comando:

```
git remote set-url origin <provider-repository-url>
```

Substitua < provider-repository-url > pela URL do repositório do seu novo provedor de repositório.

Extraindo seus dados de CodeCatalyst

Baixar artefatos

Você pode baixar e inspecionar artefatos gerados por suas ações de CodeCatalyst fluxo de trabalho da Amazon. Há dois tipos de artefatos que podem ser baixados:

- Artefatos de origem: um artefato que contém um snapshot do conteúdo do repositório de origem tal como ele existia quando a execução foi iniciada.
- Artefatos do fluxo de trabalho: um artefato definido na propriedade de saída do arquivo de configuração do fluxo de trabalho.

Como baixar artefatos gerados pelo fluxo de trabalho:

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

5. Abaixo do nome do fluxo de trabalho, selecione Execuções.
6. Em Histórico de execução, na coluna ID da execução, escolha uma execução. Por exemplo, Run-95a4d.
7. Abaixo do nome da execução, selecione Artefatos.
8. Ao lado de um artefato, selecione Baixar. Um arquivo será baixado. Seu nome de arquivo consiste em sete caracteres aleatórios.
9. Extraia o arquivo usando um utilitário de extração de arquivos de sua escolha.

Baixar os anexos do problema

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Escolha o problema para o qual deseja gerenciar os anexos. Para receber ajuda com a descoberta do problema, consulte [Encontrar e visualizar problemas](#)
3. Para baixar um anexo, escolha o menu de reticências ao lado do anexo que você deseja baixar e escolha Baixar.

Baixar o código-fonte de uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. Encontre a ação cujo código você deseja visualizar:
 - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
 - b. Selecione o nome de qualquer fluxo de trabalho ou crie um. Para obter informações sobre a criação de um fluxo de trabalho, consulte [Criar um fluxo de trabalho](#).
 - c. Escolha Editar.
 - d. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
 - e. Na lista suspensa, escolha Amazon CodeCatalyst para visualizar CodeCatalyst, CodeCatalyst Labs e ações de terceiros.
 - f. Pesquise uma ação e selecione seu nome. Não selecione o sinal de mais (+).

Os detalhes sobre a ação são exibidos.

4. Na caixa de diálogo de detalhes da ação, na parte inferior, selecione Download.

Uma página é exibida com o bucket do Amazon S3 em que o código-fonte da ação está localizado. Para ter informações sobre o Amazon S3, consulte [O que é o Amazon S3?](#) no Guia do usuário do Amazon Simple Storage Service.

Excluindo seus dados do CodeCatalyst

Antes de excluir seus dados CodeCatalyst, informe sua equipe sobre a migração do serviço e verifique se ainda não são necessários recursos. Depois que os dados e os recursos são excluídos, não é possível recuperá-los.

Solicite que a equipe de serviço exclua dados em seu nome

Um administrador do Space pode solicitar que a equipe de serviço exclua um espaço em seu nome entrando em contato conosco pelo Support Center no CodeCatalyst console. Os administradores do espaço devem estar autenticados no CodeCatalyst console para solicitar a exclusão do espaço. Após o envio da sua solicitação, a equipe de atendimento entrará em contato com você para confirmá-la antes de agir em seu nome.

Exclua seu CodeCatalyst espaço

Você pode excluir um espaço para remover o acesso a todos os recursos do espaço. Você deve ter o perfil de administrador do espaço para excluir um espaço.

Observação: você não pode desfazer uma exclusão de espaço, e não é possível recuperar os dados depois que um espaço é excluído.

Depois de excluir um espaço, nenhum membro do espaço poderá acessar os recursos do espaço. O faturamento por recursos do espaço também será interrompido e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

Se você pertencer a mais de um espaço, escolha-o na barra de navegação superior.

Como excluir um espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, Excluir.
4. Digite **delete** para confirmar a exclusão.
5. Escolha Excluir.

 Note

Se você pertencer a mais de um espaço, será redirecionado para a página de visão geral do espaço. Se você pertencer a um espaço, será redirecionado para a página de criação do espaço.

Se você excluir um espaço, mas pertencer a mais de um, será redirecionado para a página de visão geral do espaço. Se você pertencer a um espaço, será redirecionado para a página de criação do espaço.

Se você criou recursos em outros serviços AWS ou em serviços de terceiros a partir do CodeCatalyst console, precisará acessar esses serviços individualmente para encerrar os recursos da conta de cobrança na qual eles foram criados. A exclusão de um espaço excluirá apenas os CodeCatalyst dados e os recursos.

Excluir um projeto

Você pode excluir um projeto para remover todo o acesso aos recursos do projeto. Você deve ter o perfil de Administrador do espaço ou Administrador do projeto para excluir um projeto. Depois que você excluir um projeto, os membros do projeto não conseguirão acessar os recursos do projeto e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

Para excluir o projeto:

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.

3. No painel de navegação, escolha Configurações do projeto.
4. Escolha Excluir projeto.
5. Insira **delete** para confirmar a exclusão.
6. Escolha Excluir projeto.

Se você criou recursos em outros serviços AWS ou em serviços de terceiros a partir de um CodeCatalyst projeto, precisará acessar esses serviços individualmente para encerrar os recursos da conta de faturamento em que foram criados. A exclusão de um espaço excluirá apenas os CodeCatalyst dados e os recursos.

Excluir um repositório de origem

Você pode excluir seus repositórios de origem para um CodeCatalyst projeto da Amazon. A exclusão de um repositório de origem também exclui todas as informações do projeto armazenadas no repositório. Se algum fluxo de trabalho depender do repositório de origem, esses fluxos de trabalho serão excluídos da lista de fluxos de trabalho do projeto após a exclusão do repositório. Os problemas que fazem referência ao repositório de origem não serão excluídos nem alterados, mas quaisquer links para o repositório de origem adicionados aos problemas falharão quando o repositório for excluído.

Importante: a exclusão de um repositório de origem não pode ser desfeita. Depois de excluir um repositório de origem, você não poderá mais cloná-lo, extrair nem enviar dados para ele. A exclusão de um repositório de origem não exclui nenhuma cópia local do repositório (repositórios locais). Para excluir um repositório local, use as ferramentas de gerenciamento de arquivos e diretórios do seu computador local.

Observação: você não pode excluir um repositório vinculado no CodeCatalyst console. Para excluir um repositório vinculado, selecione o link na lista de repositórios para abrir esse repositório no serviço que o hospeda e, depois, exclua-o. Para obter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.

Para remover um repositório vinculado de um projeto, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira](#) em. CodeCatalyst

Como excluir um repositório de origem

1. Navegue até o projeto que contém o repositório de origem que deseja excluir.

2. Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Selecione o nome do repositório na lista de repositórios de origem do projeto.
3. Na página inicial do repositório, escolha Mais, Gerenciar configurações e Excluir repositório.
4. Revise a ramificação, as solicitações pull e as informações relacionadas ao fluxo de trabalho para ajudar a garantir que você não esteja excluindo um repositório que ainda esteja em uso ou tenha um trabalho inacabado. Se quiser continuar, digite excluir e selecione Excluir.

Excluir um esquema personalizado

Quando você exclui um blueprint do seu CodeCatalyst espaço na Amazon, todo o seu acesso é removido aos recursos do projeto ou da versão do blueprint. Quando você excluir um esquema, os membros do projeto não conseguirão acessar os recursos do projeto, e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

Se você excluir um esquema, isso não afetará um projeto ao qual o esquema foi aplicado. Os recursos do esquema não são removidos do projeto.

Importante: para excluir um esquema personalizado publicado ou a versão do catálogo de um esquema personalizado do seu espaço, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja excluir um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Na tabela Configurações, escolha o botão de opção do esquema personalizado que você deseja excluir e, depois, selecione Excluir esquema.
5. Insira delete para confirmar a exclusão da versão do catálogo de esquemas.
6. Escolha Excluir.

Excluir os anexos do problema

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Escolha o problema para o qual deseja gerenciar os anexos. Para receber ajuda com a descoberta do problema, consulte [Encontrar e visualizar problemas](#).
3. Para remover um anexo, escolha o menu de reticências ao lado do anexo que você deseja remover e escolha Excluir.

Excluindo um arquivo em ambientes de desenvolvimento acessados pela Amazon CodeCatalyst

Você pode excluir arquivos em um Ambiente de Desenvolvimento, localmente no seu computador ou em um ambiente de desenvolvimento integrado (IDE). Você não pode excluir arquivos no CodeCatalyst console da Amazon.

Excluir um Ambiente de Desenvolvimento para o espaço

Para obter mais informações sobre considerações para excluir um Ambiente de Desenvolvimento, consulte [Excluir um Ambiente de Desenvolvimento](#).

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar Ambientes de Desenvolvimento no nível do espaço. Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

Como excluir um Ambiente de Desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço
3. Selecione Configurações e, depois, selecione Ambientes de Desenvolvimento.
4. Escolha o seletor ao lado do Ambiente de Desenvolvimento que você deseja gerenciar. Escolha Excluir.
5. Digite **delete** para confirmar a exclusão do Ambiente de Desenvolvimento.
6. Escolha Excluir.

Excluir conexões de conta

Você pode excluir uma conexão de conta no CodeCatalyst console que você adicionou anteriormente ao seu espaço. Depois que uma conexão de conta for excluída, você não poderá reconectá-la e precisará estabelecer uma nova.

Uma conta de cobrança deve ser designada para seu CodeCatalyst espaço, mesmo que o uso do espaço não exceda o nível gratuito. Antes de remover um espaço de uma conta que é uma conta de faturamento designada, você precisará adicionar outra conta ao seu espaço. Se você quiser excluir sua conta de cobrança do espaço, precisará excluir seu espaço. Consulte [Gerenciamento de faturamento](#) no Guia do CodeCatalyst Administrador da Amazon.

Para gerenciar conexões de conta para seu espaço, você deve ter o perfil de Administrador do espaço ou Usuário avançado.

Uma conta que foi removida pode ser adicionada novamente mais tarde, mas você deve criar uma conexão entre a conta e o espaço. Será necessário associar novamente todos os perfis do IAM à conta.

Como excluir uma conexão de conta

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
3. Em Nome de CodeCatalyst exibição da Amazon, escolha o seletor ao lado da conexão da conta que você deseja remover.
4. Escolha Remover Conta da AWS. Confirme a exclusão inserindo o nome no campo e selecione Remover.

Será exibido um banner de sucesso e a conexão da conta é removida da lista de conexões.

Remover uma conta de um CodeCatalyst espaço no AWS Management Console

Você pode usar a página CodeCatalyst de entrada AWS para remover uma conta que foi adicionada a um espaço. Para esse procedimento, usando permissões administrativas para a conta específica que você está gerenciando, faça login na página Amazon CodeCatalyst Spaces no AWS Management Console para remover uma AWS conta do seu espaço. Para remover uma conta que é uma conta de cobrança designada para seu CodeCatalyst espaço, certifique-se primeiro de especificar uma nova conta de cobrança.

Uma conta que foi removida pode ser adicionada novamente mais tarde, mas você deve criar uma conexão entre a conta e o espaço. Será necessário reassociar todos os perfis do IAM à conta adicionada.

Uma conta de cobrança deve ser designada para seu CodeCatalyst espaço, mesmo que o uso do espaço não exceda o nível gratuito. Antes de remover um espaço de uma conta que é uma conta de faturamento designada, você precisará adicionar outra conta ao seu espaço.

Você deve ter o perfil de Administrador do espaço ou Usuário avançado para gerenciar conexões de conta para seu espaço.

Para remover uma conta adicionada

1. No Console de gerenciamento da AWS, verifique se você está logado com a mesma conta que deseja gerenciar.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Configurações e, depois, Faturamento.
4. Confira as informações da conta de faturamento na página para verificar se a conta que você quer remover não é a conta de faturamento designada para o espaço.
5. Escolha Gerenciar cobrança em AWS. Isso abre o Amazon CodeCatalyst Spaces no Console de gerenciamento da AWS. Se solicitado, faça login na AWS e selecione o botão novamente para carregar a página.
6. Na página Amazon CodeCatalyst Spaces, escolha o espaço com a conta que você deseja remover. A página de detalhes do espaço é exibida.
7. Escolha Remover espaço.
8. Em Remover CodeCatalyst espaço desta conta, insira o nome do espaço para confirmar. Escolha Remover .

Excluir um segredo

Use o procedimento a seguir para excluir um segredo e o identificador de referência do segredo. Antes de excluir um segredo, recomendamos que você remova o identificador de referência do segredo de todas as ações do fluxo de trabalho. Se você excluir o segredo sem excluir o identificador de referência, a ação falhará na próxima vez em que for executada.

Como excluir o identificador de referência de um segredo de um fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione YAML.
6. Pesquise no fluxo de trabalho a seguinte string:

```
${Secrets.
```

Todos os identificadores de referência de todos os segredos serão encontrados.

7. Exclua o identificador de referência do segredo selecionado ou substitua-o por um valor de texto simples.
8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Para excluir um segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Segredos.
3. Na lista de segredos, selecione o segredo que você quer excluir.
4. Escolha Excluir.
5. Insira **delete** para confirmar a exclusão.
6. Escolha Excluir.

Exclusão de uma equipe

Você pode excluir uma equipe da qual não precisa mais. Quando você exclui uma equipe, as permissões associadas são removidas para todos os membros da equipe de todos os projetos e recursos no espaço. Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Excluir uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.

3. Em Ações, selecione Excluir equipe. Isso mudará o perfil da equipe inteira.
4. Escolha Excluir.

Exclusão de uma frota provisionada

Use as instruções a seguir para excluir uma frota provisionada.

Antes de excluir uma frota provisionada, remova-a de todas as ações excluindo a propriedade da frota do código YAML da ação. Qualquer ação que continue referenciando uma frota provisionada depois que ela for excluída falhará na próxima vez em que a ação for executada.

Para excluir uma frota provisionada

1. No painel de navegação, selecione CI/CD e Computação.
2. Na lista Frota provisionada, selecione a frota que você quer excluir.
3. Escolha Excluir.
4. Insira **delete** para confirmar a exclusão.
5. Escolha Excluir.

Exclusão de um repositório de pacotes

Execute as etapas a seguir para excluir um repositório de pacotes no CodeCatalyst.

Para excluir um repositório de pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto que contém o repositório de pacotes que deseja excluir.
3. No painel de navegação, selecione Pacotes.
4. Na página Repositórios de pacotes, escolha o repositório que deseja excluir.
5. Escolha Excluir.
6. Analise as informações fornecidas sobre os efeitos da exclusão de um repositório de pacotes.
7. Insira **delete** no campo de entrada e selecione Excluir.

Todos os outros recursos armazenados na Amazon CodeCatalyst serão excluídos quando seu espaço for excluído. Isso não inclui recursos e dados criados em outros serviços AWS ou serviços

3P por meio do CodeCatalyst console. Todos os recursos criados em serviços fora do CodeCatalyst console precisarão ser excluídos por meio desses serviços para interromper o acúmulo de cobranças.

Se você tiver outras dúvidas, entre em contato conosco em aws-codecatalyst-service@amazon.com ou entre em contato pelo Support Center no CodeCatalyst console da Amazon.

CodeCatalyst conceitos

Familiarize-se com os principais conceitos para ajudar a acelerar sua colaboração e o desenvolvimento de aplicativos na Amazon CodeCatalyst. Esses conceitos incluem termos usados em controle de origem, integração contínua e entrega contínua (CI/CD) e modelagem e configuração de processos de versão automatizados.

Para ter informações conceituais adicionais, consulte os seguintes tópicos:

- [Conceitos de repositório de origem](#)
- [Conceitos de fluxos de trabalho](#)

Tópicos

- [AWS Espaços de ID do construtor em CodeCatalyst](#)
- [Espaços que oferecem suporte à federação de identidades em CodeCatalyst](#)
- [Projetos](#)
- [Esquemas](#)
- [Conexões de conta](#)
- [Conexões VPC](#)
- [AWS ID do construtor](#)
- [Perfis de usuário em CodeCatalyst](#)
- [Repositórios de origem](#)
- [Confirmações](#)
- [Ambientes de Desenvolvimento](#)
- [Fluxos de trabalho](#)
- [Ações](#)
- [Problemas](#)
- [Tokens de acesso pessoal \(PATs\)](#)
- [Conexões pessoais](#)
- [Perfis](#)

AWS Espaços de ID do construtor em CodeCatalyst

O administrador do espaço convida os usuários CodeCatalyst enviando e-mails de convite individuais a partir da página de membros. Usuários que são convidados ou se inscrevem para CodeCatalyst criar seu próprio AWS Builder ID. O perfil é gerenciado no AWS Builder ID e é exibido como o nome do usuário e as informações do perfil nas configurações do usuário em CodeCatalyst.

Espaços que oferecem suporte à federação de identidades em CodeCatalyst

Usuários que foram adicionados aos usuários e grupos de SSO da instância do IAM Identity Center e são gerenciados no armazenamento de identidades e convidados para seu espaço por meio do IAM Identity Center. O administrador do Space sincroniza a página dos CodeCatalyst membros para obter as atualizações mais recentes. Os usuários fazem login usando o portal de login SSO, conforme configurado na instância do IAM Identity Center da empresa. Os espaços que oferecem suporte à federação de identidades são conectados à instância do armazenamento de identidades por meio da aplicação do Centro de Identidade e do mapeamento para o ID do armazenamento de identidades.

Projetos

Um projeto representa um esforço colaborativo CodeCatalyst que apoia equipes e tarefas de desenvolvimento. Depois que tiver um projeto, você poderá adicionar, atualizar ou remover usuários e recursos, personalizar o painel do projeto e monitorar o andamento do trabalho da equipe. Você pode ter vários projetos em um espaço.

Para ter mais informações sobre projetos, consulte [Organize o trabalho com projetos em CodeCatalyst](#).

Esquemas

Um blueprint é um sintetizador de projeto que gera e estende arquivos de suporte de aplicativos e dependências para você, além de criar seu CodeCatalyst projeto no console. Você escolhe um tipo de projeto em uma seleção de esquemas CodeCatalyst, visualiza o arquivo README e visualiza o repositório do projeto e os recursos que serão gerados. Seu projeto é gerado a partir da configuração básica especificada pelo esquema. Você sintetiza periodicamente o esquema do projeto, o que atualiza os arquivos do projeto, como dependências de software, e gera os recursos novamente. Os projetos usam uma ferramenta chamada Projen para sintetizar projetos sincronizando as

atualizações mais recentes do projeto e gerando arquivos de suporte. Esses arquivos podem incluir `package.json`, `Makefile`, `eslint` e muito mais com base no tipo e na linguagem da aplicação. Os blueprints do projeto podem gerar arquivos que suportam AWS recursos, como construções, CloudFormation modelos e modelos de CDK. AWS Serverless Application Model

Para ter mais informações sobre esquemas de projeto, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#).

Conexões de conta

Uma conexão de conta associa um CodeCatalyst espaço ao seu Conta da AWS. Depois que a conexão da sua conta for configurada, Conta da AWS ela será disponibilizada para o espaço. Em seguida, você pode adicionar funções do IAM para CodeCatalyst que ele possa acessar recursos em seu Conta da AWS. Você também pode usar essas funções para suas ações CodeCatalyst de fluxo de trabalho.

Você pode limitar quais projetos e recursos têm acesso às conexões da conta ativando conexões de conta restritas ao projeto. Conexões de contas restritas a projetos estão Contas da AWS conectadas e só podem ser acessadas por projetos específicos no espaço. Isso permite que as equipes em um espaço restrinjam o uso de AWS recursos integrados Contas da AWS por projeto. Por exemplo, a conta usada para fluxos de trabalho de implantação e conexões VPC em projetos específicos só estará disponível com uma conexão de conta restrita ao projeto. Para ter mais informações, consulte [Configuração de conexões de contas restritas ao projeto](#).

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Conexões VPC

Uma conexão VPC é um CodeCatalyst recurso que contém todas as configurações necessárias para que seu fluxo de trabalho acesse uma VPC. Os administradores do espaço podem adicionar suas próprias conexões de VPC no console da CodeCatalyst Amazon em nome dos membros do espaço. Ao adicionar uma conexão VPC, os membros do espaço podem executar ações de fluxo de trabalho e criar ambientes de desenvolvimento que sigam as regras de rede e possam acessar recursos na VPC associada.

Para obter mais informações sobre conexões VPC, consulte Gerenciando [Amazon Virtual Private Clouds no Guia](#) do CodeCatalyst Administrador.

AWS ID do construtor

Um AWS Builder ID é uma identidade pessoal que você pode usar para se inscrever CodeCatalyst e entrar em outros aplicativos participantes. Não é o mesmo que um Conta da AWS. Seu AWS Builder ID gerencia metadados, como alias de usuário e endereço de e-mail. Seu AWS Builder ID é uma identidade exclusiva que oferece suporte a usuários em todos os espaços em CodeCatalyst. Para obter informações sobre como acessar seu perfil AWS Builder ID, consulte [Atualização de um perfil](#). Para saber mais sobre o AWS Builder ID, consulte [AWS Builder ID](#) no Referência geral da AWS.

Para ter mais informações sobre como se inscrever e fazer login, consulte [Configuração e login no CodeCatalyst](#).

Perfis de usuário em CodeCatalyst

Você acessa seu perfil de CodeCatalyst usuário escolhendo a opção de perfil no menu suspenso abaixo de suas iniciais de login em qualquer página em. CodeCatalyst Você pode criar tokens de acesso pessoal (PATs) na sua página de perfil, mas só pode visualizar ou excluir PATs usando AWS CLI o. Seu nome de usuário é o alias que você escolheu quando se inscreveu. Não é possível alterar seu nome de usuário. Para ver a página de perfil de outro CodeCatalyst usuário, acesse a guia Membros do seu projeto e escolha o usuário apropriado.

Você acessa sua ID do AWS Construtor visualizando seu CodeCatalyst perfil e, em seguida, escolhendo acessar a ID do AWS Construtor. Você será redirecionado para sua página de perfil AWS Builder ID. O nome completo, endereço de e-mail e senha do seu perfil são gerenciados pelo seu ID do AWS Construtor, e você pode editar essas informações usando a página ID do AWS Construtor. Você inseriu essas informações ao se inscrever. Quando estiver pronto para configurar o MFA para usar um aplicativo autenticador para fazer login, você usará a página AWS Builder ID. Para obter mais informações sobre como visualizar seu perfil de AWS Builder ID, consulte [Atualização de um perfil](#).

Para ter mais informações sobre como se inscrever e fazer login, consulte [Configuração e login no CodeCatalyst](#).

Repositórios de origem

Em um repositório de origem, você armazena com segurança códigos e arquivos do seu projeto. Ele também armazena o histórico de versões dos seus arquivos. Por padrão, um repositório de

origem é compartilhado com os outros usuários em seu CodeCatalyst projeto. É possível ter mais de um repositório de origem para um projeto. Você pode criar repositórios de origem para projetos em CodeCatalyst, ou você pode optar por vincular um repositório de origem existente hospedado por outro serviço, se esse serviço for suportado por uma extensão instalada. Por exemplo, você pode vincular um GitHub repositório a um projeto depois de instalar a extensão GitHub Repositories. Para obter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#) e [Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst](#).

Os repositórios de origem também são onde as informações de configuração do seu CodeCatalyst projeto são armazenadas, como o arquivo de configuração que define os atributos e as ações do seu fluxo de trabalho de CI/CD. Se você criar seu projeto usando um esquema, um repositório de origem será criado com as informações de configuração do projeto armazenadas nele. Se você criar um projeto vazio, deverá criar um repositório de origem antes de criar recursos que exijam informações de configuração, como fluxos de trabalho.

Para conhecer mais conceitos que podem ajudar você a trabalhar com repositórios de origem e controle de origem, consulte [Conceitos de repositório de origem](#).

Confirmações

Uma confirmação é uma alteração em um arquivo ou conjunto de arquivos. No CodeCatalyst console da Amazon, um commit salva suas alterações e as envia para um repositório de origem. A confirmação inclui informações sobre a alteração, incluindo a identidade do usuário que fez a alteração, a hora e a data da alteração, o título da confirmação e qualquer mensagem incluída sobre a alteração. Para obter mais informações, consulte [Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst](#).

No contexto de um repositório de origem em CodeCatalyst, os commits são instantâneos das alterações no conteúdo do seu repositório. Sempre que um usuário confirma e envia uma alteração, CodeCatalyst salva informações que incluem quem cometeu a alteração, a data e a hora da confirmação e as alterações feitas como parte da confirmação. Você também pode adicionar tags Git a confirmações para ajudar a identificar confirmações específicas.

Para ter mais informações sobre confirmações, consulte [Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst](#).

Ambientes de Desenvolvimento

Um ambiente de desenvolvimento é um ambiente de desenvolvimento baseado em nuvem que você pode usar CodeCatalyst para trabalhar rapidamente no código armazenado nos repositórios de origem do seu projeto. As ferramentas do projeto e as bibliotecas de aplicações incluídas no Ambiente de Desenvolvimento são definidas por um arquivo de desenvolvimento no repositório de origem do projeto. Se você não tiver um arquivo de desenvolvimento no repositório de origem, um arquivo padrão será aplicado automaticamente. O arquivo de desenvolvimento padrão inclui ferramentas para as linguagens e estruturas de programação usadas com mais frequência. Por padrão, um Ambiente de Desenvolvimento é configurado para ter um processador de 2 núcleos, 4 GB de RAM e 16 GiB de armazenamento persistente.

Fluxos de trabalho

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

Tip

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Para obter mais informações sobre fluxos de trabalho, consulte [Compilação, teste e implantação com fluxos de trabalho](#).

Ações

Uma ação é o principal componente de um fluxo de trabalho e define uma unidade lógica de trabalho, ou tarefa, a ser realizada durante a execução de um fluxo de trabalho. Normalmente, um fluxo de trabalho inclui várias ações que são executadas de modo sequencial ou paralelo, dependendo de como você as configurou.

Para ter mais informações sobre as ações, consulte [Configuração de ações de fluxo de trabalho](#).

Problemas

Um problema é um registro que rastreia o trabalho relacionado ao seu projeto. Você pode criar um problema para um recurso, uma tarefa, um bug ou qualquer outro trabalho relacionado ao projeto. Se você estiver usando o desenvolvimento ágil, um problema também poderá descrever uma história épica ou de usuário.

Para ter mais informações sobre problemas, consulte [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#).

Tokens de acesso pessoal (PATs)

Um token de acesso pessoal (PAT) é semelhante a uma senha. Ele está associado à sua identidade de usuário para uso em todos os espaços e projetos em CodeCatalyst. Você usa PATs para acessar CodeCatalyst recursos que incluem ambientes de desenvolvimento integrados (IDEs) e repositórios de origem baseados em Git. PATs representam você CodeCatalyst e você pode gerenciá-los em suas configurações de usuário. Um usuário pode ter mais de um PAT. Os tokens de acesso pessoal são exibidos apenas uma vez. Como prática recomendada, armazene-os com segurança em seu computador local. Por padrão, PATs expira após um ano.

Para obter mais informações sobre PATs, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).

Conexões pessoais

Uma conexão pessoal é uma autorização entre sua CodeCatalyst identidade e seu provedor de origem externo, como GitHub. Você usa conexões pessoais para permitir que um CodeCatalyst usuário adicione repositórios de origem de terceiros. Por exemplo, você pode conectar um GitHub repositório a um CodeCatalyst espaço. Um aplicativo conector instalado é instalado na GitHub conta para uso com repositórios designados pelo proprietário da conta. Você pode criar uma conexão pessoal para uma identidade de usuário (CodeCatalyst alias) em todos os espaços de um tipo específico de provedor, como GitHub. As conexões pessoais estão associadas ao seu AWS Builder ID ou ao seu usuário de SSO.

Para obter mais informações, consulte [Acessando GitHub recursos com conexões pessoais](#).

Perfis

Um perfil define o acesso de um usuário aos recursos de um projeto ou espaço e quais ações esse usuário pode realizar. Você escolhe o perfil de um usuário ao convidá-lo para um projeto. Existem funções em nível de espaço e funções em nível de projeto em CodeCatalyst. Um usuário com um perfil administrativo no nível correto pode alterar os perfis atribuídos. Por exemplo, um usuário com o perfil de Administrador do projeto tem controle total sobre esse projeto e pode alterar os perfis dos usuários nesse projeto. Para ter informações sobre quais perfis estão disponíveis e quais permissões cada perfil tem, consulte [Concessão de acesso com perfis de usuário](#).

Para ter mais informações sobre funções, consulte [Concessão de acesso com perfis de usuário](#).

Configuração e login no CodeCatalyst

Há dois tipos de espaço que você pode configurar no CodeCatalyst: espaços compatíveis com usuários do ID do builder AWS e criar um espaço compatível com a federação de identidades, onde usuários e grupos de SSO são gerenciados no Centro de Identidade do IAM. Os usuários em um espaço do ID do builder AWS entram no CodeCatalyst com seu ID do builder AWS, e os usuários em um espaço configurado para federação de identidades fazem login no CodeCatalyst usando o portal SSO da empresa associada ao espaço.

Note

Os nomes de usuário do CodeCatalyst têm um comprimento mínimo de 3 e máximo de 100 caracteres. Nomes de usuário com mais de 100 caracteres fornecidos ficarão truncados. Isso pode resultar em um nome de usuário que parece ser uma duplicata de outro nome de usuário de 100 caracteres. Para obter mais informações, consulte [Não consigo acessar meu espaço BID como um novo usuário ou não posso ser adicionado como um novo usuário de SSO devido ao nome de usuário truncado](#).

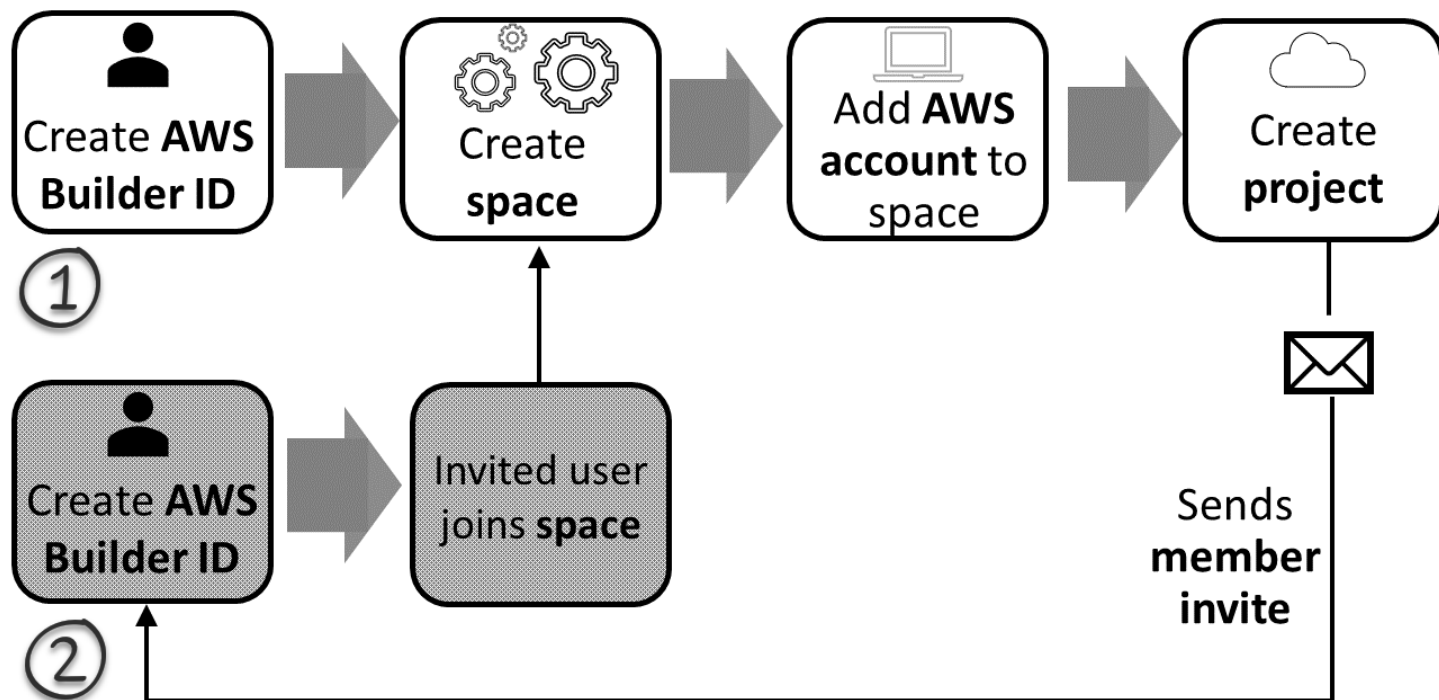
As etapas para configurar e administrar um espaço de ID do builder AWS são fornecidas neste guia. Para trabalhar com um espaço de ID do builder AWS do CodeCatalyst, configure o CodeCatalyst usando as configurações do usuário e o ID do builder AWS que você usa para entrar no CodeCatalyst.

As etapas para configurar e administrar um espaço compatível com a federação de identidades são fornecidas no Guia do administrador do CodeCatalyst. Para trabalhar com espaços configurados para federação de identidades, consulte [Configuração e administração de espaços do CodeCatalyst](#) no Guia do administrador do Amazon CodeCatalyst.

Esta seção fornece dois caminhos comuns para configurar o trabalho no Amazon CodeCatalyst com um espaço de ID do builder AWS: criar um espaço e um projeto como primeiro usuário e aceitar um convite para um espaço ou projeto existente. Esses fluxos de trabalho de configuração são necessariamente bem diferentes. O diagrama a seguir mostra os dois processos de inscrição da seguinte forma:

1. No primeiro caso, você cria e configura um espaço para sua empresa, equipe ou grupo e cria um projeto antes de convidar outras pessoas para esses recursos. Uma Conta da AWS deve ser fornecida para fins de faturamento, onde você ainda pode usar o nível gratuito como padrão.

2. No segundo caso, se você entrar no CodeCatalyst aceitando um convite para um projeto, outra pessoa já criou um espaço e um projeto para você. No entanto, você ainda desejará configurar seu perfil para estar pronto para começar a trabalhar com outras pessoas.



i Tip

O CodeCatalyst usa espaços para agrupar projetos e recursos. Ao se inscrever pela primeira vez no CodeCatalyst, você será solicitado a criar um espaço e um projeto.

Se você se inscrever para criar um espaço e um projeto ou se inscrever como parte da aceitação de um convite, você cria um ID do builder AWS que usará para fazer login no CodeCatalyst. Para criar um ID do builder AWS, você fornece o nome completo, a senha e o endereço de e-mail usado para entrar nas aplicações da AWS. Use o e-mail e a senha para entrar no CodeCatalyst após esse ponto. Você também pode usar esse ID do builder AWS para fazer login em outras aplicações que usam as credenciais do ID do builder AWS.

No CodeCatalyst e no ID do builder AWS, um perfil é gerado com base nas suas informações de login. Seu perfil contém suas preferências do CodeCatalyst para configurações de idioma e notificação em seus projetos do CodeCatalyst.

 Tip

Se você encontrar algum problema ao se inscrever no perfil do Amazon CodeCatalyst, siga as etapas fornecidas nesta página. Se precisar de mais ajuda, consulte [Problemas de inscrição](#).

Tópicos

- [Criar um espaço e perfil de desenvolvimento \(começando sem um convite\)](#)
- [Aceitando um convite e criando um AWS Builder ID](#)
- [Login com um ID do builder AWS](#)
- [Fazer login com SSO](#)
- [Visualização de todos os espaços e projetos de um usuário](#)
- [Visualização e gerenciamento de perfis do CodeCatalyst](#)
- [Configurando para usar o AWS CLI com CodeCatalyst](#)

Criar um espaço e perfil de desenvolvimento (começando sem um convite)

Você pode se inscrever no Amazon CodeCatalyst sem um convite para um espaço ou projeto existente. Ao fazer isso, você criará um espaço e um projeto depois de criar seu ID do builder AWS. Como parte da criação de um espaço, você precisará adicionar uma Conta da AWS para fins de faturamento.

 Tip

Se você encontrar algum problema ao se inscrever no perfil do Amazon CodeCatalyst, siga as etapas fornecidas nesta página. Se precisar de mais ajuda, consulte [Problemas de inscrição](#).

Aqui está um fluxo possível para um usuário que está começando com o CodeCatalyst sem um convite para um projeto ou espaço.

Mary Major é uma desenvolvedora interessada no CodeCatalyst e decide testá-lo. Ela navega até o console do CodeCatalyst e escolhe a opção de se inscrever e criar um ID do builder AWS. Mary fornece um endereço de e-mail e uma senha para criar o ID do builder AWS. Ela poderá usar o ID do builder AWS para fazer login no CodeCatalyst e em outras aplicações. Quando solicitado que ela escolha um alias, ela especifica `MaryMajor1` como o nome de usuário do CodeCatalyst que será exibido no CodeCatalyst e que outros membros do projeto usarão para @mencionar Mary.

Depois, Mary é automaticamente direcionada para criar um espaço. Como parte desse fluxo, Mary é convidada a associar uma Conta da AWS ao espaço que ela está criando para que possa ver o código de amostra no primeiro projeto de compilação e implantação. Ela adiciona essas informações e cria o espaço, onde escolhe a opção de criar um perfil de desenvolvimento de pré-visualização que pode ser usado para projetos no novo espaço. Mary opta por criar um projeto e, depois, visualiza uma lista de esquemas para projetos. Depois de analisar as informações dos esquemas disponíveis, ela decide experimentar o esquema de aplicação web moderna de três níveis para o primeiro projeto. Ela preenche os campos obrigatórios e cria o projeto. Assim que o projeto estiver pronto, ela será direcionada para uma página de resumo do projeto que inclui atividades recentes, bem como links para o código do projeto e o fluxo de trabalho que compila e implanta automaticamente esse código. Ela explora o código e o fluxo de trabalho, inclusive a visualização da aplicação web de exemplo implantada. Gostando do que vê, ela decide convidar alguns de seus colegas de trabalho para o projeto para começar a explorar o CodeCatalyst.

Quando tem um momento livre, Mary configura o ID do builder AWS para fazer login no CodeCatalyst com a autenticação multifator (MFA). Com a MFA configurada, Mary pode entrar no CodeCatalyst usando uma combinação da senha do CodeCatalyst e uma senha ou um token de uma aplicação de autenticação de terceiros aprovada.

Criar um espaço e perfis do IAM

Siga estas etapas para se inscrever no seu perfil do Amazon CodeCatalyst, criar um espaço e adicionar uma conta, um perfil de suporte e um perfil de desenvolvedor para o espaço.

O procedimento final cria e adiciona o perfil de desenvolvedor. O perfil de desenvolvedor é um perfil do IAM da AWS que permite que seus fluxos de trabalho do CodeCatalyst acessem recursos da AWS. O perfil de desenvolvedor é um perfil de serviço usado para gerenciar Serviços da AWS e será criado na conta que está conectada. O perfil de serviço é um [perfil](#)

[do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. O perfil terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).

 Note

Como prática recomendada de segurança, atribua acesso administrativo somente aos usuários administrativos e desenvolvedores que precisam gerenciar o acesso aos recursos da AWS no espaço.

Antes de começar, deve estar tudo pronto para fornecer um ID da Conta da AWS para uma conta que tenha privilégios administrativos. Tenha o seu ID de 12 dígitos da Conta da AWS em mãos. Para ter informações sobre como encontrar o ID da Conta da AWS, consulte [O ID da sua Conta da AWS e seu alias](#).

Como inscrever um novo usuário

1. Antes de começar no console do CodeCatalyst, abra o Console de gerenciamento da AWS e, depois, verifique se você está conectado com a mesma Conta da AWS que deseja usar para criar seu espaço.
2. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
3. Na página de boas-vindas, selecione Cadastrar-se. A página Criar seu ID do builder AWS é exibida. O ID do builder AWS é uma identidade que você cria para fazer login. Não é o mesmo que uma Conta da AWS.
4. Em Seu endereço de e-mail, insira o endereço de e-mail que você deseja associar ao CodeCatalyst. Escolha Próximo.
5. Em Seu nome, insira o nome e o sobrenome a serem exibidos nas aplicações em que você usa o ID do builder AWS. Espaços são permitidos. Esse será o nome do perfil do ID do builder AWS, como Mary Major. Você pode alterar esse nome depois.

Escolha Próximo. A página Verificação de e-mail é exibida.

6. Um código de verificação será enviado para o e-mail que você especificou. Insira este código em Código de verificação e selecione Verificar. Se você não receber o código após cinco minutos e não conseguir encontrá-lo nas pastas de spam ou de lixo eletrônico, selecione Reenviar código.

7. Depois de verificarmos seu código, insira uma senha que atenda aos requisitos em Senha e Confirmar senha.

Marque a caixa de seleção confirmando que você concorda com o Contrato do cliente da AWS e os Termos de serviço da AWS e, depois, selecione Criar ID do builder AWS.

8. Na página Criar seu alias do CodeCatalyst, insira um alias que você deseja usar como seu identificador de usuário exclusivo no CodeCatalyst. Escolha uma versão abreviada do seu nome sem espaços, como MaryMajor. Outros usuários do CodeCatalyst a usarão para @mencionar você em comentários e solicitações pull. Seu perfil do CodeCatalyst conterá seu nome completo do seu ID do builder AWS e seu alias do CodeCatalyst. Não é possível alterar o alias do CodeCatalyst posteriormente.

Seu nome completo e seu alias serão exibidos em diferentes áreas no CodeCatalyst. Por exemplo, o nome do seu perfil é exibido para a sua atividade listada no feed de atividades, mas os membros do projeto usarão seu alias para @mencionar você.

Escolha Próximo. A página é atualizada para mostrar a seção Criar seu espaço do CodeCatalyst.

9. Em Nome do espaço, insira o nome do seu espaço. Não é possível alterar essa opção mais tarde.

 Note

Os nomes de espaço devem ser exclusivos no CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

10. No menu suspenso Região da AWS, selecione a região onde você deseja armazenar o espaço e os dados do projeto. Não é possível alterar essa opção mais tarde.
11. Escolha Próximo. A página é atualizada para mostrar a página para adicionar uma Conta da AWS. Essa conta será usada como a conta de faturamento do espaço.
12. Em ID da Conta da AWS, insira o ID de 12 dígitos da conta que você deseja conectar ao seu espaço.

Em Token de verificação da conta da AWS, copie o ID do token gerado. O token é copiado automaticamente, mas talvez você queira armazená-lo enquanto aprova a solicitação de conexão da AWS.

13. Escolha Ir para o console AWS para verificar.

14. A página Verificar espaço do Amazon CodeCatalyst é aberta no Console de gerenciamento da AWS. Esta é a página Espaços do Amazon CodeCatalyst. Talvez seja necessário fazer login para acessá-la.

No Console de gerenciamento da AWS, selecione a mesma Região da AWS em que deseja criar o espaço.

Para acessar diretamente a página, faça login em Espaços do Amazon CodeCatalyst, no Console de gerenciamento da AWS em <https://console.aws.amazon.com/codecatalyst/home/>.

O campo do token de verificação no Console de gerenciamento da AWS é preenchido automaticamente com o token gerado no CodeCatalyst.

15. (Opcional) Em Níveis pagos autorizados, selecione Autorizar níveis pagos (Standard, Enterprise) para ativar os níveis pagos na conta de faturamento.

 Note

Isso não eleva o nível de faturamento para um nível pago. No entanto, configura a Conta da AWS para que você possa alterar o nível de faturamento do espaço a qualquer momento no CodeCatalyst. É possível ativar os níveis pagos a qualquer momento. Sem essa alteração, o espaço só pode usar o nível gratuito.

16. Escolha Verificar espaço.

Uma mensagem de sucesso da conta verificada é exibida para mostrar que a conta foi adicionada ao espaço.

17. Permaneça na página Verificar espaço do Amazon CodeCatalyst. Selecione o seguinte link: Para adicionar perfis do IAM para esse espaço, visualizar os detalhes do espaço.

A página de conexões com Detalhes do espaço do CodeCatalyst é aberta no Console de gerenciamento da AWS. Esta é a página Espaços do Amazon CodeCatalyst. Talvez seja necessário fazer login para acessá-la.

18. Retorne à página do CodeCatalyst e escolha Próximo.

19. Uma mensagem de status é exibida enquanto seu espaço está sendo criado. Quando o espaço é criado, o CodeCatalyst exibe a seguinte mensagem: Seu espaço está pronto. Sua última etapa é criar um projeto. Você pode executar uma das seguintes ações:

- Selecione Ignorar por enquanto.

- Selecione Criar seu primeiro projeto para seu espaço. Para ver um tutorial que mostra como criar um projeto com um esquema, consulte [Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis](#)

 Note

Se um erro de permissão ou um banner for exibido, atualize a página e tente visualizá-la novamente.

Como criar e adicionar o CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Antes de começar no console do CodeCatalyst, abra o Console de gerenciamento da AWS e, depois, verifique se você está conectado com a mesma Conta da AWS do espaço.
2. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
3. Acesse o espaço no CodeCatalyst. Escolha Settings (Configurações) e Contas da AWS.
4. Selecione o link da Conta da AWS na qual você deseja criar o perfil. A página Detalhes da Conta da AWS é exibida.
5. Selecione Gerenciar perfis no Console de gerenciamento da AWS.

A página Adicionar perfil do IAM ao espaço do Amazon CodeCatalyst é aberta no Console de gerenciamento da AWS. Esta é a página Espaços do Amazon CodeCatalyst. Talvez seja necessário fazer login para acessá-la.

6. Selecione Criar perfil de administrador de desenvolvimento do CodeCatalyst no IAM. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento. O perfil terá um nome CodeCatalystWorkflowDevelopmentRole-*spaceName*. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).

 Note

Esse perfil só é recomendado para uso com contas de desenvolvedor e usa a política AdministratorAccess gerenciada da AWS, dando a ela acesso total para criar políticas e recursos nessa Conta da AWS.

7. Selecione Criar perfil de desenvolvimento.
8. Na página de conexões, em Perfis do IAM disponíveis para o CodeCatalyst, veja o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` na lista de perfis do IAM adicionados à conta.
9. Para retornar ao espaço, selecione Ir para o Amazon CodeCatalyst.

Como criar e adicionar o CodeCatalyst AWSRoleForCodeCatalystSupport

1. Antes de começar no console do CodeCatalyst, abra o Console de gerenciamento da AWS e, depois, verifique se você está conectado com a mesma Conta da AWS do espaço.
2. Acesse o espaço no CodeCatalyst. Escolha Settings (Configurações) e Contas da AWS.
3. Selecione o link da Conta da AWS na qual você deseja criar o perfil. A página Detalhes da Conta da AWS é exibida.
4. Selecione Gerenciar perfis no Console de gerenciamento da AWS.

A página Adicionar perfil do IAM ao espaço do Amazon CodeCatalyst é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

5. Em Detalhes do espaço do CodeCatalyst, selecione Adicionar perfil de suporte do CodeCatalyst. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento de demonstração. O perfil terá um nome `AWSRoleForCodeCatalystSupport` com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço AWSRoleForCodeCatalystSupport](#).
6. Na página Adicionar perfil para suporte do CodeCatalyst, deixe o padrão selecionado e escolha Criar perfil.
7. Em Perfis do IAM disponíveis para o CodeCatalyst, veja o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` na lista de perfis do IAM adicionados à conta.
8. Para retornar ao espaço, selecione Ir para o Amazon CodeCatalyst.

Depois de criar o ID do builder AWS, criar o primeiro espaço e adicionar uma conta, você poderá criar um projeto. Para obter mais informações, consulte [Como criar um projeto](#). Se esta for a primeira vez que você usa o CodeCatalyst, sugerimos começar com [Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis](#).

Aceitando um convite e criando um AWS Builder ID

Você pode se inscrever na Amazon CodeCatalyst como parte da aceitação de um convite para um projeto ou espaço. Como parte da aceitação do convite, você será solicitado a criar uma ID do AWS construtor. Você usará seu AWS Builder ID para acessar recursos em CodeCatalyst.

Tip

Se precisar de mais ajuda, consulte [Problemas de inscrição](#).

Aqui está um fluxo possível para um usuário que está começando CodeCatalyst com um convite para um projeto ou espaço.

Saanvi Sarkar é uma desenvolvedora que recebeu um convite para participar de um CodeCatalyst projeto como administradora de projetos. Saanvi aceita o convite, que abre a página de login do CodeCatalyst. Ela opta por se inscrever e fornece um endereço de e-mail e uma senha para criar o ID do builder AWS. Saanvi poderá usar seu AWS Builder ID para fazer login em outros CodeCatalyst aplicativos. Posteriormente, ela pode editar o perfil para alterar o endereço de e-mail ou a senha de login. Quando solicitado a escolher um alias, Saanvi especifica SaanviSarkar como o CodeCatalyst alias que será exibido em CodeCatalyst e que outros membros do projeto usarão para @mention Saanvi. Depois de se inscrever, Saanvi também poderá usar suas credenciais de login para outros aplicativos que usam as credenciais do Builder ID. AWS

Ao concluir a inscrição, a Saanvi se junta automaticamente ao CodeCatalyst projeto e ao espaço especificados no convite. O convite também fornece permissões predeterminadas para os perfis no projeto e no espaço. Nas configurações do projeto, o alias de Saanvi aparece na lista de membros com o perfil atribuído ao projeto. Para trabalhar com repositórios de origem CodeCatalyst, a Saanvi dedica um momento para criar um token de acesso pessoal (PAT). O PAT será usado CodeCatalyst para autenticação ao fazer alterações na fonte ou ações que precisem de um token de autenticação.

Quando Saanvi trabalha em um projeto, seu alias será listado no log de atividades de trabalho do projeto. As edições e comentários de Saanvi mostrarão o alias, e outros membros do projeto poderão usá-lo para @mencioná-la nas respostas. Para @mention, outro membro do projeto, Saanvi procura seu alias em seu perfil. CodeCatalyst

Quando ela tem um momento, Saanvi configura seu AWS Builder ID para fazer login CodeCatalyst com autenticação multifator (MFA). Com o MFA configurado, Saanvi pode fazer login CodeCatalyst

usando uma combinação de sua CodeCatalyst senha e uma senha ou token de um aplicativo de autenticação de terceiros aprovado.

Aceitando um convite e criando um AWS Builder ID

Ao ser convidado para um projeto ou espaço na Amazon CodeCatalyst, você receberá um e-mail de `notify@codecatalyst.aws` solicitando que você aceite o convite. Se você já tiver uma ID do AWS Builder e estiver conectado CodeCatalyst, escolher Aceitar convite abrirá automaticamente o projeto ou o espaço em uma guia do navegador. Se você não estiver conectado ao console, mas tiver um AWS Builder ID, você será direcionado para a página de login. Para obter mais informações, consulte [Login com um ID do builder AWS](#).

Se você não tiver um AWS Builder ID, ao escolher Aceitar convite, você será direcionado para a página de login, onde deverá escolher a opção de criar seu AWS Builder ID.

Important

Ao aceitar um convite e abrir repositórios em ambientes de desenvolvimento, os scripts podem ser executados com acesso às suas CodeCatalyst credenciais. Certifique-se de confiar na fonte antes de continuar.

Para aceitar um convite e criar uma ID do AWS construtor

1. No e-mail de convite, selecione Aceitar convite.
2. Na página de login, selecione Não está inscrito? Crie seu ID de AWS construtor.

Tip

Seu AWS Builder ID é uma identidade que você cria para fazer login. Não é o mesmo que uma Conta da AWS.

3. Na página Criar sua ID do AWS Construtor, em Endereço de e-mail, insira o endereço de e-mail que você deseja usar para sua ID do AWS Construtor.

Em Seu nome, forneça o nome e o sobrenome que você deseja exibir nos aplicativos em que você usa seu ID do AWS Builder. Espaços são permitidos. Esse será o nome do seu perfil AWS Builder ID, como Mary Major. Você pode alterar esse nome depois.

Escolha Próximo.

Um código de verificação será enviado para o e-mail que você especificou. Insira este código em Código de verificação e selecione Verificar. Se você não receber o código após cinco minutos e não conseguir encontrá-lo nas pastas de spam ou de lixo eletrônico, selecione Reenviar código.

4. Depois que o código for verificado, insira uma senha que atenda aos requisitos em Senha e Confirmar senha.
5. Escolha Criar ID do AWS Construtor.
6. Na página Criar seu alias, insira um alias que você deseja usar como seu identificador de usuário exclusivo. CodeCatalyst Escolha uma versão abreviada do seu nome sem espaços, como MaryMajor. Outros CodeCatalyst usuários usarão isso para @mention você em comentários e pull requests. Seu CodeCatalyst perfil conterá seu nome completo do seu ID do AWS Construtor e seu CodeCatalyst alias. Você não pode alterar seu CodeCatalyst alias.

Seu nome completo e seu alias serão exibidos em diferentes áreas em CodeCatalyst. Por exemplo, o nome do perfil é exibido para a sua atividade listada no feed de atividades, mas os membros do projeto usarão seu alias para @mencionar você.

Escolha Criar alias. Você será direcionado para o projeto ou o espaço para o qual foi convidado.

Login com um ID do builder AWS

Siga estas etapas para fazer login no seu perfil do Amazon CodeCatalyst.

Note

Você já registrou um dispositivo para autenticação multifator (MFA)? É altamente recomendável configurar a MFA no Amazon CodeCatalyst para aumentar sua segurança. Para obter mais informações, consulte [Como registrar um dispositivo para uso com autenticação multifator](#).

Para fazer login com o ID do builder AWS

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Digite seu Endereço de e-mail. Se desejar, selecione Salvar meu endereço de e-mail se quiser salvar seu endereço de e-mail para futuros logins. Escolha Continuar.

3. Insira a Senha. Escolha Logon. Caso não se lembre da sua senha, siga as etapas em [Esqueci minha senha](#).

Dispositivos confiáveis

Depois que você escolhe a opção Este é um dispositivo confiável na página de login, o Amazon CodeCatalyst considera todos os logins futuros desse dispositivo como autorizados. O Amazon CodeCatalyst não apresentará a opção de inserir um código de MFA enquanto você estiver usando esse dispositivo confiável. Algumas exceções incluem fazer login em um novo navegador ou quando seu dispositivo recebe um endereço IP desconhecido.

Fazer login com SSO

Siga estas etapas para usar o SSO para fazer login no Amazon CodeCatalyst.

Para fazer login com seu ID do builder AWS, consulte [Login com um ID do builder AWS](#).

Como fazer login com SSO

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Em Selecionar uma opção de login, selecione Usar autenticação única (SSO).
3. Em Nome da aplicação do Centro de Identidade da AWS, insira o nome da aplicação fornecida pelo administrador da federação de identidades.
4. Selecione Continuar com o Centro de Identidade do IAM.

Visualização de todos os espaços e projetos de um usuário

Você pode ver uma lista de seus espaços e projetos na página inicial do usuário. A página inicial do usuário mostra uma lista de cada espaço ao qual o usuário pertence, o perfil do usuário nesse espaço, como Administrador do espaço, e os projetos em cada espaço de que o usuário é membro.

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No navegador, digite o seguinte endereço: <https://codecatalyst.aws/home>

's spaces (9) [Manage AWS Builder ID](#) [Create space](#)

EnchantedForest [Create project](#)
Space administrator

Projects (2) < 1 >

WildWaves
Pull requests
Workflows
Source repositories
Environments

FracturedFairyTales
Pull requests
Workflows
Source repositories
Environments

org [Create project](#)
Space administrator

Projects (4) < 1 2 >

migration
Pull requests
Workflows
Source repositories
Environments

test
Pull requests
Workflows
Source repositories
Environments

12597
Pull requests
Workflows
Source repositories
Environments

AnyCompany [Create project](#)
Space member

Projects (1) < 1 >

newproject
Pull requests
Workflows
Source repositories
Environments

- Escolha o espaço ou projeto que você deseja abrir. Se você não encontrar um espaço ou projeto que esperava ver, talvez seja necessário fazer login como um usuário diferente.

Visualização e gerenciamento de perfis do CodeCatalyst

Você pode visualizar perfis de usuário no Amazon CodeCatalyst para ter informações, como endereços de e-mail e aliases do CodeCatalyst. Você também pode atualizar seu perfil e o ID do builder AWS. Se você esquecer sua senha, poderá solicitar uma redefinição de senha.

Visualização dos seus perfis do CodeCatalyst

Forneça informações no cadastro que serão usadas como suas credenciais para fazer login no Amazon CodeCatalyst e que serão gerenciadas em seu perfil. Isso inclui seu Nome, Apelido e o Endereço de e-mail que você usa para entrar no CodeCatalyst.

Note

O Apelido do ID do builder AWS não é seu alias do CodeCatalyst. Você selecionou seu alias CodeCatalyst no cadastro.

Para visualizar seus perfis do CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No canto superior direito, selecione a seta ao lado do ícone com sua primeira inicial e, depois, selecione Minhas configurações. A página Minhas configurações do CodeCatalyst é aberta.
3. Para atualizar seu endereço de e-mail ou senha do ID do builder AWS, ou para configurar a MFA, selecione Gerenciar ID do builder AWS. A página ID do builder AWS é aberta.

Visualização do perfil do CodeCatalyst de outro usuário

Para visualizar o perfil do CodeCatalyst de outro usuário

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No painel de navegação lateral, selecione Configurações do projeto. Escolha a guia Membros. Visualize a lista de membros do seu projeto do CodeCatalyst.
3. Selecione o nome do membro que deseja pesquisar ou @mencione. A página Minhas configurações mostra o alias, o endereço de e-mail e o nome completo do usuário. Use o alias do CodeCatalyst para @mencionar os membros do projeto.

Note

O Apelido do ID do builder AWS de um usuário não é seu alias do CodeCatalyst. Ele selecionou seu alias CodeCatalyst no cadastro.

Para visualizar o perfil de outro usuário em seu projeto, selecione o nome dele na lista.

Atualização de um perfil

No CodeCatalyst, seu perfil consiste em informações pessoais gerenciadas pelo ID do builder AWS e configurações gerenciadas no CodeCatalyst.

- O nome completo, endereço de e-mail e senha do seu perfil são gerenciados pelo ID do builder AWS. Você inseriu essas informações ao se inscrever. Quando você configura a MFA para usar uma aplicação autenticadora para login na aplicação, o CodeCatalyst o direciona para a página ID do builder AWS.
- As configurações do CodeCatalyst para seu token de acesso pessoal (PAT), notificações do CodeCatalyst e preferências de idioma são gerenciadas na página Minhas configurações no CodeCatalyst. Para obter mais informações, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).

Note

Você pode atualizar o nome completo do ID do builder AWS (nome de exibição do CodeCatalyst) e o primeiro nome. No entanto, não é possível alterar o alias do CodeCatalyst.

Atualização de um ID do builder AWS ou endereço de e-mail

Como atualizar o ID do builder AWS ou o endereço de e-mail

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No canto superior direito, selecione a seta ao lado do ícone com sua primeira inicial e, depois, selecione Minhas configurações. A página Minhas configurações do CodeCatalyst é aberta.
3. Na página de perfil, selecione Gerenciar ID do builder AWS. A página ID do builder AWS é aberta.
4. No lado esquerdo da página, selecione Meus detalhes.
5. Em Informações do perfil, selecione Editar para atualizar seu Nome ou Apelido. Se você não especificou um apelido, o campo Apelido será preenchido com o primeiro nome no nome completo. Não é o alias do CodeCatalyst.

 Note

Isso atualiza o nome completo e o primeiro nome do ID do builder AWS. Não é possível atualizar o alias do CodeCatalyst.

Em Informações de contato, selecione Editar para atualizar seu Endereço de e-mail.

 Note

Isso atualiza o endereço de e-mail que você usará para fazer login no CodeCatalyst.

Alteração de uma senha do CodeCatalyst associada a um ID do builder AWS

Use as instruções a seguir para alterar a senha do Amazon CodeCatalyst associada ao ID do builder AWS.

 Note

Se você estiver usando o SSO para entrar no CodeCatalyst, entre em contato com o administrador sobre como alterar sua senha.

Para alterar sua senha do CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No canto superior direito, escolha a seta ao lado do ícone com sua primeira inicial e, depois, selecione Perfil do usuário. A página Minhas configurações do CodeCatalyst é aberta.
3. Na página de perfil, selecione Gerenciar ID do builder AWS. A página ID do builder AWS é aberta.
4. No lado esquerdo da página, selecione Segurança.
5. Selecione Alterar senha e siga as instruções.

Configurando para usar o AWS CLI com CodeCatalyst

O CodeCatalyst console da Amazon é onde você trabalhará na maioria das suas tarefas diárias. No entanto, talvez você queira configurar e configurar o AWS CLI quando estiver trabalhando com ambientes de desenvolvimento, tokens de acesso pessoal ou registros de eventos em CodeCatalyst. Você deve instalar AWS CLI e configurar um perfil antes de poder usá-lo com CodeCatalyst.

Para configurar o AWS CLI para CodeCatalyst

1. Instale a versão mais recente do AWS CLI. Se você já tiver uma versão do AWS CLI instalado, verifique se ela é recente e inclui comandos para CodeCatalyst, e atualize-a se necessário. Para verificar se você tem uma versão instalada que inclui CodeCatalyst comandos, abra um prompt de comando e execute o seguinte comando:

```
aws codecatalyst help
```

Se você ver uma lista de CodeCatalyst comandos, você tem uma versão compatível CodeCatalyst. Se o comando não for reconhecido, atualize sua versão do AWS CLI para a versão mais recente. Para obter mais informações, consulte [Instalando ou atualizando a versão mais recente do AWS CLI](#) no Guia AWS Command Line Interface do Usuário.

2. Execute o `aws configure` comando para criar um perfil se você não tiver um ou se quiser usar um perfil nomeado especificamente para CodeCatalyst. Recomendamos criar um perfil nomeado para usar especificamente CodeCatalyst, mas você também pode usar o perfil padrão. Para obter mais informações, consulte [Noções básicas de configuração](#).
3. Edite o config arquivo do perfil para adicionar uma seção à qual se conectar da CodeCatalyst seguinte maneira. O arquivo config está localizado em `~/.aws/config` no Linux ou MacOS ou em `C:\Users\USERNAME\.aws\config` no Windows.

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

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

4. Salve o arquivo.

5. Antes de tentar executar qualquer CodeCatalyst comando, abra um novo terminal ou prompt de comando e execute o comando a seguir para solicitar e recuperar credenciais para executar comandos. `aws codecatalyst` Substitua `codecatalyst` pelo nome do seu perfil, se necessário.

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

Para ver exemplos de comandos `codecatalyst`, consulte os seguintes tópicos:

- [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#)
- [Acesso a eventos de logs usando o registro de eventos](#)

Tutoriais de conceitos básicos

O Amazon CodeCatalyst fornece vários modelos diferentes para ajudar você a começar a trabalhar com seus projetos. Você também pode optar por começar com um projeto vazio e adicionar recursos a ele. Siga as etapas desses tutoriais para saber mais sobre como trabalhar no CodeCatalyst.

Se esta for a primeira vez que você usa o CodeCatalyst, sugerimos começar com [Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis](#).

Note

Para seguir esses tutoriais, você deve primeiro concluir a configuração. Para obter mais informações, consulte [Configuração e login no CodeCatalyst](#).

Tópicos

- [Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis](#)
- [Tutorial: começo com um projeto vazio e adição manual de recursos](#)
- [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#)

Para ver tutoriais adicionais que se concentram em áreas funcionais específicas no CodeCatalyst, consulte:

- [Conceitos básicos das notificações do Slack](#)
- [Introdução aos repositórios CodeCatalyst de origem e ao blueprint do aplicativo de página única](#)
- [Conceitos básicos de fluxos de trabalho](#)
- [Conceitos básicos de esquemas personalizados](#)
- [Primeiros passos com o guia do desenvolvedor de ações do Amazon CodeCatalyst](#)

Para ver tutoriais detalhados, consulte:

- [Tutorial: Fazer upload de artefatos no Amazon S3](#)
- [Tutorial: Implantar uma aplicação sem servidor](#)

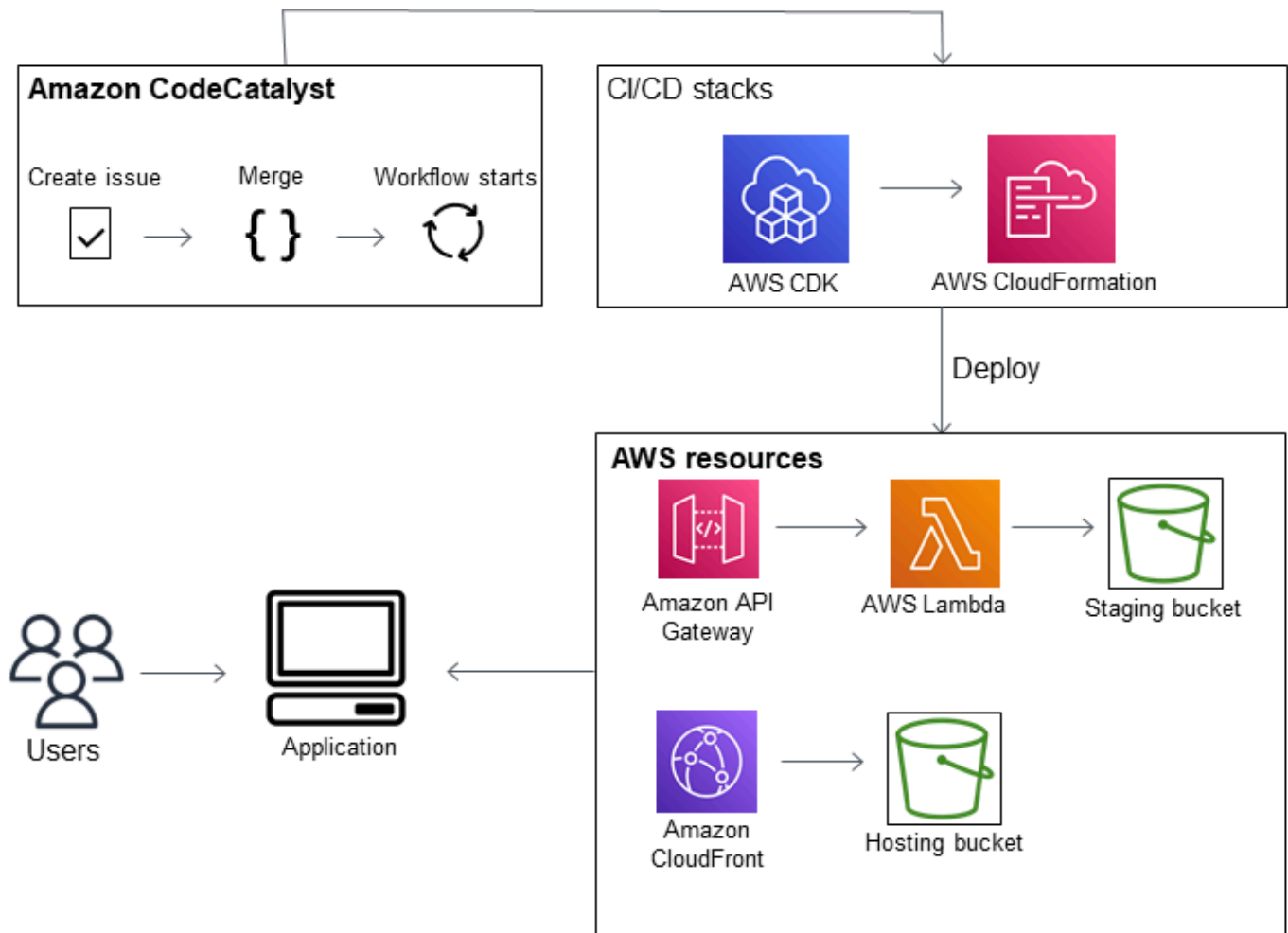
- [Tutorial: Implantar uma aplicação no Amazon ECS](#)
- [Tutorial: Implantar uma aplicação no Amazon EKS](#)
- [Tutorial: código Lint usando uma ação do GitHub](#)
- [Tutorial: Criar e atualizar uma aplicação React](#)

Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis

Você pode começar a desenvolver software mais rapidamente criando um projeto com um esquema. Um projeto criado com um esquema inclui os recursos necessários, incluindo um repositório de origem para gerenciar seu código e um fluxo de trabalho para criar e implantar a aplicação. Neste tutorial, mostraremos como usar o esquema de aplicação web moderna de três níveis para criar um projeto no Amazon CodeCatalyst. O tutorial também inclui visualizar a amostra implantada, convidar outros usuários para trabalhar nela e fazer alterações no código com solicitações pull que são automaticamente criadas e implantadas nos recursos na Conta da AWS conectada quando a solicitação pull é mesclada. Onde o CodeCatalyst cria seu projeto com relatórios, feeds de atividades e outras ferramentas, o esquema cria recursos da AWS da Conta da AWS associada ao projeto. Os arquivos de esquema permitem que você crie e teste um exemplo de aplicação moderna e o implante na infraestrutura do Nuvem AWS.

A ilustração a seguir mostra como as ferramentas no CodeCatalyst são usadas para criar um problema para rastrear, mesclar e criar automaticamente a alteração e, depois, iniciar um fluxo de trabalho no projeto do CodeCatalyst que executa ações para permitir que o AWS CDK e o CloudFormation provisionem sua infraestrutura.

As ações geram recursos na Conta da AWS associada e implantam a aplicação em um perfil do AWS Lambda com tecnologia sem servidor com um endpoint do API Gateway. A ação do AWS Cloud Development Kit (AWS CDK) converte uma ou mais pilhas do AWS CDK em modelos do CloudFormation e implanta pilhas em sua Conta da AWS. Os recursos nas pilhas incluem recursos do Amazon CloudFront para distribuir conteúdo dinâmico da Web, uma instância do Amazon DynamoDB para os dados da aplicação e os perfis e políticas que comportam a aplicação implantada.



Quando você cria um projeto com o esquema de aplicação web moderna de três níveis, seu projeto é criado com os seguintes recursos:

No projeto do CodeCatalyst:

- Um [repositório de origem](#) com código de amostra e YAML de fluxo de trabalho
- Um [fluxo de trabalho](#) que cria e implanta o código de amostra sempre que uma alteração é feita na ramificação padrão
- Um quadro de problemas e um backlog que você pode usar para planejar e monitorar o trabalho
- Um pacote de relatórios de teste com relatórios automatizados incluídos no código de amostra

Na Conta da AWS associada:

- Três pilhas do AWS CloudFormation que criam os recursos necessários para a aplicação.

Para ter detalhes sobre os recursos que serão criados na AWS e no CodeCatalyst como parte deste tutorial, consulte [Referência](#).

 Note

Os recursos e os exemplos incluídos em um projeto dependem do esquema selecionado. O Amazon CodeCatalyst oferece vários esquemas de projetos que definem recursos relacionados à linguagem ou estrutura definida. Para saber mais sobre esquemas, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar o projeto de aplicação web moderna de três níveis](#)
- [Etapa 2: convidar alguém para o projeto](#)
- [Etapa 3: criar problemas para colaborar e monitorar o trabalho](#)
- [Etapa 4: visualizar o repositório de origem](#)
- [Etapa 5: criar um Ambiente de Desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código](#)
- [Etapa 6: visualizar o fluxo de trabalho que cria a aplicação moderna](#)
- [Etapa 7: pedir que outras pessoas revisem suas alterações](#)
- [Etapa 8: encerrar o problema](#)
- [Limpar recursos](#).
- [Referência](#)

Pré-requisitos

Para criar um projeto de aplicação moderna neste tutorial, você deve ter concluído as tarefas em [Configuração e login no CodeCatalyst](#) seguinte forma:

- Tenha um ID do builder AWS para fazer login no CodeCatalyst.
- Você deve pertencer a um espaço e ter o perfil Administrador do espaço ou Usuário avançado atribuído a você nesse espaço. Para ter mais informações, consulte [Criar um espaço](#), [Concessão de permissões de espaço aos usuários](#) e [Perfil de administrador do espaço](#).

- Tenha uma Conta da AWS associada ao seu espaço e tenha o perfil do IAM que você criou durante a inscrição. Por exemplo, durante a inscrição, você tem a opção de escolher criar um perfil de serviço com uma política de perfil chamada CodeCatalystWorkflowDevelopmentRole-**spaceName**. O perfil terá um nome CodeCatalystWorkflowDevelopmentRole-**spaceName** com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-**spaceName**](#). Para as etapas necessárias para criar o perfil, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-**spaceName** para a conta e o espaço](#).

Etapa 1: criar o projeto de aplicação web moderna de três níveis

Depois de criá-lo, no projeto, você desenvolverá e testará o código, coordenará as tarefas de desenvolvimento e visualizará as métricas do projeto. Seu projeto também contém ferramentas e recursos de desenvolvimento.

Neste tutorial, você usará o esquema de aplicação web moderna de três níveis para criar uma aplicação interativa. O fluxo de trabalho criado e executado automaticamente como parte do projeto criará e implantará a aplicação. O fluxo de trabalho só é executado com êxito depois que todos os perfis e informações da conta estiverem configurados para o seu espaço. Depois que o fluxo de trabalho for executado com êxito, acesse o URL do endpoint para ver a aplicação.

Como criar um projeto com um esquema

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço onde você deseja criar um projeto.
3. Escolha Criar projeto.
4. Selecione Começar com um esquema.
5. Na barra de pesquisa, insira **modern**.
6. Selecione o esquema de aplicação web moderna de três níveis e selecione Próximo.
7. Em Nome do projeto, insira um nome para o projeto. Por exemplo:

MyExampleProject.

 Note

O nome deve ser exclusivo no espaço.

8. Em Conta, escolha a Conta da AWS que você adicionou durante a inscrição. O esquema instalará recursos nessa conta.
9. Em Perfil de implantação, escolha o perfil que você adicionou durante a inscrição. Para este exemplo, selecione `CodeCatalystWorkflowDevelopmentRole-spaceName`.

Se não houver perfis listados, adicione um. Para adicionar um perfil, selecione Adicionar perfil do IAM e adicione o perfil à sua Conta da AWS. Para obter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

10. Em Plataforma de computação, selecione Lambda.
11. Em Opção de hospedagem de frontend, selecione Ampliar hospedagem. Para ter informações sobre o AWS Amplify, consulte [O que é hospedagem do AWS Amplify?](#) no Manual do usuário do AWS Amplify.
12. Em Região de implantação, insira o código da Região da AWS em que você deseja que o esquema implante a aplicação Mysfits e os recursos de suporte. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.
13. Em Nome da aplicação, deixe o padrão de `mysfitsstring`.
14. (Opcional) Em Gerar visualização prévia do projeto, selecione Exibir código para visualizar os arquivos de origem que o esquema instalará. Selecione Exibir fluxo de trabalho para visualizar os arquivos de definição do fluxo de trabalho de CI/CD que o esquema instalará. A visualização é atualizada dinamicamente com base nas suas seleções.
15. Escolha Criar projeto.

O fluxo de trabalho do projeto começa assim que você cria o projeto. Levará um pouco de tempo para concluir a criação e a implantação do código. Enquanto isso, vá em frente e convide outra pessoa para o seu projeto.

Etapa 2: convidar alguém para o projeto

Agora que você configurou seu projeto, convide outras pessoas para trabalhar com você.

Para convidar alguém para o projeto

1. Navegue até o projeto para o qual você deseja convidar usuários.
2. No painel de navegação, escolha Configurações do projeto.
3. Na guia Membros, selecione Convidar.
4. Digite os endereços de e-mail das pessoas que você deseja convidar como usuários para o projeto. Você pode digitar vários endereços de e-mail separados por um espaço ou uma vírgula. Você também pode escolher entre membros do seu espaço que não sejam membros do projeto.
5. Escolha o perfil do usuário.

Ao concluir a adição de usuários, selecione Convidar.

Etapa 3: criar problemas para colaborar e monitorar o trabalho

O CodeCatalyst ajuda você a rastrear recursos, tarefas, bugs e qualquer outro trabalho envolvido no projeto com problemas. Você pode criar problemas para monitorar as ideias e o trabalho necessários. Por padrão, quando você cria um problema, ele é adicionado ao backlog. Você pode mover os problemas para um quadro onde é possível acompanhar o trabalho em andamento. Você também pode atribuir um problema a um membro específico do projeto.

Para criar um problema para um projeto

1. No painel de navegação, selecione Problemas.
2. Selecione Criar problema.
3. Em Título do problema, forneça um nome para o problema. Se desejar, forneça uma descrição do problema. Neste exemplo, use **make a change in the src/mysfit_data.json file**.
4. Escolha a prioridade, a estimativa, o status e os rótulos. Em responsável, selecione +Adicionar a mim para atribuir o problema a você mesmo.
5. Selecione Criar problema. O problema agora está visível no quadro. Escolha o cartão para mover o problema para a coluna Em andamento.

Para obter mais informações, consulte [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#).

Etapa 4: visualizar o repositório de origem

O esquema instala um repositório de origem que contém arquivos para definir e comportar a aplicação ou o serviço. Alguns diretórios e arquivos notáveis no repositório de origem são:

- Diretório `.cloud9`: contém arquivos de suporte para o Ambiente de Desenvolvimento do AWS Cloud9.
- Diretório `.codecatalyst`: contém o arquivo YAML de definição do fluxo de trabalho para cada fluxo de trabalho incluído no esquema.
- Diretório `.idea`: contém arquivos de suporte para os Ambientes de Desenvolvimento do JetBrains.
- Diretório `.vscode`: contém arquivos de suporte para o Ambiente de Desenvolvimento do Visual Studio Code.
- Diretório `cdkStacks`: contém os arquivos de pilha do AWS CDK que definem a infraestrutura no Nuvem AWS.
- Diretório `src`: contém o código-fonte da aplicação.
- Diretório `tests`: contém arquivos para os testes integ e de unidade que são executados como parte do fluxo de trabalho automatizado de CI/CD executado quando você cria e testa a aplicação.
- Diretório `web`: contém o código-fonte do frontend. Outros arquivos incluem arquivos de projeto, como o arquivo `package.json` que contém metadados importantes sobre seu projeto, a página `index.html` do site, o arquivo `.eslintrc.cjs` para código de linting e o arquivo `tsconfig.json` para especificar arquivos raiz e opções do compilador.
- Arquivo `Dockerfile`: descreve o contêiner da aplicação.
- Arquivo `README.md`: contém as informações de configuração do projeto.

Como navegar até os repositórios de origem de um projeto

1. Navegue até o projeto e faça o seguinte:
 - Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório.
 - No painel de navegação, selecione Código e, depois, selecione Repositórios de origem. Em Repositórios de origem, selecione o nome do repositório na lista. Você pode filtrar a lista de repositórios digitando parte do nome do repositório na barra de filtro.
2. Na página inicial do repositório, visualize o conteúdo do repositório e as informações sobre os recursos associados, como o número de solicitações pull e fluxos de trabalho. Por padrão, o

conteúdo da ramificação padrão é exibido. Você pode alterar a visualização selecionando uma ramificação diferente na lista suspensa.

Etapa 5: criar um Ambiente de Desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código

Você pode trabalhar no código rapidamente no repositório de origem criando um Ambiente de Desenvolvimento. Para este tutorial, presumimos que você vai:

- Criar um Ambiente de Desenvolvimento AWS Cloud9.
- Escolher a opção de trabalhar em uma nova ramificação fora da ramificação principal ao criar o Ambiente de Desenvolvimento.
- Usar o nome `test` dessa nova ramificação.

Em uma etapa posterior, você usará o Ambiente de Desenvolvimento para fazer uma alteração no código e criar uma solicitação pull.

Como criar um Ambiente de Desenvolvimento com uma nova ramificação

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja criar um Ambiente de Desenvolvimento.
3. Selecione o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, Repositórios de origem e selecione o repositório para o qual você deseja criar um Ambiente de Desenvolvimento.
4. Na página inicial do repositório, selecione Criar um ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Escolha o repositório a ser clonado, selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.
7. Opcionalmente, adicione um alias para o Ambiente de Desenvolvimento.
8. Opcionalmente, escolha o botão de edição da configuração do Ambiente de Desenvolvimento para editar a configuração de computação, armazenamento ou tempo limite do Ambiente de Desenvolvimento.

- Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento. Uma nova guia será aberta com seu ambiente de desenvolvimento no IDE de sua escolha. Você pode editar o código, confirmar e enviar suas alterações.

Nesta seção, você trabalhará com a aplicação de exemplo gerada no CodeCatalyst fazendo alterações no código com solicitações pull que são automaticamente criadas e implantadas nos recursos da Conta da AWS conectada quando a solicitação pull é mesclada.

Como fazer uma alteração no arquivo `src/mysfit_data.json`

- Navegue até o Ambiente de Desenvolvimento do projeto. Em AWS Cloud9, expanda o menu de navegação lateral para navegar pelos arquivos. Expanda `mysfits`, `src` e abra `src/mysfit_data.json`.
- No arquivo, altere o valor do campo `"Age"`: de 6 para 12. A linha deve ser semelhante a esta:

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

- Salve o arquivo.
- Mude para o repositório `mysfits` com o comando `cd /projects/mysfits`.
- Adicione, confirme e envie as alterações com os comandos `git add`, `git commit` e `git push`.


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

Etapa 6: visualizar o fluxo de trabalho que cria a aplicação moderna

Depois de criar o projeto de aplicação moderna, o CodeCatalyst gera vários recursos em seu nome, incluindo um fluxo de trabalho. Um fluxo de trabalho é um procedimento automatizado definido em um arquivo .yaml que descreve como criar, testar e implantar seu código.

Neste tutorial, o CodeCatalyst criou um fluxo de trabalho e o iniciou automaticamente quando você criou seu projeto. (O fluxo de trabalho ainda pode estar em execução, dependendo de há quanto tempo você criou o projeto.) Use os procedimentos a seguir para verificar o progresso do fluxo de trabalho, analisar os logs gerados e os relatórios de teste e, por fim, navegar até o URL da aplicação implantada.

Como conferir o progresso do fluxo de trabalho

1. No console do CodeCatalyst, no painel de navegação, escolha CI/CD e Fluxos de trabalho.

Uma lista de fluxos de trabalho é exibida. Esses são os fluxos de trabalho que o esquema do CodeCatalyst gerou e iniciou quando você criou o projeto.

2. Observe a lista de fluxos de trabalho. Você deve ver quatro:
 - Os dois fluxos de trabalho na parte superior correspondem à ramificação test que você criou anteriormente em [Etapa 5: criar um Ambiente de Desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código](#). Esses fluxos de trabalho são clones dos fluxos de trabalho na ramificação main. O ApplicationDeploymentPipeline não está ativo porque está configurado para uso com a ramificação main. O fluxo de trabalho OnPullRequest não foi executado porque nenhuma solicitação pull foi feita.
 - Os dois fluxos de trabalho na parte inferior correspondem à ramificação main que foi criada quando você executou o esquema anteriormente. O fluxo de trabalho ApplicationDeploymentPipeline está ativo e tem uma execução em andamento (ou concluída).

Note

Se a execução de `ApplicationDeploymentPipeline` falhar com um erro `Build@cdk_bootstrap` ou `DeployBackend`, você talvez tenha executado a aplicação web moderna de três níveis anteriormente e tenha deixado recursos antigos para trás em conflito com o esquema atual. Você precisará excluir esses recursos antigos e executar novamente o fluxo de trabalho. Para obter mais informações, consulte [Limpar recursos](#).

3. Escolha o fluxo de trabalho `ApplicationDeploymentPipeline` associado à ramificação `main`, na parte inferior. Esse fluxo de trabalho foi executado usando o código-fonte na ramificação `main`.

Um diagrama do fluxo de trabalho é exibido. O diagrama mostra vários blocos, cada um representando uma tarefa ou ação. A maioria das ações é organizada na vertical, com as ações na parte superior sendo executadas antes das ações abaixo. As ações organizadas lado a lado são executadas em paralelo. Todas as ações agrupadas devem ser executadas com êxito antes que a ação abaixo delas possa ser iniciada.

Os blocos principais são:

- **WorkflowSource** — Esse bloco representa seu repositório de origem. Ele mostra, entre outras informações, o nome do repositório de origem (`mysfits`) e a confirmação que iniciou automaticamente a execução do fluxo de trabalho. O CodeCatalyst gerou essa confirmação quando você criou seu projeto.
 - **Build** — Esse bloco representa um agrupamento de duas ações que devem ser concluídas para que a próxima ação seja iniciada.
 - **DeployBackend**: esse bloco representa uma ação que implanta os componentes de backend da aplicação na nuvem da AWS.
 - **Tests**: esse bloco representa um agrupamento de duas ações de teste que devem ser concluídas para que a próxima ação seja iniciada.
 - **DeployFrontend**: esse bloco representa uma ação que implanta os componentes de frontend da aplicação na nuvem da AWS.
4. Escolha a guia Definição (na parte superior). O [arquivo de definição do fluxo de trabalho](#) aparece à direita. O arquivo tem as seguintes seções dignas de nota:

- Uma seção `Triggers`, na parte superior. Esta seção indica que o fluxo de trabalho deve começar sempre que o código for enviado para a ramificação `main` do repositório de origem. Os envios para outras ramificações (como `test`) não iniciarão esse fluxo de trabalho. O fluxo de trabalho é executado usando os arquivos na ramificação `main`.
 - Uma seção `Actions`, em `Triggers`. Esta seção define as ações que você vê no diagrama do fluxo de trabalho.
5. Escolha a guia Estado mais recente (na parte superior) e escolha qualquer ação no diagrama do fluxo de trabalho.
 6. À direita, escolha a guia Configuração para ver as configurações usadas pela ação durante a última execução. Cada definição de configuração tem uma propriedade correspondente no arquivo de definição do fluxo de trabalho.
 7. Deixe o console aberto e vá para o próximo procedimento.

Para revisar os logs de compilação e os relatórios de teste

1. Escolha a guia Estado mais recente.
2. No diagrama do fluxo de trabalho, escolha a ação `DeployFrontend`.
3. Aguarde o término da ação. Observe que o ícone “em andamento” () muda para um ícone de “sucesso” (.
4. Escolha a ação `build_backend`.
5. Escolha a guia Logs e expanda algumas seções para ver as mensagens de log dessas etapas. Você pode ver mensagens relacionadas à configuração do backend.
6. Escolha a guia Relatórios e, depois, escolha o relatório `backend-coverage.xml`. O CodeCatalyst exibe o relatório associado. O relatório mostra os testes de cobertura de código que foram executados e indica a proporção de linhas de código que foram validadas com sucesso por meio de testes, como 80%.

Para ter mais informações sobre os relatórios de teste, consulte [Teste com fluxos de trabalho](#).

Tip

Você também pode visualizar os relatórios de teste escolhendo Relatórios no painel de navegação.

7. Deixe o console do CodeCatalyst aberto e vá para o próximo procedimento.

Como confirmar que a aplicação moderna foi implantada

1. Retorne ao fluxo de trabalho ApplicationDeploymentPipeline e escolha o link Run-**string** da última execução.
2. No diagrama do fluxo de trabalho, encontre a ação DeployFrontend e escolha o link Exibir aplicação. O site da Mysfit é exibido.

Note

Se você não vir o link Exibir aplicação dentro da ação DeployFrontend, escolha o link do ID de execução.

3. Procure o Pegasus Mysfit chamado Twilight Glitter. Anote o valor da idade. É 6. Você fará uma alteração no código para atualizar a idade.

Etapa 7: pedir que outras pessoas revisem suas alterações

Agora que você tem alterações em uma ramificação chamada `test`, pode pedir que outras pessoas as revisem criando uma solicitação pull. Execute as etapas a seguir para criar uma solicitação pull para mesclar as alterações da ramificação `test` na ramificação `main`.

Para criar uma solicitação pull

1. Navegue até o projeto.
2. Execute um destes procedimentos:
 - No painel de navegação, escolha Código, Solicitações pull e, depois, selecione Criar solicitação pull.
 - Na página inicial do repositório, escolha Mais e, depois, selecione Criar solicitação pull.
 - Na página do projeto, selecione Criar solicitação pull.

3. No Repositório de origem, certifique-se de que o repositório de origem especificado seja aquele que contém o código confirmado. Essa opção só aparece se você não criou a solicitação pull na página principal do repositório.
4. Em Ramificação de destino, selecione a ramificação na qual mesclar o código depois que ele for revisado.
5. Em Ramificação de origem, selecione a ramificação que contém o código confirmado.
6. Em Título da solicitação pull, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
7. (Opcional) Na Descrição da solicitação pull, forneça informações, como um link para problemas ou uma descrição das alterações.

 Tip

Você pode selecionar Escrever descrição para mim para que o CodeCatalyst gere automaticamente uma descrição das alterações contidas na solicitação pull. Você pode fazer alterações na descrição gerada automaticamente depois de adicioná-la à solicitação pull.

Essa funcionalidade exige que os recursos de IA generativa estejam habilitados para o espaço e não estejam disponíveis para solicitações pull em repositórios vinculados. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

8. (Opcional) Em Problemas, selecione Vincular problemas e, depois, selecione um problema na lista ou insira seu ID. Para desvincular um problema, selecione o ícone de desvinculação.
9. (Opcional) Em Revisores obrigatórios, selecione Adicionar revisores obrigatórios. Selecione na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que a solicitação pull seja mesclada na ramificação de destino.

 Note

Não é possível adicionar um revisor como revisor obrigatório e revisor opcional. Não é possível se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, selecione Adicionar revisores opcionais. Selecione na lista de membros do projeto para adicioná-los. Revisores opcionais não precisam aprovar as alterações como um requisito antes que a solicitação pull seja mesclada na ramificação de destino.

11. Revise as diferenças entre as ramificações. A diferença exibida em uma solicitação pull são as alterações entre a revisão na ramificação de origem e a base de mesclagem, que é a confirmação principal da ramificação de destino no momento em que a solicitação pull foi criada. Se nenhuma alteração for exibida, as ramificações podem ser idênticas ou você pode ter escolhido a mesma ramificação para a origem e o destino.
12. Quando a solicitação pull tiver o código e as alterações que você deseja revisar, selecione Criar.

 Note

Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos, bem como à solicitação pull geral. Você pode adicionar links para recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

Quando você cria a solicitação pull, o fluxo de trabalho OnPullRequest começa a usar os arquivos de origem na ramificação test. Enquanto os revisores aprovam sua alteração de código, você pode observar os resultados escolhendo o fluxo de trabalho e visualizando o resultado do teste.

Depois de revisar a alteração, você pode mesclar o código. A mesclagem do código com a ramificação padrão iniciará automaticamente o fluxo de trabalho que criará e implantará suas alterações.

Para mesclar uma solicitação pull do console do CodeCatalyst

1. Navegue até o seu projeto de aplicação moderna.
2. Na página do projeto, em Abrir solicitações pull, escolha a solicitação pull que você deseja mesclar. Se você não vir a solicitação pull, selecione Exibir tudo e, depois, escolha-a na lista. Escolha Merge (Mesclar).
3. Selecione entre as estratégias de mesclagem disponíveis para a solicitação pull. Também é possível selecionar ou desmarcar a opção de excluir a ramificação de origem depois de mesclar a solicitação pull e, depois, selecionar Mesclar.

 Note

Se o botão Mesclar não estiver ativo ou se a opção Não mesclável for exibida, significa que um ou mais revisores obrigatórios ainda não aprovaram a solicitação pull ou a solicitação pull não pode ser mesclada no console do CodeCatalyst. Um revisor que

não aprovou uma solicitação pull é indicado por um ícone de relógio em Visão geral na área Detalhes da solicitação pull. Se todos os revisores obrigatórios aprovaram a solicitação pull, mas o botão Mesclar ainda não estiver ativo, você pode ter um conflito de mesclagem. Você pode resolver conflitos de mesclagem para a ramificação de destino no console do CodeCatalyst e, depois, mesclar a solicitação pull, ou pode resolver conflitos e mesclar localmente e, depois, enviar a confirmação que contém a mesclagem para o CodeCatalyst. Para ter mais informações, consulte [Mescla de uma solicitação pull \(Git\)](#) e a documentação do Git.

Depois de mesclar as alterações da ramificação `test` na ramificação `main`, a alteração inicia automaticamente o fluxo de trabalho `ApplicationDeploymentPipeline` que cria e implanta sua alteração.

Para ver a confirmação mesclada, execute o fluxo de trabalho `ApplicationDeploymentPipeline`

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Em Fluxos de trabalho, em `ApplicationDeploymentPipeline`, expanda Execuições recentes. Você pode ver a execução do fluxo de trabalho iniciada pela confirmação de mesclagem. Se desejar, escolha-a para observar o progresso da execução.
3. Quando a execução for concluída, recarregue o URL que você visitou anteriormente. Veja o pegasus para verificar se a idade mudou.



Etapa 8: encerrar o problema

Quando um problema é resolvido, ele pode ser fechado no console do CodeCatalyst.

Para fechar um problema para um projeto

1. Navegue até o projeto.
2. No painel de navegação, selecione Problemas.
3. Arraste e solte o problema na coluna Concluído.

Para obter mais informações, consulte [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#).

Limpar recursos.

Limpe no CodeCatalyst e na AWS para remover os traços deste tutorial do seu ambiente.

Você pode optar por continuar usando o projeto usado neste tutorial ou excluir o projeto e os recursos associados.

Note

A exclusão desse projeto excluirá todos os repositórios, problemas e artefatos do projeto para todos os membros.

Para excluir um projeto

1. Navegue até seu projeto e selecione Configurações do projeto.
2. Escolha a guia Geral.
3. No nome do projeto, selecione Excluir projeto.

Para excluir recursos no CloudFormation e no Amazon S3

1. Faça login no Console de gerenciamento da AWS com a mesma conta que você adicionou ao seu espaço do CodeCatalyst.
2. Vá para o serviço CloudFormation.

3. Exclua a pilha mysfits**string**.
4. Exclua a pilha development-mysfits**string**.
5. Escolha (mas não exclua) a pilha CDKToolkit. Escolha a guia Recursos. Escolha o link StagingBucket e exclua o bucket e o conteúdo do bucket no Amazon S3.

 Note

Se você não excluir esse bucket manualmente, poderá ver um erro ao executar novamente o esquema de aplicação web moderna de três níveis.

6. (Opcional) Exclua a pilha CDKToolkit.

Referência

O esquema de aplicação web moderna de três níveis implanta recursos em seu espaço do CodeCatalyst e em sua conta da AWS na nuvem da AWS. Esses recursos são:

- No seu espaço do CodeCatalyst:
 - Um projeto do CodeCatalyst que inclui os seguintes recursos:
 - Um [repositório de origem](#): esse repositório contém código de amostra para uma aplicação web “Mysfits”.
 - Um [fluxo de trabalho](#): esse fluxo de trabalho cria e implanta o código da aplicação Mysfits sempre que uma alteração é feita na ramificação padrão
 - [Quadro de problemas](#) e backlog: esse quadro e o backlog podem ser usados para planejar e acompanhar o trabalho.
 - Um [pacote de relatórios de teste](#): esse pacote inclui relatórios automatizados incluídos no código de amostra.
- Na Conta da AWS associada:
 - Uma pilha CDKToolkit: essa pilha implanta os seguintes recursos:
 - Um bucket de armazenamento do Amazon S3, uma política de bucket e a chave AWS KMS usada para criptografar o bucket.
 - Um perfil de implantação do IAM para a ação de implantação.
 - Perfis e políticas do AWS IAM em apoio aos recursos na pilha.

 Note

O CDKToolkit não é desmontado e recriado para cada implantação. Essa pilha é iniciada em cada conta para apoiar o AWS CDK.

- Uma pilha development-mysfits**string**BackEnd — Essa pilha implanta os seguintes recursos de backend:
 - Um endpoint do Amazon API Gateway.
 - Perfis e políticas do AWS IAM em apoio aos recursos na pilha.
 - Um perfil do AWS Lambda e uma camada fornecem a plataforma de computação tecnologia sem servidor para a aplicação moderna.
 - Uma política e um perfil do IAM para a implantação do bucket e a função do Lambda.
- Uma pilha mysfits**string** — Essa pilha implanta a aplicação de frontend AWS Amplify.

Consulte também

Para ter mais informações sobre os serviços da AWS em que os recursos são criados como parte deste tutorial, consulte o seguinte:

- Amazon S3: um serviço de armazenamento de ativos de frontend em um serviço de armazenamento de objetos oferecendo escalabilidade, disponibilidade de dados, segurança e performance líderes do setor. Para ter mais informações, consulte o [Guia do usuário do Amazon S3](#).
- Amazon API Gateway: serviço para criação, publicação, manutenção, monitoramento e proteção de APIs REST, HTTP e de WebSocket em qualquer escala. Para ter mais informações, consulte o [Guia do desenvolvedor do API Gateway](#).
- Amplify: serviço para hospedar a aplicação de frontend. Para ter mais informações, consulte o [Guia do usuário de hospedagem do AWS Amplify](#).
- AWS Cloud Development Kit (AWS CDK): uma estrutura para definir a infraestrutura de nuvem em código e provisioná-la pelo AWS CloudFormation. O AWS CDK inclui o AWS CDK Toolkit, que é uma ferramenta de linha de comando para interagir com aplicações e pilhas do AWS CDK. Para ter mais informações, consulte o [Guia do desenvolvedor do AWS Cloud Development Kit \(AWS CDK\)](#).

- Amazon DynamoDB: um serviço de banco de dados NoSQL totalmente gerenciado para armazenar dados. Para ter mais informações, consulte o [Guia do desenvolvedor do Amazon DynamoDB](#).
- AWS Lambda — Um serviço para invocar seu código em uma infraestrutura de computação de alta disponibilidade sem provisionar ou gerenciar servidores. Para obter mais informações, consulte o [Guia do desenvolvedor do AWS Lambda](#).
- AWS IAM – Um serviço para controlar o acesso à AWS e aos recursos associados de forma segura. Para obter mais informações, consulte o [Manual do usuário do IAM](#).

Tutorial: começo com um projeto vazio e adição manual de recursos

Você pode criar um projeto vazio sem nenhum recurso predefinido dentro dele escolhendo o esquema Projeto vazio ao criar o projeto. Depois de criar um projeto vazio, você pode criar e adicionar recursos a ele de acordo com as necessidades do projeto. Como os projetos criados sem um esquema ficam vazios na criação, essa opção requer mais conhecimento sobre como criar e configurar recursos do CodeCatalyst para começar.

Tópicos

- [Pré-requisitos](#)
- [Criar um projeto vazio](#)
- [Criar um repositório de origem](#)
- [Criar um fluxo de trabalho para criar, testar e implantar uma alteração de código](#)
- [Convidar alguém para o projeto](#)
- [Criar problemas para colaborar e monitorar o trabalho](#)

Pré-requisitos

Para criar um projeto vazio, você deve ter o perfil de Administrador do espaço ou Usuário avançado. Se esta for a primeira vez que você está se conectando ao CodeCatalyst, consulte [Configuração e login no CodeCatalyst](#).

Criar um projeto vazio

Criar um projeto é o primeiro passo para poder trabalhar em conjunto. Se quiser criar seus próprios recursos, como repositórios de origem e fluxos de trabalho, você pode começar com um projeto vazio.

Para criar um projeto vazio

1. Navegue até o espaço onde você deseja criar um projeto.
2. No painel do espaço, escolha Criar projeto.
3. Escolha Começar do zero.
4. Em Dê um nome ao projeto, insira o nome que você deseja atribuir ao projeto. O nome deve ser exclusivo em seu espaço.
5. Escolha Criar projeto.

Agora que você tem um projeto vazio. A próxima etapa é criar um repositório de origem.

Criar um repositório de origem

Crie um repositório de origem para armazenar e colaborar no código do projeto. Os membros do projeto podem clonar esse repositório nos computadores locais para trabalhar no código. Como alternativa, você pode optar por vincular um repositório hospedado em um serviço compatível, mas isso não será abordado neste tutorial. Para obter mais informações, consulte [Vincular um repositório de origem](#).

Como criar um repositório de origem

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o seu projeto.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha Adicionar repositório e selecione Criar repositório.
5. Em Nomes de repositórios, forneça um nome para o repositório. Neste guia, usamos *codecatalyst-source-repository*, mas você pode escolher um nome diferente. Os nomes dos repositórios devem ser exclusivos em um projeto. Para ter mais informações sobre os requisitos para nomes de repositórios, consulte [Cotas para repositórios de origem em CodeCatalyst](#).

6. (Opcional) Em Descrição, adicione uma descrição para o repositório que ajudará outros usuários no projeto a entender para que o repositório é usado.
7. Selecione Criar repositório (padrão). Essa opção cria um repositório que inclui uma ramificação padrão e um arquivo README.md. Ao contrário de um repositório vazio, você pode usar esse repositório assim que ele for criado.
8. Na Ramificação padrão, deixe o nome como principal, a menos que você tenha um motivo para escolher um nome diferente. Todos os exemplos neste guia usam o nome principal para a ramificação padrão.
9. (Opcional) Adicione um arquivo `.gitignore` para o tipo de código que você planeja enviar.
10. Escolha Criar.

 Note

O CodeCatalyst adiciona um arquivo README.md ao seu repositório quando você o cria. O CodeCatalyst também cria uma confirmação inicial para o repositório em uma ramificação padrão chamada principal. Você pode editar ou excluir o arquivo README.md, mas não pode excluir a ramificação padrão.

Você pode adicionar código rapidamente ao repositório criando um Ambiente de Desenvolvimento. Para este tutorial, recomendamos que você crie um Ambiente de Desenvolvimento usando o AWS Cloud9 e escolha a opção de criar uma ramificação a partir da ramificação principal ao criar o Ambiente de Desenvolvimento. Usamos o nome **test** nessa ramificação, mas você pode inserir um nome de ramificação diferente, se preferir.

Como criar um Ambiente de Desenvolvimento com uma nova ramificação

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja criar um Ambiente de Desenvolvimento.
3. Selecione o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, Repositórios de origem e selecione o repositório para o qual você deseja criar um Ambiente de Desenvolvimento.
4. Na página inicial do repositório, selecione Criar um ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.

6. Escolha o repositório a ser clonado, selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.
7. Opcionalmente, adicione um alias para o Ambiente de Desenvolvimento.
8. Opcionalmente, escolha o botão de edição da configuração do Ambiente de Desenvolvimento para editar a configuração de computação, armazenamento ou tempo limite do Ambiente de Desenvolvimento.
9. Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento. Uma nova guia será aberta com seu ambiente de desenvolvimento no IDE de sua escolha. Você pode editar o código, confirmar e enviar suas alterações.

Criar um fluxo de trabalho para criar, testar e implantar uma alteração de código

No CodeCatalyst, você organiza a criação, o teste e a implantação de aplicações ou serviços em fluxos de trabalho. Os fluxos de trabalho consistem em ações e podem ser configurados para serem executados automaticamente após a ocorrência de eventos específicos do repositório de origem, como envio de código ou abertura ou atualização de uma solicitação pull. Para obter mais informações sobre fluxos de trabalho, consulte [Compilação, teste e implantação com fluxos de trabalho](#).

Siga as instruções descritas em [Conceitos básicos de fluxos de trabalho](#) para criar seu primeiro fluxo de trabalho.

Convidar alguém para o projeto

Agora que você configurou seu projeto personalizado, convide outras pessoas para trabalhar com você.

Para convidar alguém para o projeto

1. Navegue até o projeto para o qual você deseja convidar usuários.
2. No painel de navegação, escolha Configurações do projeto.
3. Na guia Membros, selecione Convidar.

4. Digite os endereços de e-mail das pessoas que você deseja convidar como usuários para o projeto. Você pode digitar vários endereços de e-mail separados por um espaço ou uma vírgula. Você também pode escolher entre membros do seu espaço que não sejam membros do projeto.
5. Escolha o perfil do usuário.

Ao concluir a adição de usuários, selecione Convidar.

Criar problemas para colaborar e monitorar o trabalho

O CodeCatalyst ajuda você a rastrear recursos, tarefas, bugs e qualquer outro trabalho envolvido no projeto com problemas. Você pode criar problemas para monitorar as ideias e o trabalho necessários. Por padrão, quando você cria um problema, ele é adicionado ao backlog. Você pode mover os problemas para um quadro onde é possível acompanhar o trabalho em andamento. Você também pode atribuir um problema a um membro específico do projeto.

Para criar um problema para um projeto

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.

Verifique se você está navegando no projeto em que deseja criar problemas. Para visualizar todos os projetos, no painel de navegação, selecione Amazon CodeCatalyst e, se necessário, selecione Visualizar todos os projetos. Escolha o projeto em que deseja criar ou trabalhar com os problemas.

2. No painel de navegação, selecione Rastrear e, depois, selecione Backlog.
3. Selecione Criar problema.
4. Em Título do problema, forneça um nome para o problema. Se desejar, forneça uma descrição do problema. Escolha o status, a prioridade e a estimativa do problema, se desejar. Você também pode atribuir o problema a um membro do projeto na lista de membros do projeto.

Tip

Você pode optar por atribuir um problema ao Amazon Q para que o Amazon Q tente resolver o problema. Se a tentativa for bem-sucedida, uma solicitação pull será criada e o status do problema mudará para Em análise para que você possa revisar e testar o código. Para obter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#).

Essa funcionalidade exige que recursos de IA generativa estejam habilitados para o espaço. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

5. Escolha Salvar.

Depois de criar problemas, você pode atribuí-los aos membros do projeto, estimá-los e monitorá-los em um quadro Kanban. Para obter mais informações, consulte [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#).

Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento

Se você tem um projeto e um repositório de origem na Amazon CodeCatalyst em um espaço onde os recursos generativos de IA estão habilitados, você pode usar esses recursos para ajudar a acelerar o desenvolvimento de software. Os desenvolvedores geralmente têm mais tarefas a fazer do que tempo para realizá-las. Eles geralmente não se dão ao trabalho de explicar as alterações de código aos colegas de equipe ao criar solicitações pull para análise dessas alterações, esperando que outros usuários achem as alterações autoexplicativas. Os criadores e revisores do pull request também não têm tempo para encontrar e ler todos os comentários em um pull request minuciosamente, especialmente se o pull request tiver várias revisões. CodeCatalyst se integra ao Amazon Q Developer Agent para desenvolvimento de software para fornecer recursos generativos de IA que podem ajudar os membros da equipe a realizar suas tarefas mais rapidamente e aumentar o tempo necessário para se concentrar nas partes mais importantes do trabalho.

O Amazon Q Developer é um assistente conversacional generativo baseado em IA que pode ajudar você a entender, criar, estender e operar aplicativos. AWS Para acelerar sua construção AWS, o modelo que impulsiona o Amazon Q é aprimorado com AWS conteúdo de alta qualidade para produzir respostas mais completas, acionáveis e referenciadas. Para obter mais informações, consulte [O que é Amazon Q Developer?](#) no Guia do usuário do Amazon Q Developer.

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os

controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

Neste tutorial, você aprenderá a usar os recursos generativos de IA CodeCatalyst para ajudá-lo a criar projetos com blueprints, bem como adicionar blueprints a projetos existentes. Além disso, você aprenderá a resumir as alterações entre as ramificações ao criar solicitações pull e a resumir os comentários deixados em uma solicitação pull. Você também aprenderá a criar problemas com suas ideias para alterações ou melhorias no código e atribuí-los ao Amazon Q. Como parte do trabalho com problemas atribuídos ao Amazon Q, você aprenderá como permitir que o Amazon Q sugira tarefas e como atribuir e trabalhar em qualquer tarefa criada como parte do trabalho em um problema.

Tópicos

- [Pré-requisitos](#)
- [Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade](#)
- [Criar um resumo das alterações de código entre as ramificações ao criar uma solicitação pull](#)
- [Criar um resumo dos comentários deixados sobre alterações de código em uma solicitação pull](#)
- [Criar um problema e atribuir ao Amazon Q](#)
- [Criar um problema e ter tarefas recomendadas pelo Amazon Q para ele](#)
- [Limpar os recursos](#)

Pré-requisitos

Para trabalhar com os CodeCatalyst recursos deste tutorial, você deve primeiro ter concluído e ter acesso aos seguintes recursos:

- Você tem um AWS Builder ID ou uma identidade de login único (SSO) para fazer login. CodeCatalyst
- Você está em um espaço que tem recursos de IA generativa ativados. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).
- Você tem o perfil de colaborador ou administrador do projeto em um projeto nesse espaço.
- A menos que você esteja criando um projeto com IA generativa, seu projeto existente tem pelo menos um repositório de origem configurado. Não há suporte a repositórios vinculados.

- Ao atribuir problemas para ter uma solução inicial criada pela IA generativa, o projeto não pode ser configurado com a extensão do software Jira. A extensão não é compatível para esse recurso.

Para ter mais informações, consulte [Criar um espaço](#), [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#), [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Concessão de acesso com perfis de usuário](#).

Este tutorial é baseado em um projeto criado usando o esquema de aplicação web moderna de três níveis com Python. Se você usar um projeto criado com um esquema diferente, ainda poderá seguir as etapas, mas algumas especificidades podem variar, como código de exemplo e linguagem.

Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade

Como desenvolvedor do projeto, você pode colaborar com o Amazon Q, um assistente de IA generativa, ao criar novos projetos ou adicionar componentes a projetos existentes. Você pode fornecer ao Amazon Q os requisitos do projeto interagindo com ele em uma interface semelhante a um bate-papo. Com base em seus requisitos, o Amazon Q sugere um esquema e também descreve os requisitos que não podem ser atendidos. Se o seu espaço tiver esquemas personalizados, o Amazon Q aprenderá e incluirá esses esquemas nas recomendações também. Você pode continuar com a sugestão do Amazon Q se estiver satisfeito, e ele criará os recursos necessários, como um repositório de origem com código para seu requisito. O Amazon Q também cria problemas para requisitos que não podem ser satisfeitos com um esquema. Para saber mais sobre os CodeCatalyst esquemas disponíveis, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#). Para saber mais sobre o uso do Amazon Q com esquemas, consulte [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#).

Para criar um projeto com o Amazon Q

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja criar um blueprint.
3. No painel do espaço, selecione Criar com Amazon Q.
4. No campo de entrada de texto do prompt do Amazon Q, forneça instruções escrevendo uma breve descrição sobre o projeto que você deseja criar. Por exemplo, "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic

and functionality of the application, and a data layer that manages the storage and retrieval of the data.”.

(Opcional) Em Experimentar exemplos, você pode usar um prompt pré-escrito escolhendo um esquema. Por exemplo, se você escolher a aplicação React, o seguinte prompt será fornecido: “I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data. I also want to add authentication and authorization mechanisms for security and allowable actions.”

5. Selecione Enviar para enviar suas instruções para o Amazon Q. O assistente de IA generativa fornece uma sugestão e descreve os requisitos que não podem ser atendidos pelo esquema. Por exemplo, o Amazon Q pode sugerir o seguinte com base em seus critérios:

I recommend using the Modern three-tier web application blueprint based on your requirements. Blueprints are dynamic and can always be updated and edited later.

Modern three-tier web application

By Amazon Web Services

This blueprint creates a Mythical Mysfits 3-tier web application with a modular presentation, application, and data layers.

The application leverages containers, infrastructure as code (IaC), continuous integration and continuous delivery (CI/CD), and serverless code functions.

Version: 0.1.163

[View details](#)

The following requirements could not be met so I will create issues for you.

- Add authentication and authorization mechanisms for security and allowable actions.

6. (Opcional) Para visualizar os detalhes do esquema sugerido, selecione Exibir detalhes.
7. Execute um destes procedimentos:
 - a. Selecione Sim, usar esse esquema se estiver satisfeito com a sugestão.

- b. Selecione Editar prompt se quiser modificar o prompt.
 - c. Selecione Recomeçar se quiser limpar completamente o prompt.
8. Execute um destes procedimentos:
 - a. Selecione Configurar se quiser configurar o esquema sugerido. Você também poderá configurar o esquema mais tarde.
 - b. Selecione Ignorar se você não quiser modificar as configurações do esquema no momento.
9. Se você optar por configurar o esquema, selecione Continuar depois de modificar os recursos do projeto.
10. Quando solicitado, insira o nome que você deseja atribuir ao seu projeto e os nomes dos recursos associados. O nome deve ser exclusivo em seu espaço.
11. Selecione Criar projeto para criar um projeto com o esquema. O Amazon Q cria recursos usando o esquema. Por exemplo, se você criar um projeto com o blueprint de aplicativo de página única, será criado um repositório de origem para códigos e fluxos de trabalho relevantes. CI/CD
12. (Opcional) Por padrão, o Amazon Q também cria problemas para os requisitos que não são atendidos por um esquema. É possível escolher para quais itens você não deseja criar problemas. Depois de optar por permitir que o Amazon Q crie problemas, você também pode atribuir um problema ao Amazon Q. Ele analisará o problema no contexto dos repositórios de origem fornecidos, mostrando um resumo dos arquivos de origem e códigos relevantes. Para obter mais informações, consulte [Descoberta e visualização de problemas](#), [Criar um problema e atribuir ao Amazon Q](#) e [Práticas recomendadas ao criar e trabalhar com problemas atribuídos ao Amazon Q](#).

Depois de criar um projeto com o Amazon Q, você também pode usar o Amazon Q para adicionar novos componentes, pois ele sugere CodeCatalyst esquemas com base em seus requisitos.

Para adicionar um esquema com o Amazon Q

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o projeto em que você deseja adicionar um blueprint.
3. Selecione Adicionar com Amazon Q.
4. No campo de entrada de texto do prompt do Amazon Q, forneça instruções escrevendo uma breve descrição sobre o projeto que você deseja criar. Por exemplo, "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic

and functionality of the application, and a data layer that manages the storage and retrieval of the data.”.

(Opcional) Em Experimentar exemplos, você pode usar um prompt pré-escrito escolhendo um esquema. Por exemplo, se você escolher a aplicação React, o seguinte prompt será fornecido: “I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data. I also want to add authentication and authorization mechanisms for security and allowable actions.”

5. Selecione Enviar para enviar suas instruções para o Amazon Q. O assistente de IA generativa fornece uma sugestão e descreve os requisitos que não podem ser atendidos pelo esquema. Por exemplo, o Amazon Q pode sugerir o seguinte com base em seus critérios:

I recommend using the Single-page application blueprint based on your requirements. Blueprints are dynamic and can always be updated and edited later.

Single-page application

By Amazon Web Services

This blueprint creates a SPA (single-page application) using React, Vue, or Angular frameworks and deploys to AWS Amplify Hosting.

Version: 0.2.15

[View details](#)

The following requirements could not be met so I will create issues for you.

- The application should have reusable UI components
- The application should support for client-side routing
- The application may require server-side rendering for improved performance and SEO

6. (Opcional) Para visualizar os detalhes do esquema sugerido, selecione Exibir detalhes.
7. Execute um destes procedimentos:
 - a. Selecione Sim, usar esse esquema se estiver satisfeito com a sugestão.
 - b. Selecione Editar prompt se quiser modificar o prompt.
 - c. Selecione Recomeçar se quiser limpar completamente o prompt.

8. Execute um destes procedimentos:
 - a. Selecione Configurar se quiser configurar o esquema sugerido. Você também poderá configurar o esquema mais tarde.
 - b. Selecione Ignorar se você não quiser modificar as configurações do esquema no momento.
9. Se você optar por configurar o esquema, selecione Continuar depois de modificar os recursos do projeto.
10. Selecione Adicionar ao projeto para adicionar recursos a um projeto com o esquema. O Amazon Q cria recursos usando o esquema. Por exemplo, se você adicionar a um projeto com o blueprint de aplicativo de página única, será criado um repositório de origem para códigos e fluxos de trabalho relevantes. CI/CD
11. (Opcional) Por padrão, o Amazon Q também cria problemas para os requisitos que não são atendidos por um esquema. É possível escolher para quais itens você não deseja criar problemas. Depois de optar por permitir que o Amazon Q crie problemas, você também pode atribuir um problema ao Amazon Q. Ele analisará o problema no contexto dos repositórios de origem fornecidos, mostrando um resumo dos arquivos de origem e códigos relevantes. Para obter mais informações, consulte [Criar um problema e atribuir ao Amazon Q](#) e [Práticas recomendadas ao criar e trabalhar com problemas atribuídos ao Amazon Q](#).

Criar um resumo das alterações de código entre as ramificações ao criar uma solicitação pull

Uma solicitação pull é a principal maneira para você e outros membros do projeto revisarem, comentarem e mesclarem alterações de código de uma ramificação para outra. Você pode usar solicitações pull para revisar alterações de código de maneira colaborativa em caso de alterações ou correções relativamente pequenas, adições significativas de recursos ou novas versões do seu software lançado. Resumir as alterações no código e a intenção por trás das mudanças como parte da descrição da solicitação pull é útil para outras pessoas que revisarão o código e também ajuda a entender o histórico das alterações no código ao longo do tempo. No entanto, os desenvolvedores geralmente confiam no código para se explicar ou fornecer detalhes ambíguos, em vez de descrever alterações com detalhes suficientes para que os revisores entendam o que estão revisando ou qual era a intenção por trás das alterações no código.

Você pode usar o recurso Escrever descrição para mim ao criar solicitações pull para que o Amazon Q crie uma descrição das alterações contidas em uma solicitação pull. Quando você escolhe essa opção, o Amazon Q analisa as diferenças entre a ramificação de origem que contém as alterações

de código e a ramificação de destino na qual você deseja mesclar essas alterações. Em seguida, ele cria um resumo dessas alterações, bem como sua melhor interpretação da intenção e do efeito dessas alterações.

Note

Esse recurso não funciona com submódulos do Git. Ele não resumirá nenhuma alteração em um submódulo do Git que faz parte da solicitação pull.

Esse recurso não está disponível para solicitações pull em repositórios vinculados.

Você pode experimentar esse recurso com qualquer solicitação pull que criar, mas neste tutorial, vamos testá-lo fazendo algumas alterações simples no código contido em um projeto criado em um esquema de aplicação web moderna de três níveis baseado em Python.

Tip

Se você estiver usando um projeto criado com um esquema diferente ou com seu próprio código, ainda poderá seguir este tutorial, mas os exemplos deste tutorial não corresponderão ao código do seu projeto. Em vez do exemplo sugerido abaixo, faça alterações simples no código do projeto em uma ramificação e, depois, crie uma solicitação pull para testar o recurso, conforme mostrado nas etapas a seguir.

Primeiro, você criará uma ramificação no repositório de origem. Em seguida, você fará uma rápida alteração no código de um arquivo nessa ramificação usando o editor de texto no console. Depois, você criará uma solicitação pull e usará o recurso Escrever descrição para mim para resumir as alterações feitas.

Como criar uma ramificação (console)

1. No CodeCatalyst console, navegue até o projeto em que seu repositório de origem reside.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.
3. Selecione o repositório onde você deseja criar uma ramificação.
4. Na página de visão geral do repositório, selecione Mais e, depois, escolha Criar ramificação.
5. Insira um nome para a ramificação.

6. Selecione uma ramificação a partir da qual criar a ramificação e, depois, selecione Criar.

Depois de ter uma ramificação, edite um arquivo nessa ramificação com uma simples alteração. Neste exemplo, você editará o arquivo `test_endpoint.py` para alterar o número de novas tentativas de testes de **3** para 5.

 Tip

Você também pode optar por criar ou usar um Ambiente de Desenvolvimento para fazer essa alteração no código. Para obter mais informações, consulte [Criar um Ambiente de Desenvolvimento](#).

Para editar o arquivo **`test_endpoint.py`** no console

1. Na página de visão geral do repositório de origem **`mysfits`**, escolha o menu suspenso da ramificação e escolha a ramificação que você criou no procedimento anterior.
2. Em Arquivos, navegue até o arquivo que você deseja editar. Por exemplo, para editar o arquivo `test_endpoint.py`, expanda os testes, expanda integ e escolha `test_endpoint.py`.
3. Escolha Editar.
4. Na linha 7, altere o número de vezes que todos os testes serão repetidos em:

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

para:

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

5. Selecione Confirmar e confirme as alterações na ramificação.

Agora que você tem uma ramificação com uma alteração, pode criar uma solicitação pull.

Crie uma solicitação pull com um resumo das alterações

1. Na página de visão geral do repositório, selecione Mais e, depois, selecione Criar solicitação pull.

2. Em Ramificação de destino, selecione a ramificação na qual mesclar o código depois que ele for revisado.

 Tip

Escolha a ramificação com base na qual você criou a ramificação no procedimento anterior para ver a demonstração mais simples desse recurso. Por exemplo, se você criou sua ramificação a partir da ramificação padrão do repositório, escolha essa ramificação como a ramificação de destino da solicitação pull.

3. Em Ramificação de origem, escolha a ramificação que contém as alterações que você acabou de confirmar no arquivo `test_endpoint.py`.
4. Em Título da solicitação pull, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
5. Em Descrição da solicitação pull, selecione Escrever descrição para mim para que o Amazon Q crie uma descrição das alterações contidas na solicitação pull.
6. Um resumo das alterações é exibido. Revise o texto sugerido e selecione Aceitar e adicionar à descrição.
7. Se desejar, modifique o resumo para refletir melhor as alterações feitas no código. Você também pode optar por adicionar revisores ou vincular problemas a essa solicitação pull. Quando terminar de fazer as alterações adicionais desejadas, selecione Criar.

Criar um resumo dos comentários deixados sobre alterações de código em uma solicitação pull

Quando os usuários revisam uma solicitação pull, eles geralmente deixam vários comentários sobre as alterações nessa solicitação pull. Se houver muitos comentários de muitos revisores, pode ser difícil escolher temas comuns nos comentários ou até mesmo ter certeza de que você revisou todos os comentários em todas as revisões. Você pode usar o recurso Criar resumo de comentários para que o Amazon Q analise todos os comentários deixados sobre alterações de código em uma solicitação pull e crie um resumo desses comentários.

 Note

Os resumos dos comentários são temporários. Se você atualizar uma solicitação pull, o resumo desaparecerá. Os resumos de conteúdo não incluem comentários sobre a solicitação

pull geral, apenas comentários deixados sobre diferenças no código nas revisões da solicitação pull.

Esse recurso não funciona com nenhum comentário deixado sobre alterações de código nos submódulos do Git.

Esse recurso não está disponível para solicitações pull em repositórios vinculados.

Como criar um resumo dos comentários em uma solicitação pull

1. Navegue até a solicitação pull criada no procedimento anterior.

 Tip

Se preferir, você pode usar qualquer solicitação pull aberta em seu projeto. Na barra de navegação, selecione Código, selecione Solicitações pull e escolha qualquer solicitação pull aberta.

2. Adicione alguns comentários à solicitação pull em Alterações se a solicitação pull ainda não tiver comentários.
3. Em Visão geral, selecione Criar resumo de comentários. Quando concluída, a seção Resumo do comentário se expande.
4. Analise o resumo dos comentários deixados sobre as alterações no código nas revisões da solicitação pull e compare-o com os comentários na solicitação pull.

Criar um problema e atribuir ao Amazon Q

As equipes de desenvolvimento criam problemas para monitorar e gerenciar seu trabalho, mas às vezes um problema persiste porque não está claro quem deve trabalhar nele, ou o problema requer pesquisa em uma parte específica da base de código ou outro trabalho urgente deve ser realizado primeiro. CodeCatalyst inclui integração com o Amazon Q Developer Agent para desenvolvimento de software. Você pode atribuir problemas a um assistente de IA generativa chamado Amazon Q, que pode analisar um problema com base no título e na descrição. Se você atribuir o problema ao Amazon Q, ele tentará criar um rascunho de solução para você avaliar. Isso pode ajudar você e sua equipe a se concentrarem e otimizarem o trabalho em questões que exigem sua atenção, enquanto o Amazon Q trabalha em uma solução para problemas que você não tem recursos para resolver imediatamente.

 Tip

O Amazon Q tem melhor desempenho em questões e problemas simples. Para ter melhores resultados, use uma linguagem simples para explicar claramente o que você quer que seja feito.

Quando você atribui um problema ao Amazon Q, CodeCatalyst marcará o problema como bloqueado até que você confirme como deseja que o Amazon Q resolva o problema. Isso exige que você responda a três perguntas antes de continuar:

- Se você deseja confirmar cada etapa necessária ou se deseja prosseguir sem feedback. Se você optar por confirmar cada etapa, poderá responder ao Amazon Q com comentários sobre a abordagem criada para poder iterar a abordagem, se necessário. O Amazon Q também pode revisar os comentários que os usuários deixam sobre qualquer solicitação pull criada se você escolher essa opção. Se você optar por não confirmar cada etapa, o Amazon Q poderá concluir o trabalho mais rapidamente, mas não analisará nenhum comentário fornecido sobre o problema ou em qualquer solicitação pull criada.
- Se você deseja permitir que ele atualize os arquivos do fluxo de trabalho como parte do trabalho. Seu projeto pode ter fluxos de trabalho configurados para iniciar execuções em eventos de solicitação pull. Nesse caso, qualquer solicitação pull criada pela Amazon Q que inclua a criação ou atualização do YAML do fluxo de trabalho pode iniciar uma execução desses fluxos de trabalho incluídos na solicitação pull. Como prática recomendada, não permita que o Amazon Q trabalhe em arquivos de fluxo de trabalho, a menos que tenha certeza de que não há fluxos de trabalho em seu projeto que executem automaticamente esses fluxos de trabalho antes de você revisar e aprovar a solicitação pull criada por ele.
- Se você quer permitir que ele sugira a criação de tarefas para dividir o trabalho no problema em incrementos menores que podem ser atribuídos individualmente aos usuários, incluindo o próprio Amazon Q. Permitir que o Amazon Q sugira e crie tarefas pode ajudar a acelerar o desenvolvimento de problemas complexos, permitindo que várias pessoas trabalhem em partes distintas do problema. Também pode ajudar a reduzir a complexidade de entender todo o trabalho, pois o trabalho necessário para concluir cada tarefa é, de preferência, mais simples do que o problema ao qual ele pertence.
- Em qual repositório de origem você deseja que ele trabalhe. Mesmo que seu projeto tenha vários repositórios de origem, o Amazon Q só pode trabalhar com código em um repositório de origem. Não há suporte a repositórios vinculados.

Depois de fazer e confirmar suas escolhas, o Amazon Q moverá o problema para o estado Em andamento enquanto tenta determinar qual é a solicitação com base no título e na descrição do problema, bem como no código no repositório especificado. Ele criará um comentário fixo onde fornecerá atualizações sobre o status do trabalho. Depois de analisar os dados, a Amazon Q formulará uma possível abordagem para uma solução. O Amazon Q registra as ações atualizando o comentário fixo e comentando sobre o progresso no problema em todas as etapas. Ao contrário dos comentários fixos e das respostas, ele não mantém um registro estritamente cronológico do trabalho. Em vez disso, ele coloca as informações mais relevantes sobre o trabalho no nível superior do comentário fixo. Ele tentará criar código com base na abordagem e na análise do código que já está no repositório. Se gerar uma solução em potencial, ele criará uma ramificação e enviará o código para essa ramificação. Em seguida, ele cria uma solicitação pull que mesclará essa ramificação com a ramificação padrão. Quando o Amazon Q conclui o trabalho, ele move o problema para Em análise para que você e sua equipe saibam que há um código pronto para avaliação.

Note

Esse recurso está disponível somente por meio de Problemas na região Oeste dos EUA (Oregon). Ele não estará disponível se você tiver configurado o projeto para usar o Jira com a extensão Software Jira. Além disso, se você personalizou o layout do quadro, o problema pode não mudar de estado. Para ter melhores resultados, use esse recurso somente com projetos que tenham um layout de quadro padrão.

Esse recurso não funciona com submódulos do Git. Ele não pode fazer alterações em nenhum submódulo do Git incluído no repositório.

Depois de atribuir um problema ao Amazon Q, você não pode alterar o título ou a descrição do problema nem atribuí-lo a outra pessoa. Se você cancelar a atribuição do problema ao Amazon Q, ele concluirá a etapa atual e, depois, deixará de funcionar. Ele não pode retomar o trabalho nem ser reatribuído ao problema depois do cancelamento da atribuição.

Um problema pode ser movido automaticamente para a coluna Em análise se atribuído ao Amazon Q, se um usuário optar por permitir que ele crie tarefas. No entanto, o problema em Em análise ainda pode ter tarefas que estão em um estado diferente, como no estado Em andamento.

Nesta parte do tutorial, você criará três problemas com base em possíveis recursos do código incluído em projetos criados com o esquema de aplicação web moderna de três níveis: um para adicionar ou criar uma criatura mysfit, um para adicionar um recurso de classificação e um para atualizar um fluxo de trabalho para incluir uma ramificação chamada **test**.

 Note

Se você estiver trabalhando em um projeto com código diferente, crie problemas com títulos e descrições relacionados a essa base de código.

Como criar um problema e gerar uma solução para você avaliar

1. No painel de navegação, selecione Problemas e verifique se você está na visualização Quadro.
2. Selecione Criar problema.
3. Dê ao problema um título que explique o que você quer fazer em linguagem simples. Por exemplo, para esse problema, insira o título **Create another mysfit named Quokkapus**. Em Descrição, forneça os seguintes detalhes:

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

Name: Quokkapus

Species: Quokka-Octopus hybrid

Good/Evil: Good

Lawful/Chaotic: Chaotic

Age: 216

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

She's always got a friendly smile on her face, especially when she's using her eight limbs to wrap you up in a great big hug. She exists on a diet of code bugs and caffeine. If you've got some gnarly code that needs assistance, adopt Quokkapus and put her to work - she'll love it! Just make sure you leave enough room for her to grow, and keep that coffee coming.

4. (Opcional) Anexe uma imagem para usar como miniatura e foto do perfil do mysfit ao problema. Se você fizer isso, atualize a descrição para incluir detalhes sobre quais imagens deseja usar e por quê. Por exemplo, você pode adicionar o seguinte à descrição: "O mysfit requer que

arquivos de imagem sejam implantados no site. Adicione as imagens anexadas a este problema ao repositório de origem como parte do trabalho e implante as imagens no site.”

 Note

As imagens anexadas podem ou não ser implantadas no site durante as interações neste tutorial. Você mesmo pode adicionar as imagens ao site e, depois, deixar comentários para que o Amazon Q atualize o código para apontar para as imagens que você deseja usar depois de criar uma solicitação pull.

Analise a descrição e verifique se ela contém todos os detalhes que possam ser necessários antes de prosseguir para a próxima etapa.

5. Em Destinatários, selecione Atribuir ao Amazon Q.
6. Em Repositório de origem, escolha o nome do repositório que contém o código do projeto.
7. Deslize o seletor Exigir que o Amazon Q pare após cada etapa e aguarde a revisão do trabalho até o estado ativo, se necessário.

 Note

Escolher a opção de interromper o Amazon Q após cada etapa permite que você comente sobre o problema ou sobre qualquer tarefa criada para ter a opção de fazer com que o Amazon Q mude de abordagem até três vezes com base em seus comentários. Se você escolher a opção de não deixar o Amazon Q parar após cada etapa para poder revisar o trabalho, o trabalho poderá prosseguir mais rapidamente porque o Amazon Q não está esperando seus comentários, mas você não poderá influenciar a direção que o Amazon Q tomará ao deixar comentários. O Amazon Q também não responderá aos comentários deixados em uma solicitação pull se você escolher essa opção.

8. Deixe o seletor Permitir que o Amazon Q modifique arquivos de fluxo de trabalho no estado inativo.
9. Deslize o seletor Permitir que o Amazon Q sugira a criação de tarefas para o estado ativo.
10. Selecione Criar problema. A visualização muda para o quadro Problemas.

11. Selecione Criar problema para criar outro problema, desta vez com o título **Change the `get_all_mysfits()` API to return mysfits sorted by the Age attribute.** Atribua esse problema ao Amazon Q e crie o problema.
12. Selecione Criar problema para criar outro problema, desta vez com o título **Update the `OnPullRequest` workflow to include a branch named test in its triggers.** Se desejar, vincule ao fluxo de trabalho na descrição. Atribua esse problema ao Amazon Q, mas desta vez garanta que o seletor Permitir que o Amazon Q modifique arquivos de fluxo de trabalho esteja configurado para o estado ativo. Crie o problema para retornar ao quadro Problemas.

 Tip

Você pode procurar arquivos, incluindo arquivos de fluxo de trabalho, inserindo o símbolo de arroba (@) e o nome do arquivo.

Depois de criar e atribuir os problemas, eles passarão para o estado Em andamento. O Amazon Q adicionará comentários de acompanhamento do progresso no problema em um comentário fixo. Se for possível definir uma abordagem para uma solução, ele atualizará a descrição do problema com uma seção Contexto que contém a análise da base de código e uma seção Abordagem que detalha a abordagem proposta para criar uma solução. Se o Amazon Q conseguir encontrar uma solução para o problema descrito, ele criará uma ramificação e alterações de código nessa ramificação que implementarão a solução proposta. Se o código proposto contiver semelhanças com o código-fonte aberto que o Amazon Q conhece, ele fornecerá um arquivo que inclui links para esse código para que você possa analisá-lo. Quando o código estiver pronto, ele criará uma solicitação pull para que você possa revisar as alterações de código sugeridas, adicionará um link para essa solicitação pull para o problema e moverá o problema para Em revisão.

 Important

Você sempre deve revisar todas as alterações de código em uma solicitação pull antes de mesclá-la. A mesclagem de alterações de código feitas pelo Amazon Q, como qualquer outra alteração de código, pode afetar negativamente a base de código e o código de infraestrutura se o código mesclado não for revisado adequadamente e contiver erros ao mesclar.

Para analisar um problema e uma solicitação pull vinculada que contém alterações feitas pelo Amazon Q

1. Em Problemas, escolha um problema atribuído ao Amazon Q que esteja no estado Em andamento. Analise os comentários para monitorar o progresso do Amazon Q. Se estiver presente, analise o contexto e aborde os registros na descrição do problema. Se você optar por permitir que o Amazon Q sugira tarefas, revise todas as tarefas propostas e tome as medidas necessárias. Por exemplo, se o Amazon Q sugeriu tarefas e você gostaria de alterar a ordem ou atribuir tarefas a usuários específicos, escolha Alterar, adicionar ou reordenar tarefas e realize as atualizações necessárias. Quando terminar de visualizar o problema, escolha X para fechar o painel do problema.

 Tip

Para ver o progresso das tarefas, escolha a tarefa na lista de tarefas no problema. As tarefas não aparecem como itens separados no quadro e só podem ser acessadas por meio de um problema. Se uma tarefa for atribuída ao Amazon Q, você deverá abri-la para aprovar qualquer ação que ela queira realizar. Você também deve abrir uma tarefa para ver todas as solicitações pull vinculadas, pois elas não aparecerão como links no problema, somente na tarefa. Para retornar a um problema a partir de uma tarefa, escolha o link para o problema.

2. Agora escolha um problema atribuído ao Amazon Q que esteja no estado Em análise. Analise o contexto e aborde os registros na descrição do problema. Analise os comentários para entender as ações realizadas. Analise todas as tarefas criadas para trabalhar relacionadas a esse problema, incluindo o progresso, quaisquer ações que você talvez precise realizar e quaisquer comentários. Em Solicitações pull, escolha o link para a solicitação pull ao lado do rótulo Abrir para revisar o código.

 Tip

As solicitações pull geradas para tarefas só aparecem como solicitações pull vinculadas na visualização da tarefa. Elas não aparecem como solicitações pull vinculadas para o problema.

3. Na solicitação pull, revise as alterações no código. Para obter mais informações, consulte [Revisão de uma solicitação pull](#). Deixe comentários sobre a solicitação pull se quiser que o

Amazon Q altere qualquer código sugerido. Seja específico ao deixar comentários no Amazon Q para ter melhores resultados.

Por exemplo, ao revisar a solicitação pull criada para **Create another mysfit named Quokkapus**, você pode perceber que há um erro de digitação na descrição. Você pode deixar um comentário para o Amazon Q dizendo “Altere a descrição para corrigir o erro de digitação “needsa” adicionando um espaço entre “needs” e “a”.” Como alternativa, você pode deixar um comentário pedindo ao Amazon Q que atualize a descrição e forneça toda a descrição revisada para que ela seja incorporada.

Se você fez upload de imagens do novo mysfit para o site, pode deixar um comentário para o Amazon Q atualizar o mysfit com ponteiros para a imagem e miniatura para usar no novo mysfit.

 Note

O Amazon Q não responderá a comentários individuais. O Amazon Q só incorporará o feedback deixado nos comentários em solicitações pull se você escolher a opção padrão de parar após cada etapa de aprovação ao criar o problema.

4. (Opcional) Depois que você e outros usuários do projeto deixarem todos os comentários desejados sobre as alterações no código, selecione Criar revisão para que o Amazon Q crie uma revisão da solicitação pull que incorpore as alterações solicitadas nos comentários. O progresso da criação da revisão será relatado pelo Amazon Q em Visão geral, não em Alterações. Atualize seu navegador para ver as atualizações mais recentes do Amazon Q sobre a criação da revisão.

 Note

Somente o usuário que criou o problema pode criar uma revisão da solicitação pull. Você só pode solicitar uma revisão de uma solicitação pull. Verifique se você resolveu todos os problemas com os comentários e se está satisfeito com o conteúdo dos comentários antes de escolher Criar revisão.

5. Um fluxo de trabalho é executado para cada solicitação pull neste projeto de exemplo. Veja um fluxo de trabalho executado com sucesso antes de mesclar a solicitação pull. Você também pode optar por criar fluxos de trabalho e ambientes adicionais para testar o código antes de mesclá-lo. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#).
6. Quando estiver satisfeito com a revisão mais recente da solicitação pull, selecione Mesclar.

Criar um problema e ter tarefas recomendadas pelo Amazon Q para ele

Às vezes, um problema pode conter quantidades de trabalho complexas ou demoradas. CodeCatalyst inclui integração com o Amazon Q Developer Agent para desenvolvimento de software. Você pode pedir ao Amazon Q que analise um problema com base no título e na descrição e recomende uma divisão lógica do trabalho em tarefas separadas. Ele tentará criar uma lista de tarefas recomendadas que podem revisar, modificar e escolher se deseja criar. Isso pode ajudar você e sua equipe a atribuir partes individuais do trabalho aos usuários de maneiras mais gerenciáveis que podem ser realizadas mais rapidamente.

Para criar e revisar uma lista de tarefas recomendadas para um problema

1. No painel de navegação, selecione Problemas e verifique se você está na visualização Quadro.
2. Selecione Criar problema.
3. Dê ao problema um título que explique o que você quer fazer em linguagem simples. Por exemplo, para esse problema, insira o título **Change the `get_all_mysfits()` API to return mysfits sorted by the Good/Evil attribute**. Em Descrição, forneça os seguintes detalhes:

```
Update the API to allow sorting of mysfits by whether they are Good, Neutral, or Evil. Add a button on the website that allows users to quickly choose this sort and to exclude alignments that they don't want to see.
```

4. Analise a descrição e verifique se ela contém todos os detalhes que possam ser necessários antes de prosseguir para a próxima etapa.
5. Em Responsáveis, atribua o problema a você mesmo.
6. Selecione Criar problema. A visualização muda para o quadro Problemas.
7. Escolha o problema que você acabou de criar para abri-lo. Selecione Recomendar tarefas.
8. Escolha o repositório de origem que contém o código do problema. Selecione Começar a recomendar tarefas.

A caixa de diálogo será fechada e o Amazon Q começará a analisar a complexidade do problema. Se o problema for complexo, ele sugerirá uma divisão do trabalho em tarefas sequenciais separadas. Quando a lista estiver pronta, selecione Exibir tarefas recomendadas. Você pode adicionar outras tarefas, modificar as tarefas recomendadas e reordenar as tarefas. Se você concordar com as recomendações, selecionar a opção Criar tarefas criará as tarefas. Você pode atribuir essas tarefas aos usuários para que trabalhem nelas, ou até mesmo ao próprio Amazon Q.

Limpar os recursos

Depois de concluir este tutorial, considere realizar as ações a seguir para limpar todos os recursos que você criou durante este tutorial e que não sejam mais necessários.

- Cancele a atribuição do Amazon Q a qualquer problema que não esteja mais sendo resolvido. Se o Amazon Q tiver concluído o trabalho em um problema ou não conseguir encontrar uma solução, cancele a atribuição do Amazon Q para evitar atingir a cota máxima de recursos de IA generativa. Para ter mais informações, consulte [Gerenciamento de recursos de IA generativa](#) e [Preços](#).
- Mova qualquer problema em que o trabalho esteja concluído para Concluído.
- Se o projeto não for mais necessário, exclua-o.

Organize recursos com espaços em CodeCatalyst

Você cria um espaço que representa você, sua empresa, departamento ou grupo e fornece um local onde suas equipes de desenvolvimento podem gerenciar projetos. Você deve criar um espaço para adicionar projetos, membros e os recursos de nuvem associados que você cria na Amazon CodeCatalyst.

Note

Os nomes dos espaços devem ser exclusivos em todos os lados CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

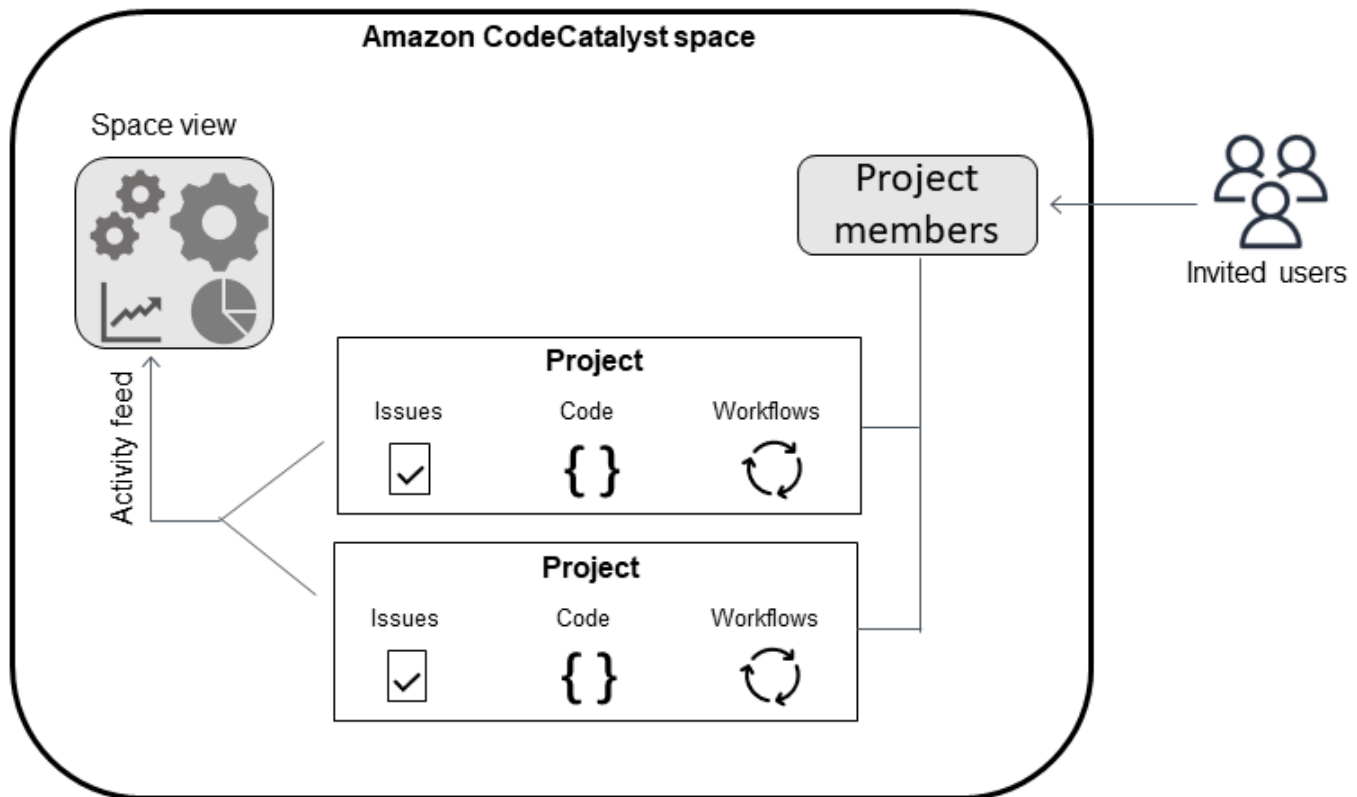
Ao criar um espaço, você recebe automaticamente o perfil de administrador do espaço. Você pode adicionar esse perfil a outros usuários no espaço.

Com o perfil de administrador do espaço, você pode gerenciar o espaço da seguinte forma:

- Adicionar outros administradores de espaço ao espaço
- Alterar perfis e permissões dos membros
- Editar ou excluir o espaço
- Criar projetos e convidar membros para o projeto
- Visualizar uma lista de todos os projetos no espaço
- Visualizar o feed de atividades de todos os projetos no espaço

Ao criar um espaço, você é automaticamente adicionado e ele com dois perfis: o perfil de administrador do espaço e o perfil de administrador do projeto que você criou como parte da criação do espaço. Usuários adicionais são adicionados como membros ao espaço automaticamente quando aceitam convites para projetos. Essa associação ao espaço não concede nenhuma permissão no espaço. O que os usuários podem fazer em um espaço é determinado pelo perfil que o usuário tem em um projeto específico.

Para ter mais informações sobre funções, consulte [Concessão de acesso com perfis de usuário](#).



Veja a seguir considerações adicionais sobre contas adicionadas:

- Contas da AWS adicionado a um CodeCatalyst espaço pode ser usado em qualquer projeto nesse espaço.
- Embora cada ambiente possa oferecer suporte a vários Contas da AWS, você só pode usar uma conta por ambiente em uma ação.
- O faturamento é configurado no nível do espaço. Várias contas podem ser configuradas para cobrança, mas somente uma pode estar ativa em um CodeCatalyst espaço. Uma Conta da AWS pode ser usada como uma conta de cobrança para mais de um espaço em CodeCatalyst. O Conta da AWS que é especificado como a conta de cobrança do seu CodeCatalyst espaço tem cotas diferentes de outras conexões de conta para um espaço. Para obter mais informações, consulte [Cotas para o CodeCatalyst](#).
- Depois de criar uma conexão, você deve adicionar funções AWS do IAM à sua conexão se seu fluxo de trabalho precisar acessar essas funções do IAM com seu CodeCatalyst ambiente. Para ter mais informações sobre como os ambientes são usados, consulte [Implantação em e Contas da AWS VPCs](#).

Tópicos

- [Criar um espaço](#)
- [Editar um espaço](#)
- [Excluir um espaço](#)
- [Monitoramento de atividades para usuários e recursos em um espaço](#)
- [Permitindo acesso a AWS recursos com conexão Contas da AWS](#)
- [Configuração de perfis do IAM para contas conectadas](#)
- [Concessão de permissões de espaço aos usuários](#)
- [Permitir o acesso ao espaço usando equipes](#)
- [Permissão do acesso ao espaço para recursos da máquina](#)
- [Administração de Ambientes de Desenvolvimento para um espaço](#)
- [Cotas para espaços](#)

Criar um espaço

Quando você se inscreve pela primeira vez na Amazon CodeCatalyst com seu ID de AWS construtor, é necessário criar um espaço. Para obter mais informações, consulte [Configuração e login no CodeCatalyst](#). Você pode optar por criar espaços adicionais para atender às suas necessidades de negócios.

Note

Os nomes dos espaços devem ser exclusivos em todos os lados CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

As informações neste guia são fornecidas para criar espaços CodeCatalyst que ofereçam suporte aos usuários do AWS Builder ID. As etapas para configurar e administrar um espaço que ofereça suporte à federação de identidades são fornecidas no Guia do CodeCatalyst Administrador. Para trabalhar com espaços configurados para federação de identidades, consulte [Configuração e administração de CodeCatalyst espaços](#) no Amazon CodeCatalyst Administrator Guide.

Para criar espaços adicionais que ofereçam suporte aos usuários do AWS Builder ID, você deve ter a função de administrador do Space.

Note

Ao criar um espaço adicional, não é solicitado que você crie um projeto. Para saber como criar projetos em um espaço, consulte [Como criar um projeto](#).

Para criar outro espaço

1. No Console de gerenciamento da AWS, verifique se você está conectado com o mesmo Conta da AWS que deseja associar ao seu CodeCatalyst espaço.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até o seu espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

4. Selecione Criar espaço.
5. Na página Criar um espaço, em Nome do espaço, insira um nome para o espaço. Não é possível alterar essa opção mais tarde.

Note

Os nomes dos espaços devem ser exclusivos em todos os lados CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

6. Em Região da AWS, selecione a região onde você deseja armazenar o espaço e os dados do projeto. Não é possível alterar essa opção mais tarde.
7. Em ID da Conta da AWS , insira o ID de 12 dígitos da conta que você deseja conectar ao seu espaço.

Em Token de verificação da conta da AWS , copie o ID do token gerado. O token é copiado automaticamente para você, mas talvez você queira armazená-lo enquanto aprova a solicitação de AWS conexão.

8. Escolha Verificar em AWS.

9. A página Verify Amazon CodeCatalyst Space é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

No Console de gerenciamento da AWS, certifique-se de escolher o mesmo Região da AWS local em que deseja criar seu espaço.

Para acessar diretamente a página, faça login no Amazon CodeCatalyst Spaces em Console de gerenciamento da AWS em <https://console.aws.amazon.com/codecatalyst/home/>.

O token de verificação é inserido automaticamente no Token de verificação. Um banner de sucesso mostra uma mensagem de que o token é válido.

10. Escolha Verificar espaço.

Uma mensagem de sucesso da conta verificada é exibida para mostrar que a conta foi adicionada ao espaço.

11. Permaneça na página Verify Amazon CodeCatalyst Space. Selecione o seguinte link: Para adicionar perfis do IAM para esse espaço, visualizar os detalhes do espaço.

A página de detalhes do CodeCatalyst espaço é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

12. Em Funções do IAM disponíveis para CodeCatalyst, escolha Adicionar função do IAM.

A CodeCatalyst página Adicionar funções do IAM disponíveis para exibição.

13. Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento.

A função de desenvolvedor é uma função AWS do IAM que permite que seus CodeCatalyst fluxos de trabalho acessem AWS recursos como Amazon S3, Lambda e CloudFormation

O perfil terá um nome CodeCatalystWorkflowDevelopmentRole-*spaceName* com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).

14. Selecione Criar perfil de desenvolvimento.
15. Na página de conexão, em Funções do IAM disponíveis para CodeCatalyst, veja a função de desenvolvedor na lista de funções do IAM adicionadas à sua conta.

16. Escolha Go to Amazon CodeCatalyst.
17. Na página de criação em CodeCatalyst, escolha Criar espaço.

Editar um espaço

Você pode alterar a descrição de um espaço para ajudar os usuários a entenderem para o que ele é usado.

Você deve ter o perfil de administrador do espaço para editar os detalhes do espaço.

As informações neste guia são fornecidas para edição de espaços CodeCatalyst que oferecem suporte aos usuários do AWS Builder ID. Para saber mais sobre as etapas para configurar e administrar um espaço que ofereça suporte à federação de identidades, consulte [Configuração e administração de CodeCatalyst espaços](#) no Amazon CodeCatalyst Administrator Guide.

Para editar a descrição de um espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Na guia Configurações do espaço, selecione Editar. Faça as alterações desejadas na descrição do espaço e selecione Salvar.

Excluir um espaço

Você pode excluir um espaço para remover o acesso a todos os recursos do espaço. Você deve ter o perfil de administrador do espaço para excluir um espaço.

Note

Não é possível desfazer uma exclusão de espaço.

Depois de excluir um espaço, nenhum membro do espaço poderá acessar os recursos do espaço. O faturamento por recursos do espaço também será interrompido e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

 Note

Os nomes dos espaços devem ser exclusivos em todos os lados CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

As informações neste guia são fornecidas para excluir espaços CodeCatalyst que oferecem suporte aos usuários do AWS Builder ID. Para saber mais sobre as etapas para configurar e administrar um espaço que ofereça suporte à federação de identidades, consulte [Configuração e administração de CodeCatalyst espaços](#) no Amazon CodeCatalyst Administrator Guide.

Como excluir um espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, Excluir.
4. Digite **delete** para confirmar a exclusão.
5. Escolha Excluir.

 Note

Se você pertencer a mais de um espaço, será redirecionado para a página de visão geral do espaço. Se você pertencer a um espaço, será redirecionado para a página de criação do espaço.

Monitoramento de atividades para usuários e recursos em um espaço

Para ver projetos criados recentemente e atualizações de status, você pode usar o CodeCatalyst console para ver um feed de atividades que mostra atualizações para recursos espaciais.

No feed de atividades, você pode visualizar métricas, como falhas na execução do fluxo de trabalho e projetos criados.

Para visualizar a atividade em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Escolha Atividade.
4. Visualize as informações em Atividade.
5. Para filtrar por atividade, escolha o seletor no canto superior direito.
6. Para visualizar todas as atividades em seu espaço, selecione Qualquer tipo de atividade.

Permitindo acesso a AWS recursos com conexão Contas da AWS

Você pode usar recursos de seus CodeCatalyst espaços Contas da AWS na Amazon. Para fazer isso, você deve configurar uma conexão entre o Contas da AWS e seu espaço em CodeCatalyst. Criar uma conexão como essa significa que projetos e fluxos de trabalho em seu CodeCatalyst espaço podem interagir com recursos em seu Contas da AWS. Você deve criar uma conexão para cada uma que Conta da AWS deseja usar com seu CodeCatalyst espaço.

Depois de criar uma conexão, você pode optar por associar funções AWS do IAM a ela.

Tópicos

- [Adicionando um Conta da AWS a um espaço](#)
- [Adicionar perfis do IAM às conexões da conta](#)

- [Adição da conexão da conta e dos perfis do IAM ao seu ambiente de implantação](#)
- [Visualização de conexões de conta](#)
- [Excluindo conexões de conta \(em CodeCatalyst\)](#)
- [Configurar uma conta de faturamento para um espaço](#)

Você pode configurar CodeCatalyst o uso autorizado Contas da AWS adicionando as contas ao seu espaço. Ao adicionar Contas da AWS ao seu CodeCatalyst espaço, você pode dar aos fluxos de trabalho do projeto acesso aos Conta da AWS recursos e à sua configuração de faturamento.

Adicionar um Conta da AWS cria uma conexão que CodeCatalyst autoriza o uso dessa conta. Você pode usar adicionado Contas da AWS para fazer o seguinte:

- Configure o faturamento de um CodeCatalyst espaço. Consulte [Gerenciamento de faturamento](#) no Guia do CodeCatalyst Administrador da Amazon. O Conta da AWS que é especificado como a conta de cobrança do seu CodeCatalyst espaço tem cotas diferentes de outras conexões de conta para um espaço. Para obter mais informações, consulte [Cotas para o CodeCatalyst](#).
- Permita CodeCatalyst que você assuma funções do IAM para acessar AWS recursos e implantá-los Serviços da AWS na conta. Consulte [Configuração de perfis do IAM para contas conectadas](#).

As conexões da conta são criadas ao concluir a autorização com a Conta da AWS. Depois que a conexão é criada, você configura ainda mais a conexão para fluxos de trabalho e projetos a serem usados adicionando perfis do IAM.

Para ver as etapas para configurar conexões de conta na Console de gerenciamento da AWS página CodeCatalyst como administrador do Conta da AWS e do espaço, consulte [Gerenciando contas conectadas](#) no Guia do CodeCatalyst administrador. As conexões da conta podem ser configuradas para restringir projetos específicos. Você só pode associar fluxos de trabalho ou conexões de VPC a Conta da AWS uma que tenha acesso ao seu projeto. Para ter mais informações, consulte [Configuração de conexões de contas restritas ao projeto](#).

Adicionando um Conta da AWS a um espaço

Você usa o CodeCatalyst console e o Console de gerenciamento da AWS para conectar seu espaço a um Conta da AWS.

Antes de adicionar um Conta da AWS a um espaço em CodeCatalyst, preencha os seguintes pré-requisitos:

- Crie um Conta da AWS e adquira permissões para criar funções AWS do IAM na conta que você deseja conectar.
- Crie os perfis do IAM que você deseja associar à conexão da sua conta, incluindo as políticas do IAM com permissões para os perfis.
- Adquira a função de administrador do CodeCatalyst Space no espaço em que você deseja criar a conexão.

Tópicos

- [Etapa 1: criação de uma solicitação de conexão](#)
- [Etapa 2: aceitar uma solicitação de conexão de conta](#)
- [Etapa 3: revisar uma conexão aprovada](#)
- [Etapa 4: adicionar perfis do IAM à conexão](#)
- [Próximas etapas: criar perfis adicionais do IAM para a conexão da sua conta](#)

Etapa 1: criação de uma solicitação de conexão

A criação de uma solicitação de conexão no CodeCatalyst console gera um token de conexão que você pode usar para concluir a autorização.

Você deve ter a função de administrador do espaço ou usuário avançado no CodeCatalyst espaço em que deseja criar a conexão. Você também deve ter permissões administrativas para a Conta da AWS que deseja adicionar.

Para criar uma conexão

1. No Console de gerenciamento da AWS, verifique se você está logado com a mesma conta com a qual deseja criar uma conexão.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
4. Escolha Adicionar um Conta da AWS.
5. Na CodeCatalyst página Conta da AWS Associar à Amazon, em Conta da AWS ID, insira o ID de doze dígitos da conta que você deseja conectar ao seu espaço. Para obter informações sobre como encontrar seu Conta da AWS ID, consulte [Seu Conta da AWS ID e seu alias](#).
6. Em Nome de CodeCatalyst exibição da Amazon, insira um nome de referência para a conta.

7. (Opcional) Em Descrição da conexão, insira uma descrição para a conta que ajudará você a escolher os projetos aos quais a conta e os perfis serão aplicados.
8. Selecione Associar Conta da AWS.
9. A página retorna a Detalhes da Conta da AWS , onde um banner de sucesso é exibido.

Etapa 2: aceitar uma solicitação de conexão de conta

Depois de enviar uma solicitação no CodeCatalyst console para se conectar ao seu Conta da AWS, trabalhe com o AWS administrador para aceitar a solicitação de conexão enviando-a com o token de conexão fornecido.

Verifique se você tem permissões de administrador para sua conta e se está conectado Console de gerenciamento da AWS com a mesma Conta da AWS para a qual está criando a conexão.

Para aprovar uma solicitação de conexão (console)

1. No Console de gerenciamento da AWS, verifique se você está logado com a mesma conta com a qual deseja criar uma conexão.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
4. Na página Detalhes da Conta da AWS , selecione Configuração completa no Console de gerenciamento da AWS.
5. A página Verify Amazon CodeCatalyst Space é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

Para acessar diretamente a página, faça login no Amazon CodeCatalyst Spaces em Console de gerenciamento da AWS em <https://console.aws.amazon.com/codecatalyst/home/>.

O token de verificação é inserido automaticamente no Token de verificação. Uma mensagem de sucesso mostra uma mensagem de que o token é válido.

6. (Opcional) Em Níveis pagos autorizados, selecione Autorizar níveis pagos (Standard, Enterprise) para ativar os níveis pagos na conta de faturamento.

 Note

Isso não eleva o nível de faturamento para um nível pago. No entanto, isso configura o Conta da AWS para que você possa alterar o nível de cobrança do seu espaço a qualquer momento. CodeCatalyst É possível ativar os níveis pagos a qualquer momento. Sem essa alteração, o espaço só pode usar o nível gratuito.

7. Escolha Verificar espaço.

Uma mensagem de sucesso da conta verificada é exibida para mostrar que a conta foi adicionada ao espaço.

Etapa 3: revisar uma conexão aprovada

Depois de ter a aprovação de uma conexão, você pode ver a conexão no console, junto com os perfis do IAM adicionados a ela.

Como revisar uma conexão aprovada

1. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
2. A conexão da conta é listada com a data em que foi criada.
3. Escolha o nome de exibição da conta. A página Detalhes da Conta da AWS é exibida.

Etapa 4: adicionar perfis do IAM à conexão

Se você estiver usando uma função do IAM configurada para uma ação de CodeCatalyst implantação, adicione a função ao seu ambiente de implantação. Para obter mais informações, consulte [Adicionar perfis do IAM às conexões da conta](#).

Próximas etapas: criar perfis adicionais do IAM para a conexão da sua conta

Depois de criar uma conexão, você pode criar perfis do IAM adicionais para incluir. Os perfis do IAM adicionados dependem dos seus fluxos de trabalho. Por exemplo, uma ação de CodeCatalyst construção requer a função de CodeCatalyst construção.

Para conectar a conta, você precisará do nome do recurso da Amazon (ARN) para os perfis criados. Copie o ARN dos perfis conforme detalhado aqui. Para obter mais informações sobre como trabalhar com ARNs funções do IAM, consulte [Amazon Resource Name \(ARN\)](#).

Como acessar o ARN do perfil do IAM

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Perfis.
3. Na caixa de pesquisa, insira o nome do perfil que você deseja adicionar.
4. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

5. Na parte superior, copie o valor do ARN do perfil.

Adicionar perfis do IAM às conexões da conta

Parte da criação da conexão da sua conta inclui adicionar a função ou funções do IAM que você deseja usar com projetos em seu CodeCatalyst espaço.

Note

Para usar funções do IAM com uma conexão de conta, certifique-se de que a política de confiança esteja atualizada para usar o CodeCatalyst serviço principal.

Adicionar perfis do IAM a uma conexão de conta (console)

1. No Console de gerenciamento da AWS, verifique se você está logado com a mesma conta que deseja gerenciar.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
4. Escolha o nome de CodeCatalyst exibição da Amazon da conexão da sua conta e, em seguida, escolha Gerenciar funções em Console de gerenciamento da AWS.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é exibida.

5. Execute um destes procedimentos:
 - Para criar uma função de serviço que contenha a política de permissões e a política de confiança para a função de desenvolvedor, escolha Criar função de administrador de CodeCatalyst desenvolvimento no IAM. O perfil terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName` com um identificador exclusivo

anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).

Selecione Criar perfil de desenvolvimento.

- Para adicionar um perfil que já foi criado no IAM, selecione Adicionar um perfil do IAM existente. Em Selecionar perfil do IAM existente, escolha o perfil na lista suspensa.

Escolha Add role (adicionar função).

A página é aberta no Console de gerenciamento da AWS. Talvez seja necessário fazer login para acessá-la.

6. No painel de navegação da página Amazon CodeCatalyst Spaces, escolha Spaces.

Para acessar diretamente a página, faça login no Amazon CodeCatalyst Spaces Console de gerenciamento da AWS em <https://console.aws.amazon.com/codecatalyst/home/>.

7. Escolha a conta adicionada ao seu CodeCatalyst espaço. A página de conexão é exibida.
8. Na página de conexão, em Funções do IAM disponíveis para CodeCatalyst, veja a lista de funções do IAM adicionadas à sua conta. Escolha Associar função do IAM CodeCatalyst a.
9. No pop-up Associar uma função do IAM, em ARN da função, insira o Amazon Resource Name (ARN) da função do IAM que você deseja associar ao seu espaço. CodeCatalyst

Em Propósito, escolha uma finalidade do perfil que descreva como você deseja usar o perfil na conexão da sua conta. Especifique RUNNER para os perfis que você usa para executar ações em fluxos de trabalho. Especifique SERVICE para os perfis que você usa para acessar outro serviço.

Você pode especificar mais de uma finalidade.

 Note

É necessário escolher uma finalidade para o ARN do perfil.

10. Selecione Associar um perfil do IAM. Repita essas etapas para perfis adicionais do IAM.

Adição da conexão da conta e dos perfis do IAM ao seu ambiente de implantação

Para acessar AWS recursos, como o Amazon ECS ou AWS Lambda recursos para implantações, as ações de CodeCatalyst criação e implantação exigem funções do IAM com permissões para acessar esses recursos. Com a função de administrador de espaço ou usuário avançado, você pode conectar sua CodeCatalyst conta ao Conta da AWS local onde seus recursos são criados. Depois, adicione o perfil do IAM à conexão da sua conta. Para implantar ações, você deve então adicionar a função do IAM a um CodeCatalyst ambiente.

Você deve adicionar os perfis do IAM que deseja usar com ambientes de implantação em seus projetos. Adicionar os perfis à conexão da conta não adiciona os perfis e a conexão aos ambientes de implantação do projeto. Para adicionar sua conexão de conta e perfis do IAM ao seu ambiente de implantação, certifique-se de que a conexão da conta e os perfis sejam criados conforme detalhado em [Etapa 4: adicionar perfis do IAM à conexão](#).

Em seguida, use a página Ambientes no CodeCatalyst console para adicionar sua conexão de conta e função do IAM a um ambiente de implantação em um projeto.

Note

Você só adiciona uma função do IAM a um ambiente se a função do IAM for usada para uma CodeCatalyst ação que exija uma função do IAM. Todas as ações de fluxo de trabalho que exigem funções do IAM, incluindo ações de criação, devem usar um CodeCatalyst ambiente.

Para adicionar a conexão da conta e os perfis do IAM ao seu ambiente de implantação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto com o ambiente de implantação em que você deseja adicionar a conexão da conta e os perfis do IAM.
3. Expanda CI/CD e selecione Ambientes.
4. Escolha seu ambiente e, depois, as guias adicionais serão exibidas.
5. Escolha a guia Conexões de Conta da AWS . Em Nome da conexão, as contas que foram adicionadas ao ambiente, se houver, são listadas.
6. Selecione Associar Conta da AWS. A página Associar Conta da AWS a <environment_name> é exibida.

7. Em Conexão, escolha o nome da conexão da conta com os perfis do IAM que você deseja adicionar. Selecione Associar .

Visualização de conexões de conta

Você pode ver uma lista das conexões e ver detalhes sobre cada conexão.

Você deve ter o perfil de Administrador do espaço ou Usuário avançado para gerenciar conexões para seu espaço.

Para ver todas as conexões de um CodeCatalyst espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com a conexão de conta que você quer exibir.
3. Selecione a guia Contas da AWS .
4. Em Contas da AWS , veja a lista de conexões de conta do espaço, incluindo o ID da conta e o status de cada conexão.

Para ver os detalhes da conexão da conta

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
3. No nome de CodeCatalyst exibição da Amazon, escolha o nome da conexão. Na página Detalhes, veja a lista de perfis do IAM associados à conexão junto com outros detalhes.

Excluindo conexões de conta (em CodeCatalyst)

Você pode excluir uma conexão de conta caso não precise mais dela. Para esse procedimento, você usará CodeCatalyst para excluir uma conexão de conta que você adicionou anteriormente ao seu espaço. Isso exclui a conexão da conta do seu espaço, desde que a conta não seja a conta de faturamento do espaço.

 Important

Depois que uma conexão de conta for excluída, você não poderá reconectá-la. Você deve criar uma conexão de conta e, depois, associar perfis e ambientes do IAM ou configurar o faturamento, conforme necessário.

Uma conta de cobrança deve ser designada para seu CodeCatalyst espaço, mesmo que o uso do espaço não exceda o nível gratuito. Antes de remover um espaço de uma conta que é uma conta de faturamento designada, você precisará adicionar outra conta ao seu espaço. Consulte [Gerenciamento de faturamento](#) no Guia do CodeCatalyst Administrador da Amazon.

 Important

Embora você possa usar essas etapas para remover uma conta, isso não é recomendado. A conta também pode ser configurada para oferecer suporte aos fluxos de trabalho em CodeCatalyst.

Para gerenciar conexões de conta para seu espaço, você deve ter o perfil de Administrador do espaço ou Usuário avançado.

Uma conta que foi removida pode ser adicionada novamente mais tarde, mas você deve criar uma conexão entre a conta e o espaço. Será necessário reassociar todos os perfis do IAM à conta adicionada.

Como excluir uma conexão de conta

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
3. Em Nome de CodeCatalyst exibição da Amazon, escolha o seletor ao lado da conexão da conta que você deseja remover.
4. Escolha Remover Conta da AWS. Confirme a exclusão inserindo o nome no campo e selecione Remover.

Será exibido um banner de sucesso e a conexão da conta é removida da lista de conexões.

Configurar uma conta de faturamento para um espaço

Uma conta de cobrança deve ser designada para seu CodeCatalyst espaço, mesmo que o uso do espaço não exceda o nível gratuito.

Para configurar uma conta de faturamento, consulte [Faturamento](#) no Guia do CodeCatalyst administrador. O Conta da AWS que é especificado como a conta de cobrança do seu CodeCatalyst espaço tem cotas diferentes de outras conexões de conta para um espaço. Para obter mais informações, consulte [Cotas para o CodeCatalyst](#).

Para remover uma conta que é uma conta de cobrança designada para seu CodeCatalyst espaço, certifique-se primeiro de especificar uma nova conta de cobrança.

Configuração de perfis do IAM para contas conectadas

Você cria funções no AWS Identity and Access Management (IAM) para a conta à qual deseja adicionar CodeCatalyst. Se você estiver adicionando uma conta de faturamento, não precisará criar perfis.

No seu Conta da AWS, você deve ter permissões para criar funções para o que Conta da AWS deseja adicionar ao seu espaço. Para ter mais informações sobre as políticas e perfis do IAM, incluindo referências do IAM e políticas de exemplo, consulte [Identity and Access Management e Amazon CodeCatalyst](#). Para obter mais informações sobre a política de confiança e os princípios de serviço usados em CodeCatalyst, consulte [Entendendo o modelo de CodeCatalyst confiança](#).

Em CodeCatalyst, você deve estar conectado com a função de administrador do espaço para concluir as etapas para adicionar contas (e as funções, se aplicável) ao seu espaço.

É possível adicionar perfis às conexões da sua conta usando um dos métodos a seguir.

- Para criar um perfil de serviço que contenha a política de permissões e a política de confiança para o perfil CodeCatalystWorkflowDevelopmentRole-**spaceName**, consulte [Perfil do CodeCatalystWorkflowDevelopmentRole-**spaceName**](#).
- Para ver um exemplo de como criar um perfil e adicionar uma política para criar um projeto a partir de um esquema, consulte [Criação de uma função do IAM e uso da política de CodeCatalyst confiança](#).
- Para ver uma lista de exemplos de políticas de perfil a serem usadas ao criar seus perfis do IAM, consulte [Conceda acesso aos AWS recursos do projeto com funções do IAM](#).

- Para ver etapas detalhadas para criar perfis para ações de fluxo de trabalho, consulte o tutorial do fluxo de trabalho para essa ação da seguinte forma:
 - [Tutorial: Fazer upload de artefatos no Amazon S3](#)
 - [Tutorial: Implantar uma aplicação sem servidor](#)
 - [Tutorial: Implantar uma aplicação no Amazon ECS](#)
 - [Tutorial: código Lint usando uma ação do GitHub](#)

Tópicos

- [Perfil do CodeCatalystWorkflowDevelopmentRole-spaceName](#)
- [Perfil do AWSRoleForCodeCatalystSupport](#)
- [Criação de uma função do IAM e uso da política de CodeCatalyst confiança](#)

Perfil do CodeCatalystWorkflowDevelopmentRole-*spaceName*

Você cria o perfil de desenvolvedor como um perfil de 1 clique no IAM. Você deve ter o perfil de administrador do espaço ou usuário avançado no espaço em que deseja adicionar a conta. Você também deve ter permissões administrativas para o que Conta da AWS deseja adicionar.

Antes de iniciar o procedimento abaixo, você deve fazer login no Console de gerenciamento da AWS com a mesma conta que deseja adicionar ao seu CodeCatalyst espaço. Caso contrário, o console retornará um erro de conta desconhecido.

Para criar e adicionar o CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Antes de começar no CodeCatalyst console, abra o e Console de gerenciamento da AWS, em seguida, verifique se você está logado com o mesmo Conta da AWS em seu espaço.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
4. Escolha o link para Conta da AWS onde você deseja criar a função. A página Detalhes da Conta da AWS é exibida.
5. Escolha Gerenciar funções em Console de gerenciamento da AWS.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

- Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento. O perfil terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName*](#).

 Note

Essa função só é recomendada para uso com contas de desenvolvedor e usa a política `AdministratorAccess AWS` gerenciada, dando a ela acesso total para criar novas políticas e recursos nela Conta da AWS.

- Selecione Criar perfil de desenvolvimento.
- Na página de conexões, em Funções do IAM disponíveis para CodeCatalyst, veja a `CodeCatalystWorkflowDevelopmentRole-spaceName` função na lista de funções do IAM adicionadas à sua conta.
- Para retornar ao seu espaço, escolha Go to Amazon CodeCatalyst.

Perfil do AWSRoleForCodeCatalystSupport

Você cria o perfil de suporte como um perfil de 1 clique no IAM. Você deve ter o perfil de administrador do espaço ou usuário avançado no espaço em que deseja adicionar a conta. Você também deve ter permissões administrativas para o que Conta da AWS deseja adicionar.

Antes de iniciar o procedimento abaixo, você deve fazer login no Console de gerenciamento da AWS com a mesma conta que deseja adicionar ao seu CodeCatalyst espaço. Caso contrário, o console retornará um erro de conta desconhecido.

Para criar e adicionar o CodeCatalyst AWSRoleForCodeCatalystSupport

- Antes de começar no CodeCatalyst console, abra o e Console de gerenciamento da AWS, em seguida, verifique se você está logado com o mesmo Conta da AWS em seu espaço.
- Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
- Escolha o link para Conta da AWS onde você deseja criar a função. A página Detalhes da Conta da AWS é exibida.
- Escolha Gerenciar funções em Console de gerenciamento da AWS.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

5. Em detalhes do CodeCatalyst espaço, escolha Adicionar função de CodeCatalyst Support. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento de demonstração. O perfil terá um nome `AWSRoleForCodeCatalystSupport` com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço `AWSRoleForCodeCatalystSupport`](#).
6. Na página Adicionar função para CodeCatalyst Support, deixe o padrão selecionado e escolha Criar função.
7. Em Funções do IAM disponíveis para CodeCatalyst, veja a `CodeCatalystWorkflowDevelopmentRole-spaceName` função na lista de funções do IAM adicionadas à sua conta.
8. Para retornar ao seu espaço, escolha Go to Amazon CodeCatalyst.

Criação de uma função do IAM e uso da política de CodeCatalyst confiança

As funções do IAM a serem usadas CodeCatalyst com Conta da AWS conexões devem ser configuradas para usar a política de confiança fornecida aqui. Use essas etapas para criar uma função do IAM e anexar uma política que permita criar projetos a partir de blueprints em CodeCatalyst.

Você também pode criar um perfil de serviço que contenha a política de permissões e a política de confiança para o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para obter mais informações, consulte [Adicionar perfis do IAM às conexões da conta](#).

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Selecione Funções e, depois, Criar função.
3. Selecione Política de confiança personalizada.
4. No formulário Política de confiança personalizada, cole a seguinte política de confiança.

```
"Version": "2012-10-17",
```



```

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

```

5. Escolha Próximo.
6. Em Adicionar permissões, pesquise e selecione uma política personalizada que você já criou no IAM.
7. Escolha Próximo.
8. Em Nome do perfil, insira um nome para seu perfil, por exemplo, `codecatalyst-project-role`.
9. Selecione Criar perfil.
10. Copie o nome do recurso da Amazon (ARN) do perfil. Você precisará fornecer essas informações ao adicionar o perfil à conexão ou ao ambiente da sua conta.

Concessão de permissões de espaço aos usuários

Você pode gerenciar membros de um espaço visualizando, adicionando, removendo ou alterando os perfis dos usuários que ingressam no espaço.

As informações neste guia são fornecidas para convidar e gerenciar usuários em espaços CodeCatalyst que oferecem suporte aos usuários do AWS Builder ID. Para saber mais sobre as etapas para configurar e administrar um espaço que ofereça suporte à federação de identidades,

consulte [Configuração e administração de CodeCatalyst espaços](#) no Amazon CodeCatalyst Administrator Guide.

Visualização dos membros em um espaço

Você pode visualizar os usuários em seu espaço, incluindo informações sobre seus nomes de exibição, aliases e o perfil que eles têm no espaço. Há três perfis para membros em um espaço:

- **Administrador de espaço** — Essa função tem todas as permissões CodeCatalyst, incluindo a criação de projetos. Atribua esse perfil somente aos usuários que precisam administrar todos os aspectos de um espaço, como acessar todos os projetos no espaço.

Você não pode alterar esse perfil posteriormente sem primeiro remover o usuário. Para obter mais informações, consulte [Perfil de administrador do espaço](#).

- **Usuário avançado** — Essa função é a segunda função mais poderosa nos CodeCatalyst espaços da Amazon, mas não tem acesso a projetos em um espaço. Ele foi projetado para usuários que precisam criar projetos em um espaço e ajudar a gerenciar os usuários e recursos do espaço. Para obter mais informações, consulte [Perfil de usuário avançado](#).
- **Acesso limitado** – Esse perfil é atribuído por padrão aos usuários que participam do espaço aceitando convites para projetos no espaço. Os membros do projeto recebem um perfil em um projeto. Para ter informações sobre o gerenciamento de membros do projeto, consulte [Concessão de permissões de projeto aos usuários](#).

A tabela Administradores do espaço mostra os usuários com o perfil de Administrador do espaço. Esses usuários não são exibidos em membros do espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm um perfil em um projeto.

A tabela de membros do espaço mostra todos os membros no espaço que têm um perfil em um projeto, mas não têm o perfil de administrador do espaço.

Os usuários são mostrados com base no fato de o usuário ter a função de administrador do Space da CodeCatalyst seguinte forma:

- Um usuário com o perfil de administrador do espaço que posteriormente aceitar um convite e um perfil do projeto não aparecerá na tabela de membros do espaço em espaços ou na tabela de membros do projeto em projetos. Eles continuarão sendo exibidos na tabela de administradores do espaço em ambos os lugares. Em cada projeto, todos os usuários com o perfil de administrador do espaço são mostrados na tabela de administradores do espaço do projeto.

- Um usuário que aceita um convite de projeto para participar com um perfil de projeto é adicionado ao espaço com o perfil de acesso limitado. Se o perfil do usuário mudar posteriormente para o perfil de administrador do espaço, mas também passar da tabela de membros do espaço para a tabela de administradores do espaço. No projeto, o usuário passará da tabela de membros do projeto para a tabela de administradores do espaço.

Para visualizar usuários e perfis em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e Membros.

Os usuários que são membros do espaço são exibidos na tabela de membros do espaço.

 Tip

Se você tiver o perfil de administrador do espaço, poderá visualizar para quais projetos foi convidado diretamente. Navegue até Configurações do projeto e selecione Meus projetos.

Na coluna Status, estes são valores válidos:

- CodeCatalyst Convidado — enviou o convite, mas o usuário ainda não aceitou ou recusou.
- Membro – O usuário aceitou o convite.

Convite de um usuário diretamente para um espaço

Você pode convidar usuários diretamente para o seu CodeCatalyst espaço. Isso é útil quando você deseja convidar esse usuário para ajudá-lo a gerenciar o espaço atribuindo o perfil de administrador do espaço ou usuário avançado. Atribuir um desses perfis a outros usuários pode ajudar a distribuir

as responsabilidades de gerenciar o espaço entre mais pessoas sem precisar convidar esses usuários para nenhum projeto.

 Note

Você deve ter o perfil de administrador do espaço ou usuário avançado para convidar membros.

A tabela Administradores do espaço mostra os usuários com o perfil de Administrador do espaço. Esses usuários não são exibidos na tabela de membros do espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm um perfil em um projeto.

Os membros que aceitam um convite de projeto são adicionados ao espaço por padrão. A tabela de membros do projeto mostra todos os membros no espaço que têm um perfil em um projeto.

Para ter mais informações sobre como aceitar um convite e entrar pela primeira vez, consulte [Configuração e login no CodeCatalyst](#).

Como convidar um usuário para o espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.
3. Selecione Configurações e Membros.
4. Escolha Convidar.
5. Insira o e-mail da pessoa que você gostaria de convidar para entrar no seu espaço. Em Perfil, escolha o perfil que você deseja atribuir ao usuário no espaço.
6. Selecione Convidar.

Cancelamento de um convite para um espaço

Se você quiser cancelar um convite para participar de um espaço enviado recentemente, mas que ainda não foi aceito, você pode cancelá-lo.

Para gerenciar convites de espaço, você deve ter o perfil de administrador do espaço ou usuário avançado.

Para cancelar um convite de membro do espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e Membros.
4. Verifique se o membro tem o status Convidado.

 Note

Você só pode cancelar um convite que ainda não tenha sido aceito.

5. Selecione a opção ao lado da linha com o membro convidado e selecione Cancelar convite.
6. Uma janela de confirmação é exibida. Selecione Cancelar convite para confirmar.

Alteração do perfil de um membro do espaço

Você pode alterar o perfil atribuído a um membro do seu espaço. Você deve ter o perfil de administrador do espaço para alterar o perfil de um usuário no espaço.

A tabela Administradores do espaço mostra os usuários com o perfil de Administrador do espaço. Esses usuários não são exibidos na tabela de membros do espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço.

Para alterar o perfil de um usuário em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e Membros.
4. Na tabela de membros do espaço, selecione o usuário cujo perfil você deseja alterar. Selecione Alterar perfil.

Remoção de um membro do espaço

Você pode remover um membro do seu espaço quando ele não precisar acessar nenhum dos recursos do espaço. Você deve ter o perfil de administrador do espaço para remover um membro de um espaço.

A tabela Administradores do espaço mostra os usuários com o perfil de Administrador do espaço. Esses usuários não são exibidos na tabela de membros do espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm um perfil em um projeto. Você só pode remover diretamente um membro do seu espaço nessa tabela.

Para remover um usuário da tabela de membros do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e Membros.
4. Selecione o usuário na tabela de membros do projeto. Escolha Remover .

Note

A remoção de um membro do espaço removerá o usuário de todos os projetos no espaço, junto com as permissões associadas aos recursos desses projetos.

Remoção ou alteração do perfil de um usuário com o perfil de administrador do espaço

Você pode remover ou alterar o perfil de um usuário com o perfil de administrador do espaço no seu espaço.

Você deve ter o perfil de administrador do espaço para remover um usuário com o perfil de administrador do espaço de um espaço. Alterar o perfil de um usuário com o perfil de administrador do espaço basicamente remove o usuário da tabela de administradores do espaço. Se esse usuário não tiver um perfil de projeto em nenhum projeto no espaço, remover o perfil de administrador do espaço do usuário o removerá do espaço.

Note

Como usuário com o perfil de administrador do espaço, você não pode se remover. Entre em contato com outro usuário com o perfil de administrador do espaço.

Para remover um usuário com o perfil de administrador do espaço da tabela de membros do espaço

Note

Para um usuário que não foi adicionado explicitamente a um projeto, ele não tem nenhum perfil no projeto (administrador do projeto ou colaborador). Se o perfil de administrador do espaço for o único perfil do usuário, o usuário será totalmente removido do espaço.

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço onde você deseja remover ou alterar o perfil de um usuário com o perfil de administrador do espaço.
3. Selecione Configurações e Membros.
4. Veja o status do convite da lista de membros e certifique-se de que a lista não contenha convites pendentes não autorizados para o espaço (um status de Convidado).

 Important

Antes de remover um usuário com o perfil de administrador do espaço, você deve verificar se nenhum convite pendente foi iniciado.

- Escolha a guia Membros. Na tabela de administradores do espaço, selecione o usuário e, depois, selecione Remover.

Na caixa de diálogo Remover membro, execute uma das seguintes ações.

- Selecione a opção para remover somente o perfil de administrador do espaço do usuário. Escolha Remover .

 Important

Se o usuário não tiver nenhum outro perfil atribuído, a alteração do perfil de administrador do espaço removerá o usuário do espaço.

- Selecione a opção para remover um usuário com o perfil Administrador do espaço do espaço e de todos os projetos. Escolha Remover.
- Atualize a guia Membros. O usuário é adicionado automaticamente à lista de membros do projeto em qualquer projeto em que o usuário tenha sido membro por meio de perfis no projeto. Se o perfil de administrador do espaço foi o único perfil do usuário, o usuário será totalmente removido do espaço.

Permitir o acesso ao espaço usando equipes

Depois de criar um espaço, você pode adicionar equipes. As equipes permitem que você agrupe usuários para que eles possam compartilhar permissões e gerenciar projetos, rastreamento de problemas, funções e recursos em CodeCatalyst.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

As equipes também são gerenciadas no project/space nível em CodeCatalyst. Para saber mais sobre equipes em espaços/projetos, consulte [Permitir o acesso ao espaço usando equipes](#).

Tópicos

- [Criação de uma equipe](#)

- [Visualizar uma equipe](#)
- [Conceder perfis de espaço para uma equipe](#)
- [Concessão de perfis de projeto para uma equipe no nível do espaço](#)
- [Adição de um usuário diretamente a uma equipe](#)
- [Remover um usuário de uma equipe diretamente](#)
- [Adição de um grupo de SSO a uma equipe](#)
- [Exclusão de uma equipe](#)

Criação de uma equipe

Uma equipe pode ter permissões de perfil, como Usuário avançado, em um espaço. Uma equipe também pode ter permissões de projeto, como Administrador do projeto, em um projeto. As equipes podem ser associadas a vários projetos com perfis diferentes para cada projeto. Você pode gerenciar equipes em que os membros da equipe sejam usuários individuais para um espaço de ID do AWS Builder ou grupos de SSO para um espaço que ofereça suporte à federação de identidades.

Na página de membros para usuários do espaço e do projeto, os usuários podem ter vários perfis. Os usuários com vários perfis mostrarão um indicador quando tiverem vários perfis e serão exibidos primeiro com o perfil com mais permissões.

Note

Se o espaço oferecer suporte à federação de identidades, você já deve ter usuários de SSO ou grupos de SSO configurados no IAM Identity Center.

A forma como você gerencia os membros da equipe depende de como adicionará e removerá usuários. Há duas opções de gerenciamento de membros da equipe:

- Adicionar usuários diretamente — Você adiciona ou remove usuários individualmente. Por exemplo, você adiciona usuários a uma equipe escolhendo usuários do AWS Builder ID ou usuários de SSO que já estão configurados no IAM Identity Center. Quando você escolhe gerenciar os membros da equipe adicionando usuários do AWS Builder ID ou usuários de SSO diretamente, a opção de usar grupos de SSO não estará mais disponível.

- Usar grupos de SSO: você gerencia os membros da equipe por meio de grupos de SSO já configurados no IAM Identity Center. Quando você decidir gerenciar membros da equipe usando Grupos de SSO, a opção de adicionar usuários diretamente não estará mais disponível.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Para criar uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Selecione Criar equipe.
4. Em Nome da equipe, insira um nome descritivo para a sua equipe.

 Note

O nome da equipe deve ser exclusivo no espaço.

(Opcional) Em Descrição da equipe, insira uma descrição para sua equipe.

5. Em Função espacial, escolha uma função na lista de funções espaciais disponíveis CodeCatalyst que você deseja atribuir à equipe. O perfil será herdado por todos os membros da equipe.
 - Administrador do espaço: para saber detalhes, consulte [Perfil de administrador do espaço](#).
 - Acesso limitado: para saber detalhes, consulte [Perfil de acesso limitado](#).
 - Usuário avançado: para saber detalhes, consulte [Perfil de usuário avançado](#).
6. Em Associação à equipe, escolha uma das opções a seguir para escolher o método para adicionar membros à equipe.
 - Selecione Adicionar membros diretamente para gerenciar usuários individualmente. Isso inclui adicionar usuários do AWS Builder ID a um espaço ou adicionar usuários de SSO a um espaço que ofereça suporte à federação de identidades.
 - Selecione Usar grupos de SSO para escolher grupos de SSO que você já configurou no IAM Identity Center.

Em Grupos de SSO, selecione a caixa ao lado dos grupos que você deseja adicionar. Você pode adicionar até cinco grupos de SSO.

 Note

Não é possível alterar essa opção mais tarde. Quando você decide gerenciar membros da equipe adicionando usuários do ID do builder AWS ou usuários de SSO diretamente, a opção de usar Grupos de SSO não está mais disponível. Quando você decidir gerenciar membros da equipe usando Grupos de SSO, a opção de adicionar usuários diretamente não estará mais disponível.

7. Escolha Criar.

 Note

Ao optar por usar grupos de SSO, observe que os usuários do grupo de SSO não são retirados após a criação da equipe. Os usuários precisarão fazer login CodeCatalyst antes de serem visíveis na lista.

Visualizar uma equipe

Em CodeCatalyst, você pode visualizar os projetos e funções da sua equipe. Na página de membros, você pode ver os perfis do projeto e uma lista de usuários. Para equipes do tipo grupo de SSO, você também poderá ver uma lista de grupos de SSO associados à equipe.

Para visualizar uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Em Perfil do espaço, visualize o perfil atribuído à equipe nesse espaço.
4. Na guia Funções do projeto, visualize o projeto e a função do projeto atribuída à equipe para cada CodeCatalyst projeto no espaço em que a equipe foi adicionada como membro (somente para um espaço de ID do AWS Builder).
5. Na guia Membros, visualize a lista de membros atribuídos à equipe.
6. Na guia Grupos de SSO, visualize a lista de grupos de SSO atribuídos à equipe (para um espaço que ofereça suporte somente à federação de identidades).

Conceder perfis de espaço para uma equipe

As equipes são uma forma de agrupar usuários para que você possa conceder e gerenciar o acesso da equipe aos projetos em CodeCatalyst. Como exemplo, você pode usar equipes para gerenciar rapidamente os perfis e permissões dos usuários, dando a uma equipe a capacidade de gerenciar um espaço para os usuários.

Uma equipe pode ter permissões de perfil, como Usuário avançado, em um espaço. Você pode alterar o perfil do espaço de uma equipe, mas observe que todos os membros da equipe herdarão essas permissões.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Alteração do perfil do espaço de uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Em Ações, selecione Alterar perfil do espaço. Você pode alterar o perfil do espaço com uma das seguintes opções. Isso muda o perfil de todos os membros da equipe.
 - Administrador do espaço: para saber detalhes, consulte [Perfil de administrador do espaço](#).
 - Acesso limitado: para saber detalhes, consulte [Perfil de acesso limitado](#).
 - Usuário avançado: para saber detalhes, consulte [Perfil de usuário avançado](#).
4. Escolha Salvar.

Concessão de perfis de projeto para uma equipe no nível do espaço

Uma equipe em CodeCatalyst é semelhante a um usuário, pois os membros da equipe podem ter permissões de função, como administrador do projeto, em um projeto. Uma mudança de perfil será aplicada à equipe e todos os membros da equipe herdarão essas permissões. Você pode escolher um perfil para cada projeto que será automaticamente concedido à equipe.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Como adicionar ou alterar um perfil do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.

3. Escolha a guia Perfis do projeto.
4. Para alterar um perfil, escolha o seletor ao lado do projeto nessa lista e selecione Alterar perfil. Para adicionar um perfil, selecione Adicionar perfil do projeto. Em Projeto, escolha o projeto que você deseja adicionar e, em Perfil, escolha o perfil. Escolha um dos perfis de projeto disponíveis:
 - Administrador do projeto: para saber detalhes, consulte [Perfil de administrador do projeto](#).
 - Colaborador: para saber detalhes, consulte [Perfil de colaborador](#).
 - Revisor: para saber detalhes, consulte [Perfil de revisor](#).
 - Somente leitura: para saber detalhes, consulte [Perfil somente leitura](#).
5. Escolha Salvar.

Como remover um perfil do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Escolha a guia Perfis do projeto.
4. Escolha o perfil que deseja remover.

Important

A remoção de um perfil de uma equipe remove as permissões associadas para todos os usuários da equipe.

5. Escolha Salvar.

Adição de um usuário diretamente a uma equipe

Você pode adicionar membros à sua equipe. Quando você adiciona um usuário, o novo usuário herda as permissões de todos os perfis existentes na equipe.

Se seu espaço está configurado para suporte ao usuário do AWS Builder ID ou federação de identidades, você pode configurar seu espaço para adicionar usuários diretamente.

 Note

Quando o espaço está configurado para gerenciar membros da equipe usando grupos de SSO, a opção de usar Adicionar usuários diretamente não está disponível. Para usar grupos de SSO, consulte [Adição de um grupo de SSO a uma equipe](#).

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Para adicionar um usuário diretamente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Escolha a guia Membros.
4. Escolha Adicionar membro.

 Note

Os usuários adicionados a uma equipe já devem ser membros de um espaço. Você não pode adicionar ou convidar um membro da equipe que não seja membro do espaço.

5. Escolha um usuário no campo suspenso e selecione Salvar. Escolha usuários do AWS Builder ID ou usuários de SSO que já estejam configurados no IAM Identity Center.

Remover um usuário de uma equipe diretamente

Você pode remover membros da sua equipe. Todas as permissões não serão mais herdadas pelo usuário. Você pode adicionar o usuário de volta à equipe mais tarde.

 Note

Quando você remove um membro da equipe, as permissões associadas são removidas para o usuário de todos os projetos e recursos no espaço.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Para remover um membro da equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Escolha a guia Membros.
4. Escolha o seletor ao lado do usuário que você deseja remover e escolha selecione Remover.
5. Digite remove no campo de entrada e selecione Remover.

Adição de um grupo de SSO a uma equipe

Se o espaço estiver configurado como um espaço com usuários e grupos de SSO gerenciados no IAM Identity Center, você poderá adicionar um grupo de SSO que se juntará ao espaço como uma equipe separada.

Note

Quando você escolhe gerenciar membros da equipe adicionando usuários do AWS Builder ID ou usuários de SSO diretamente, a opção de usar grupos de SSO não está disponível. Para adicionar usuários diretamente, consulte [Adição de um usuário diretamente a uma equipe](#).

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Para adicionar um grupo de SSO como uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página do seu espaço, selecione Equipes. Escolha a guia Grupos de SSO.
3. Escolha os grupos de SSO que deseja adicionar. Você pode adicionar até cinco grupos de SSO.

Exclusão de uma equipe

Você pode excluir uma equipe da qual não precisa mais.

Note

Quando você exclui uma equipe, as permissões associadas são removidas para todos os membros da equipe de todos os projetos e recursos no espaço.

Você deve ter o perfil de Administrador do espaço para gerenciar equipes.

Excluir uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações e Equipes.
3. Em Ações, selecione Excluir equipe. Isso mudará o perfil da equipe inteira.
4. Escolha Excluir.

Permissão do acesso ao espaço para recursos da máquina

Os recursos da máquina são recursos específicos nos CodeCatalyst quais são concedidas permissões para projetos ou espaços em CodeCatalyst.

Note

O termo recurso de máquina não se refere à infraestrutura em nuvem, como uma EC2 instância da Amazon, mas sim a um plano ou recurso de fluxo de trabalho com permissões para um espaço ou projeto.

Um recurso de máquina representa sua identidade a partir do seu recurso autorizado ao acessar CodeCatalyst por meio de SSO. Os recursos da máquina são usados para conceder permissões aos recursos no espaço, como esquemas e fluxos de trabalho. Você pode visualizar os recursos da máquina no espaço e escolher entre ativar ou desativar os recursos da máquina do espaço. Por exemplo, talvez você queira desativar um recurso da máquina para gerenciar o acesso e reativá-lo posteriormente.

Essas operações estão disponíveis quando um recurso da máquina precisa ser revogado ou desativado. Por exemplo, se você suspeitar que as credenciais podem ter sido comprometidas, poderá desativar o recurso da máquina. Geralmente, essas operações não precisarão ser usadas.

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar recursos de máquina no nível do espaço.

Os recursos da máquina também são gerenciados no nível do projeto em CodeCatalyst. Para saber mais sobre equipes em projetos, consulte [Permissão do acesso ao espaço para recursos da máquina](#).

Tópicos

- [Visualização do acesso ao espaço para recursos da máquina](#)
- [Desativação do acesso ao espaço para recursos da máquina](#)
- [Ativação do acesso ao espaço para recursos da máquina](#)

Visualização do acesso ao espaço para recursos da máquina

Você pode ver uma lista dos recursos da máquina que estão em uso no seu espaço.

Você deve ter o perfil de Administrador do espaço para gerenciar os recursos da máquina.

Para visualizar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu espaço e selecione Configurações. Selecione Recursos da máquina.
3. No menu suspenso, selecione Ação de fluxo de trabalho para visualizar somente os recursos da máquina para fluxos de trabalho. Selecione Esquema para visualizar somente os recursos da máquina para esquemas.

Você também pode filtrar por um nome usando o campo Filtro.

Desativação do acesso ao espaço para recursos da máquina

Você pode optar por desativar os recursos da máquina que estão em uso no seu espaço.

Important

A desativação dos recursos da máquina removerá todas as permissões de todos os esquemas ou fluxos de trabalho associados no espaço.

Você deve ter o perfil de Administrador do espaço para gerenciar os recursos da máquina.

Para desabilitar recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu espaço e selecione Configurações. Selecione Recursos da máquina.
3. Escolha uma das opções a seguir.

 Important

A desativação dos recursos da máquina removerá todas as permissões de todos os esquemas ou fluxos de trabalho associados no espaço.

- Para desabilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja desabilitar. Selecione Desabilitar e Este recurso.
- Para desabilitar todos os recursos, selecione Desabilitar e, depois, Todos os recursos.
- Para desabilitar todas as ações do fluxo de trabalho, selecione Desabilitar e, depois, Todas as ações do fluxo de trabalho.
- Para desabilitar todos os esquemas, selecione Desabilitar e, depois, Todos os esquemas.

Ativação do acesso ao espaço para recursos da máquina

Você pode optar por ativar os recursos da máquina que estão em uso no seu espaço e que foram desativados.

Você deve ter o perfil de Administrador do espaço para gerenciar os recursos da máquina.

Para habilitar recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu espaço e selecione Configurações. Selecione Recursos da máquina.
3. Escolha uma das opções a seguir.
 - Para habilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja habilitar. Selecione Habilitar e Este recurso.
 - Para habilitar todos os recursos, selecione Habilitar e, depois, Todos os recursos.

- Para habilitar todas as ações do fluxo de trabalho, selecione Habilitar e, depois, Todas as ações do fluxo de trabalho.
- Para habilitar todos os esquemas, selecione Habilitar e, depois, Todos os esquemas.

Administração de Ambientes de Desenvolvimento para um espaço

Todos os Ambientes de Desenvolvimento são criados como parte de um projeto dentro de um espaço. Os membros do espaço podem criar seus próprios Ambientes de Desenvolvimento em um projeto no nível do repositório de origem. Os administradores do espaço podem então usar o CodeCatalyst console da Amazon para visualizar, editar, excluir e interromper ambientes de desenvolvimento em nome dos membros do espaço. Resumindo, os administradores do espaço mantêm os Ambientes de Desenvolvimento no nível do espaço.

Considerações para administrar Ambientes de Desenvolvimento

- Você deve ter o perfil de administrador do espaço para visualizar a página Ambientes de Desenvolvimento em Configurações e gerenciar os Ambientes de Desenvolvimento no nível do espaço.
- Os membros do Space gerenciam os ambientes de desenvolvimento que eles criam nos projetos por meio de suas CodeCatalyst contas. Ao administrar Ambientes de Desenvolvimento como administrador do espaço, você está mantendo esses recursos em nome dos membros do espaço.
- Os Ambientes de Desenvolvimento usam como padrão uma configuração específica de computação e armazenamento. Para obter informações sobre faturamento e tarifas para atualizar sua configuração, consulte a página de [CodeCatalyst preços da Amazon](#).

Important

Os ambientes de desenvolvimento não estão disponíveis para usuários em espaços em que o Active Directory é usado como provedor de identidade. Para obter mais informações, consulte [Não consigo criar um Ambiente de Desenvolvimento quando estou conectado ao CodeCatalyst usando uma conta de login único](#).

Para outras considerações sobre Ambientes de Desenvolvimento, incluindo parar a execução de instâncias, configuração de computação padrão, atualizar sua computação, incorrer em custos

e configurar tempos limite, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

Tópicos

- [Visualização de Ambientes de Desenvolvimento para seu espaço](#)
- [Editar um Ambiente de Desenvolvimento para o espaço](#)
- [Interromper um Ambiente de Desenvolvimento para o espaço](#)
- [Excluir um Ambiente de Desenvolvimento para o espaço](#)

Visualização de Ambientes de Desenvolvimento para seu espaço

Você pode visualizar o tipo, o status e os detalhes de todos os Ambientes de Desenvolvimento em seu espaço. Para ter mais informações sobre como criar e executar Ambientes de Desenvolvimento, consulte [Criar um Ambiente de Desenvolvimento](#).

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar Ambientes de Desenvolvimento no nível do espaço.

Para visualizar Ambientes de Desenvolvimento em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, selecione Ambientes de Desenvolvimento.

A página lista todos os Ambientes de Desenvolvimento em seu espaço. Você pode visualizar o nome do recurso, o alias do recurso, se aplicável, o tipo de IDE, a computação e o armazenamento padrão ou configurados e o tempo limite configurado para cada Ambiente de Desenvolvimento.

Editar um Ambiente de Desenvolvimento para o espaço

Você pode editar a configuração de um Ambiente de Desenvolvimento, como o tempo limite configurado, se houver, para que um Ambiente de Desenvolvimento ocioso pare de ser executado. Para ter mais informações sobre como editar um Ambiente de Desenvolvimento, consulte [Edição de um Ambiente de Desenvolvimento](#).

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar Ambientes de Desenvolvimento no nível do espaço.

Para editar Ambientes de Desenvolvimento em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, selecione Ambientes de Desenvolvimento.
4. Escolha o seletor ao lado do Ambiente de Desenvolvimento que você deseja gerenciar. Escolha Editar.
5. Faça as alterações desejadas no tempo limite de computação ou inatividade do Ambiente de Desenvolvimento.
6. Escolha Salvar.

Interromper um Ambiente de Desenvolvimento para o espaço

Você pode parar a execução de um Ambiente de Desenvolvimento antes que ele fique ocioso, se estiver configurado para ter um tempo limite. Caso contrário, um Ambiente de Desenvolvimento com um tempo limite expirado já será parado. Para ter mais informações sobre como parar um Ambiente de Desenvolvimento, consulte [Interromper um Ambiente de Desenvolvimento](#).

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar Ambientes de Desenvolvimento no nível do espaço.

Para parar Ambientes de Desenvolvimento em seu espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, selecione Ambientes de Desenvolvimento.
4. Escolha o seletor ao lado do Ambiente de Desenvolvimento que você deseja gerenciar. Escolha Parar.

Excluir um Ambiente de Desenvolvimento para o espaço

Você pode excluir um Ambiente de Desenvolvimento que não seja mais necessário ou que não tenha mais um proprietário. Para ter mais informações sobre considerações para excluir um Ambiente de Desenvolvimento, consulte [Excluir um Ambiente de Desenvolvimento](#).

Você deve ter o perfil de Administrador do espaço para visualizar esta página e gerenciar Ambientes de Desenvolvimento no nível do espaço.

Como excluir Ambientes de Desenvolvimento no espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Selecione Configurações e, depois, selecione Ambientes de Desenvolvimento.
4. Escolha o seletor ao lado do Ambiente de Desenvolvimento que você deseja gerenciar. Escolha Excluir. Para confirmar, digite delete e escolha Excluir.

Cotas para espaços

A tabela a seguir descreve as cotas e os limites de espaços na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

Número máximo de canais do Slack para um espaço	500
Número máximo de convites para um endereço de e-mail	25
Número máximo de convites para um usuário	500
Número máximo de espaços ativos por usuário por Região da AWS	5
Número máximo de criações de espaço por região por mês por usuário	5
Número máximo de grupos de SSO para uma equipe	5
Número máximo de equipes por um espaço	100
Número máximo de usuários para uma equipe	1000
Descrições do espaço	As descrições do espaço são opcionais. Se especificado, elas devem ter entre 0 e 200 caracteres de comprimento. Elas podem conter qualquer combinação de letras, números, espaços, pontos, sublinhados, vírgulas, traços e os caracteres especiais a seguir: ? & \$ % + = / \ ; : \n \t \r
Nomes de espaço	Os nomes dos espaços devem ser exclusivos em todos os lados CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.

Os nomes de espaço deve ter entre 3 e 63 caracteres de comprimento. Eles também devem começar com um caractere alfanumérico. Os nomes dos espaços podem conter qualquer combinação de letras, números, pontos, sublinhados e traços. Eles não podem conter nenhum dos seguinte caracteres:

! ? @ # \$ % ^ & * () + = { } []
| \ > < ~ ` ' " ; :

Organize o trabalho com projetos em CodeCatalyst

Você usa projetos na Amazon CodeCatalyst para estabelecer um espaço de colaboração onde as equipes de desenvolvimento podem realizar tarefas de desenvolvimento com fluxos de trabalho e repositórios de integração/continuous entrega contínua compartilhada (CI/CD). Ao criar um projeto, é possível adicionar, atualizar ou remover recursos. Também é possível monitorar o andamento do trabalho da sua equipe. Você pode ter vários projetos em um espaço.

Os espaços em CodeCatalyst são compostos por projetos. Você pode ver todos os projetos em seu espaço, mas só pode usar os projetos dos quais é membro. Quando você cria um projeto, são gerados perfis padrão para o projeto, que são atribuídos aos usuários convidados para o projeto.

- Qualquer pessoa atribuída ao projeto com um perfil de projeto, como o de Colaborador, pode acessar os recursos do projeto, como um repositório de origem.
- Qualquer pessoa com o perfil de Administrador do espaço ou Administrador do projeto pode enviar convites para participar de um projeto.
- Os usuários com o perfil de Administrador do projeto podem monitorar atividades, status e outras configurações nos recursos compartilhados.
- Os usuários com a função de acesso limitado podem gerenciar atribuições de projetos para recursos, correções de código e testes como parte dos CI/CD fluxos de trabalho.

Os fluxos de trabalho são usados para criar, testar e liberar ou atualizar aplicativos como CI/CD pipeline. You can assemble workflows by adding actions that transfer and work on your source artifacts. When you run actions, your project cloud resources are used to provide on-demand compute ability for your workflow actions. You might configure more CI/CD fluxos de trabalho com base na atividade e na saída que você deseja configurar. Por exemplo, você pode criar um fluxo de trabalho somente para ações de criação e teste, onde é possível visualizar os resultados do teste e concluir o fluxo de trabalho sem uma implantação enquanto corrige bugs. Em seguida, você pode criar outro fluxo de trabalho para criar e implantar a aplicação em um ambiente de preparação.

Ao criar um projeto, você pode usar um esquema para criar um projeto que contém código de amostra e cria recursos, ou pode começar com um projeto vazio. Se você criar um projeto usando um blueprint, o blueprint escolhido determinará quais recursos serão adicionados ao seu projeto e as ferramentas que serão CodeCatalyst criadas ou configuradas para que você possa acompanhar

e usar os recursos do seu projeto. É possível adicionar ou remover recursos manualmente depois de criar um projeto.

Cada projeto rastreia a atividade relacionada como uma lista de eventos por usuário, como quando um projeto é criado ou um recurso é modificado. A atividade do projeto é monitorada e agregada no nível do espaço. Para ter mais informações sobre como trabalhar com dados de atividade, consulte [Visualização de todos os projetos em um espaço](#).

Se seu projeto usa AWS recursos, você pode conectar sua CodeCatalyst conta a uma AWS conta na qual tenha permissões administrativas para integrar recursos ao seu projeto.

Você pode adicionar repositórios de origem, problemas e outros recursos ao projeto depois de criá-lo. Você deve ter o perfil de Administrador do espaço para criar projetos.

Como criar um projeto

Com CodeCatalyst projetos, você pode realizar tarefas de desenvolvimento com fluxos de trabalho e repositórios de integration/continuous entrega contínua compartilhada (CI/CD), gerenciar recursos, rastrear problemas e adicionar usuários.

Antes de criar um projeto, você deve ter o perfil de Administrador do espaço ou Usuário avançado.

Tópicos

- [Criando um projeto vazio na Amazon CodeCatalyst](#)
- [Criar um projeto com um repositório de terceiros vinculado](#)
- [Criar um projeto com um esquema](#)
- [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#)
- [Práticas recomendadas para usar esquemas em projetos](#)
- [Adição de recursos e tarefas aos projetos criados](#)

Criando um projeto vazio na Amazon CodeCatalyst

Você pode criar um projeto vazio sem recursos e adicionar manualmente os recursos desejados posteriormente.

Antes de criar um projeto, você deve ter o perfil de Administrador do espaço ou Usuário avançado.

Para criar um projeto vazio

1. Navegue até o espaço onde você deseja criar um projeto.
2. No painel do espaço, escolha Criar projeto.
3. Escolha Começar do zero.
4. Em Dê um nome ao projeto, insira o nome que você deseja atribuir ao projeto. O nome deve ser exclusivo em seu espaço.
5. Escolha Criar projeto.

Criar um projeto com um repositório de terceiros vinculado

Você pode manter o código-fonte do seu projeto em um provedor terceirizado preferido e ainda usar todos os CodeCatalyst recursos, como esquemas, gerenciamento do ciclo de vida, fluxos de trabalho e muito mais. Para fazer isso, você pode criar um novo CodeCatalyst projeto vinculado a um GitHub repositório, repositório Bitbucket ou repositório de projetos. GitLab Em seguida, você pode usar seu repositório de origem vinculado em seu CodeCatalyst projeto.

Antes de criar um CodeCatalyst projeto, você deve ter a função de administrador do espaço ou usuário avançado. Para obter mais informações, consulte [Criar um espaço](#) e [Convite de um usuário diretamente para um espaço](#).

Para criar um projeto vinculado CodeCatalyst a um repositório de origem em sua GitHub conta, você precisará concluir as três tarefas a seguir:

1. Instale os GitHub repositórios, os repositórios do Bitbucket ou GitLab a extensão dos repositórios. Você é solicitado em um site externo a se conectar e CodeCatalyst fornecer acesso ao seu repositório, o que é feito como parte da próxima etapa.

Important

Para instalar os GitHub repositórios, os repositórios do Bitbucket ou a extensão de GitLab repositórios no seu CodeCatalyst espaço, você deve estar conectado com uma conta que tenha a função de administrador do espaço no espaço.

2. Conecte sua GitHub conta ou espaço de trabalho do Bitbucket, ou GitLab usuário a. CodeCatalyst

 Important

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket, GitLab usuário ao seu CodeCatalyst espaço, você deve ser o administrador da fonte terceirizada e o administrador do CodeCatalyst Space.

 Important

Depois de instalar uma extensão de repositório, todos os repositórios vinculados CodeCatalyst terão seu código indexado e armazenado. CodeCatalyst Isso tornará o código pesquisável em CodeCatalyst. Para entender melhor a proteção de dados do seu código ao usar repositórios vinculados CodeCatalyst, consulte [Proteção de dados](#) no Guia do CodeCatalyst usuário da Amazon.

3. Crie um CodeCatalyst projeto vinculado ao seu GitHub repositório, repositório Bitbucket ou GitLab repositório de projetos.

 Important

Embora você possa vincular um GitHub repositório, um repositório do Bitbucket ou um repositório GitLab do projeto como colaborador, você só pode desvincular um repositório de terceiros como administrador do Space ou administrador do projeto. Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

 Note

- Um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados, GitHub repositórios Bitbucket ou repositórios de projetos com GitLab projetos. CodeCatalyst
- Você não pode vincular um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto que tenha o mesmo nome de um repositório em um projeto. CodeCatalyst
- A extensão GitHub repositories não é compatível com os repositórios do GitHub Enterprise Server.
- A extensão de Repositórios do Bitbucket não é compatível com os repositórios do Bitbucket Data Center.
- A extensão GitLab repositories não é compatível com repositórios de projetos GitLab autogerenciados.
- Não é possível usar os recursos Escrever descrição para mim ou Resumir comentários com repositórios vinculados. Esses recursos só estão disponíveis em pull requests em CodeCatalyst.

Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Para instalar a extensão de terceiros

1. Navegue até o espaço onde você deseja criar um projeto.
2. No painel do espaço, escolha Criar projeto.
3. Selecione Traga seu próprio código.
4. Em Vincular repositório existente, escolha repositórios, GitHub repositórios do Bitbucket, repositórios dependendo do provedor de GitLab repositórios terceirizado que você deseja usar. Você será solicitado a conectar sua GitHub conta, espaço de trabalho do Bitbucket ou GitLab

conta, caso não tenha feito isso anteriormente. Se a extensão de terceiros de sua escolha ainda não estiver instalada, um prompt de instalação será exibido.

5. Se solicitado, selecione Instalar. Revise as permissões exigidas pela extensão e, se quiser continuar, escolha Instalar novamente.

Depois de instalar a extensão de terceiros, a próxima etapa é conectar sua GitHub conta, espaço de trabalho do Bitbucket ou GitLab usuário ao seu CodeCatalyst espaço.

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket ou GitLab usuário a CodeCatalyst

Faça o seguinte, dependendo da extensão de terceiros que você decidiu configurar:

- GitHub repositórios: conecte-se a uma GitHub conta.
 1. Escolha Connect GitHub account para acessar o site externo GitHub.
 2. Faça login na sua GitHub conta usando suas GitHub credenciais e, em seguida, escolha a conta na qual você deseja instalar a Amazon CodeCatalyst.

 Tip

Se você já conectou uma GitHub conta ao espaço, não será solicitado que você reautorize. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de um GitHub espaço, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas um GitHub espaço. Configure a aplicação para o acesso ao repositório que você deseja permitir e selecione Salvar. Se o botão Salvar não estiver ativo, faça uma alteração na configuração e tente novamente.

3. Escolha se você deseja permitir o acesso CodeCatalyst a todos os repositórios atuais e futuros ou escolha os GitHub repositórios específicos nos quais deseja usar. CodeCatalyst A opção padrão é incluir todos os GitHub repositórios na GitHub conta, incluindo repositórios futuros que serão acessados pelo. CodeCatalyst
4. Revise as CodeCatalyst permissões concedidas e escolha Instalar.

Depois de conectar sua GitHub conta CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitHub repositórios, onde poderá visualizar e gerenciar GitHub contas conectadas e GitHub repositórios vinculados.

- Repositórios do Bitbucket: conecte a um espaço de trabalho do Bitbucket.
 1. Selecione Conectar espaço de trabalho do Bitbucket para acessar o site externo do Bitbucket.
 2. Faça login no seu espaço de trabalho do Bitbucket usando suas credenciais do Bitbucket e revise as permissões concedidas a. CodeCatalyst
 3. No menu suspenso Autorizar para espaço de trabalho, escolha o espaço de trabalho do Bitbucket ao qual você deseja fornecer CodeCatalyst acesso e, em seguida, escolha Conceder acesso.

 Tip

Se você já conectou um espaço de trabalho do Bitbucket ao espaço, não poderá dar uma nova autorização. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de um espaço de trabalho do Bitbucket, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas um espaço de trabalho do Bitbucket. Configure a aplicação para a qual você deseja permitir o acesso ao espaço de trabalho e selecione Conceder acesso. Se o botão Conceder acesso não estiver ativo, faça uma alteração na configuração e tente novamente.

Depois de conectar seu espaço de trabalho do Bitbucket ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de repositórios do Bitbucket, onde poderá visualizar e gerenciar espaços de trabalho do Bitbucket conectados e repositórios vinculados do Bitbucket.

- GitLab repositórios: conecte-se a um GitLab usuário.
 1. Escolha Connect GitLab user para acessar o site externo GitLab.
 2. Faça login com seu GitLab usuário usando suas GitLab credenciais e revise as permissões concedidas a. CodeCatalyst

 Tip

Se você já conectou um GitLab usuário ao espaço, não será solicitado que você reautorize. Em vez disso, você será direcionado de volta ao CodeCatalyst console.

3. Escolha Autorizar AWS conector para GitLab.

Depois de conectar seu GitLab usuário ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitLab repositórios, onde poderá visualizar e gerenciar repositórios de GitLab usuários e GitLab projetos vinculados.

Depois de conectar sua fonte de terceiros CodeCatalyst, você pode vincular os repositórios de terceiros aos seus CodeCatalyst projetos.

Para criar seu projeto

1. Na página Criar projeto, escolha a GitHub conta que você conectou.
2. Dependendo do provedor de repositório terceirizado que você conectou, escolha o menu suspenso GitHub repositórios, repositórios do Bitbucket ou GitLab repositórios para visualizar os repositórios de terceiros e, em seguida, escolha o repositório que você deseja vincular ao seu projeto.
3. No campo de entrada de texto Dê um nome ao projeto, insira o nome que você deseja atribuir ao projeto. O nome deve ser exclusivo em seu espaço.
4. Escolha Criar projeto.

Depois de instalar os GitHub repositórios, os repositórios do Bitbucket ou a extensão de GitLab repositórios, conectar seu provedor de recursos e vincular seu repositório de terceiros ao seu CodeCatalyst projeto, você pode usá-lo em fluxos de trabalho e ambientes de desenvolvimento. CodeCatalyst Você também pode criar repositórios de terceiros na GitHub conta conectada, no espaço de trabalho do Bitbucket ou no GitLab usuário com código gerado a partir de um blueprint. Você também pode usar os repositórios vinculados com o Amazon Q Developer, esquemas e muito mais. Para obter mais informações, consulte [Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros](#) e [Criar um Ambiente de Desenvolvimento](#).

Criar um projeto com um esquema

Você pode provisionar todos os recursos do projeto e o código de amostra com um esquema do projeto. Para ter mais informações sobre esquemas, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#).

Como criar um projeto com um esquema

1. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.

2. No painel do espaço, escolha Criar projeto.
3. Selecione Começar com um esquema.

 Tip

Você pode optar por adicionar um esquema fornecendo ao Amazon Q os requisitos do projeto para que o Amazon Q sugira um esquema. Para obter mais informações, consulte [Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade](#) e [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#). Esse recurso está disponível somente na região do Oeste dos EUA (Oregon).

Essa funcionalidade exige que recursos de IA generativa estejam habilitados para o espaço. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

4. Na guia CodeCatalyst Blueprints ou Space blueprints, escolha um blueprint e, em seguida, escolha Avançar.
5. Em Nomear seu projeto, insira o nome que você deseja atribuir ao seu projeto e os nomes dos recursos associados. O nome deve ser exclusivo em seu espaço.
6. (Opcional) Por padrão, o código-fonte criado pelo blueprint é armazenado em um CodeCatalyst repositório. Também é possível armazenar o código-fonte do esquema em um repositório de terceiros. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você deseja usar:

- GitHub repositórios: Conecte uma GitHub conta.

Escolha o menu suspenso Avançado, escolha GitHub como provedor do repositório e, em seguida, escolha a GitHub conta na qual você deseja armazenar o código-fonte criado pelo blueprint.

 Note

Se você estiver conectando uma GitHub conta, deverá criar uma conexão pessoal para estabelecer um mapeamento de identidade entre sua CodeCatalyst identidade e sua GitHub identidade. Para obter mais informações, consulte [Conexões pessoais](#) e [Acessando GitHub recursos com conexões pessoais](#).

- Repositórios do Bitbucket: conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, selecione Bitbucket como provedor do repositório e, depois, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo esquema.

- GitLab repositórios: Conecte um GitLab usuário.

Escolha o menu suspenso Avançado, escolha GitLab como provedor do repositório e, em seguida, escolha o GitLab usuário no qual você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Recursos do projeto, configure os parâmetros do esquema. Dependendo do esquema, você pode ter a opção de nomear o repositório de origem.
8. (Opcional) Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, selecione Visualizar código ou Visualizar fluxo de trabalho em Gerar visualização do projeto.
9. (Opcional) Selecione Visualizar detalhes no cartão do esquema para visualizar detalhes específicos sobre o esquema, como uma visão geral da arquitetura do esquema, conexões e permissões necessárias e o tipo de recursos que o esquema cria.
10. Escolha Criar projeto.

Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas

Quando você cria um projeto ou deseja adicionar novos componentes a um projeto existente, você pode não ter certeza sobre qual blueprint usar ou como integrar recursos. CodeCatalyst inclui integração com um assistente generativo de IA chamado Amazon Q, que pode analisar os requisitos do seu projeto e sugerir um plano que melhor atenda às suas necessidades.

Você pode usar o Amazon Q para criar um projeto com um esquema que cria componentes com base em seus requisitos, ou você pode usar o Amazon Q para adicionar um esquema a um projeto existente. Por exemplo, para adicionar recursos para uma aplicação web ou uma aplicação moderna a um projeto, especifique seus requisitos e, depois, os recursos serão adicionados com um esquema recomendado. Problemas com os componentes restantes podem ser criados para você.

O Amazon Q também cria problemas para requisitos que não podem ser resolvidos por meio de um esquema sugerido. Além disso, você pode atribuir esses problemas ao Amazon Q. Se você atribuir o problema ao Amazon Q, ele tentará criar uma solução preliminar para você avaliar. Isso pode ajudar você e sua equipe a se concentrarem e otimizarem o trabalho em questões que exigem sua atenção, enquanto o Amazon Q trabalha em uma solução para problemas que você não tem recursos para resolver imediatamente.

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

Confira a seguir algumas práticas recomendadas para ajudar você a criar projetos e adicionar esquemas com o Amazon Q.

Important

Os recursos de IA generativa estão disponíveis somente na região Oeste dos EUA (Oregon).

- Use os prompts padrão fornecidos pelo Amazon Q. O Amazon Q faz melhor ao escolher esquemas a partir dos prompts fornecidos.
- Use as opções de configuração sugeridas pelo Amazon Q para visualizar os esquemas. Escolha um esquema para visualizar o código de amostra e os recursos que serão criados pelo esquema.
- Use um espaço que esteja habilitado para o Amazon Q. Para criar um projeto com o Amazon Q ou adicionar funcionalidade a um projeto com esquemas usando o Amazon Q, use um espaço habilitado para recursos de IA generativa. Para ter mais informações, consulte [Ativação ou desativação de recursos de IA generativa para um espaço](#).
- Receba mais informações sobre os esquemas recomendados pelo Amazon Q. Talvez você queira saber mais sobre o tipo de recursos do projeto, o código de amostra e os componentes que são criados com um esquema específico recomendado. Para obter mais informações sobre os blueprints disponíveis em CodeCatalyst, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#).
- Permita que o Amazon Q trabalhe com problemas. Permita que o Amazon Q crie problemas para você, atribua esses problemas e os acompanhe. Para obter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#).
- Cancele a atribuição do Amazon Q a problemas que não estão mais resolvidos. Depois de concluir o exemplo, cancele a atribuição do Amazon Q a quaisquer problemas que não estejam mais sendo resolvidos. Se o Amazon Q tiver concluído o trabalho em um problema ou não conseguir encontrar uma solução, cancele a atribuição do Amazon Q para evitar atingir a cota máxima de recursos de IA generativa. Para ter mais informações, consulte [Gerenciamento de recursos de IA generativa](#) e [Preços](#).
- Veja o uso do Amazon Q. Você pode visualizar o uso de recursos de IA generativa no nível do usuário. Acesse Minhas configurações para gerenciar cotas de IA generativa e visualizar o uso pelo ID do builder ou identidade de autenticação única (SSO). Para ter mais informações, consulte [Visualização do uso de recursos de IA generativa em um espaço](#).

Important

Os recursos generativos de IA CodeCatalyst estão sujeitos a cotas. Para ter mais informações, consulte [Preço do Amazon Q Developer](#), [Ativação ou desativação de recursos de IA generativa para um espaço](#) e [Faturamento](#).

Práticas recomendadas para usar esquemas em projetos

Confira a seguir algumas práticas recomendadas para ajudar você a criar um projeto com esquemas ou adicionar esquemas.

- Use os esquemas fornecidos por CodeCatalyst para criar ou adicionar aos projetos. Você pode usar esquemas para criar um projeto completo com código-fonte e recursos para desenvolvedores. Por exemplo, o esquema da aplicação web cria recursos de aplicação e infraestrutura e implanta uma aplicação web. Você pode criar um projeto com um esquema ou adicionar um esquema personalizado a um projeto existente. Para obter mais informações, consulte [Criar um projeto com um esquema](#). Visualize qualquer blueprint CodeCatalyst para visualizar o código de amostra e os recursos que serão criados pelo blueprint.
- Use esquemas personalizados criados pela sua organização. Você pode usar esquemas personalizados para criar um projeto completo em seu espaço. O esquema personalizado desenvolvido por sua organização pode fornecer padronização e práticas recomendadas, o que também pode ajudar a reduzir os esforços para configurar um novo projeto. Como autor de um esquema personalizado, você pode ver detalhes sobre quais projetos estão usando seu esquema em todo o seu espaço. O gerenciamento do ciclo de vida permite que você gerencie centralmente o ciclo de vida de desenvolvimento de software de cada projeto, e os usuários do esquema podem utilizar o gerenciamento do ciclo de vida para regenerar uma base de código a partir de opções ou versões atualizadas de um esquema. Para obter mais informações, consulte [Trabalho com o gerenciamento do ciclo de vida como autor do esquema](#).
- Adicione o perfil de desenvolvedor ou os perfis apropriados do IAM à conta do projeto. Durante ou depois de concluir as etapas de criação do projeto, você pode configurar as permissões do esquema escolhendo ou criando perfis do IAM em uma Conta da AWS que esteja conectada ao espaço.

Adição de recursos e tarefas aos projetos criados

Depois que o projeto estiver pronto, você poderá adicionar recursos e tarefas.

- Para saber mais sobre os CI/CD fluxos de trabalho criados com seu projeto, consulte [Conceitos básicos de fluxos de trabalho](#).
- Para trabalhar com ações de criação semelhantes às do seu novo projeto que implantam artefatos de criação em um bucket do Amazon S3, consulte [Criação com fluxos de trabalho](#) e [Tutorial: Fazer upload de artefatos no Amazon S3](#).

- Para começar com um projeto vazio e trabalhar com a implantação de um aplicativo sem servidor semelhante com uma implantação em CloudFormation pilha, consulte [Tutorial: Implantar uma aplicação sem servidor](#)
- Para adicionar um quadro de planejamento de problemas, consulte [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#).
- Para ver a visão geral do projeto, o status do projeto, a atividade recente da equipe e o trabalho atribuído, consulte [Obtenção de uma lista de projetos](#).
- Para ver o código-fonte ou criar uma solicitação pull, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).
- Para configurar notificações que enviem alertas de status sobre sucesso ou falha na execução do fluxo de trabalho, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).
- Para convidar membros para seu projeto, consulte [Concessão de permissões de projeto aos usuários](#).
- Para configurar ambientes de desenvolvimento, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

Obtenção de uma lista de projetos

No seu CodeCatalyst espaço, você pode ver os detalhes de cada projeto em que você tem permissões de projeto.

Para visualizar um projeto, você deve ser membro do projeto ou ter a função de administrador do espaço.

Se você ainda não criou um projeto, consulte [Como criar um projeto](#). Você deve ter o perfil de Administrador do espaço no espaço em que deseja criar um projeto.

- Na visão geral do projeto, você pode visualizar os membros do projeto, os repositórios de origem, as execuções do fluxo de trabalho, as solicitações pull abertas, os ambientes de desenvolvimento do projeto e os problemas.
- Nas configurações do projeto, você pode visualizar e gerenciar detalhes do projeto, excluir o projeto, convidar novos membros para o projeto, gerenciar membros do projeto e configurar notificações.

Visualização de tarefas do projeto e ambientes de desenvolvimento

Para ver um resumo das tarefas do projeto, como problemas em aberto e pull requests atribuídos a você ou criados por você, e os ambientes de desenvolvimento associados ao projeto, use o console.

Para visualizar um projeto, você deve ser membro do projeto ou ter a função de administrador do espaço.

Para visualizar seus repositórios de origem, execuções de fluxo de trabalho, problemas, pull requests, ambientes de desenvolvimento e problemas

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, selecione Visão geral.
4. Visualize as tarefas do projeto atribuídas a você e criadas por você.
 - Acesse a lista Membros + Exibir tudo para ver uma lista dos membros do projeto.
 - Acesse o cartão Repositórios para ver os repositórios de origem associados ao projeto.
 - Acesse o cartão Execuções do fluxo de trabalho para visualizar os fluxos de trabalho associados ao projeto.
 - Acesse o cartão Solicitações pull abertas para ver um resumo do status do repositório de código, além das solicitações pull atribuídas a você e criadas por você.
 - Acesse o cartão Meus ambientes de desenvolvimento para ver um resumo dos ambientes de desenvolvimento associados ao projeto.
 - Acesse o cartão Problemas para ver um resumo das tarefas atribuídas ou das tarefas que você criou.

Visualização de todos os projetos em um espaço

Na lista Projetos do seu espaço, você pode ver todos os projetos nos quais tem permissões.

Para ver um resumo das tarefas do projeto, como problemas em aberto e pull requests atribuídos a você ou criados por você, e os ambientes de desenvolvimento associados ao projeto, use o console.

Para visualizar um projeto, você deve ser membro do projeto ou ter a função de administrador do espaço.

Para visualizar seus repositórios de origem, execuções de fluxo de trabalho, problemas, pull requests, ambientes de desenvolvimento e problemas

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Visualize o nome, o caminho, o ID do projeto e a descrição.

Visualização das configurações do projeto

Nas Configurações do projeto, você pode visualizar os membros do projeto, os repositórios de origem, as execuções do fluxo de trabalho, as solicitações pull abertas, os ambientes de desenvolvimento do projeto e os problemas.

Para ver um resumo das tarefas do projeto, como problemas em aberto e pull requests atribuídos a você ou criados por você, e os ambientes de desenvolvimento associados ao projeto, use o console.

Para visualizar seus repositórios de origem, execuções de fluxo de trabalho, problemas, pull requests, ambientes de desenvolvimento e problemas

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Visualize o nome, o caminho, o ID do projeto e a descrição.

Mudando para um projeto diferente em CodeCatalyst

Para mudar para um projeto diferente, use o console para escolher em uma lista de projetos aos quais você tem acesso.

Para mudar para um projeto diferente

1. No CodeCatalyst console, escolha o seletor de projetos na parte superior.
2. Expanda o menu suspenso e escolha o projeto para o qual você deseja navegar.

Excluir um projeto

Você pode excluir um projeto para remover todo o acesso aos recursos do projeto. Você deve ter o perfil de Administrador do espaço ou Administrador do projeto para excluir um projeto. Depois que você excluir um projeto, os membros do projeto não conseguirão acessar os recursos do projeto e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

Para excluir o projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha Excluir projeto.
5. Insira **delete** para confirmar a exclusão.
6. Escolha Excluir projeto.

Concessão de permissões de projeto aos usuários

Você pode gerenciar os membros em seus projetos usando o CodeCatalyst console da Amazon. Você pode adicionar ou remover usuários, gerenciar os perfis dos membros atuais, enviar convites para participar do seu projeto e cancelar convites que ainda não foram aceitos.

Na página de membros para usuários do espaço e do projeto, os usuários podem ter vários perfis. Os usuários com vários perfis mostrarão um indicador quando tiverem vários perfis e serão exibidos primeiro com o perfil com mais permissões.

Obtenção de uma lista dos membros e perfis no projeto

Ao adicionar um usuário ao seu projeto, você atribui um perfil que concede permissões ao projeto da seguinte forma:

- O perfil de Administrador do projeto tem todas as permissões em um projeto. Atribua esse perfil somente aos usuários que precisam administrar todos os aspectos de um projeto, incluindo a edição das configurações do projeto, o gerenciamento de permissões do projeto e a exclusão do projeto. Para obter mais informações, consulte [Perfil de administrador do projeto](#).

- O perfil de Colaborador tem as permissões necessárias para trabalhar em um projeto. Atribua esse perfil aos usuários que precisam trabalhar com código, fluxos de trabalho, problemas e ações em um projeto. Para obter mais informações, consulte [Perfil de colaborador](#).
- O perfil de Revisor tem permissões de revisão. Para obter detalhes, consulte [Perfil de revisor](#).
- O perfil Somente leitura tem permissões de leitura. Para obter detalhes, consulte [Perfil somente leitura](#).

Você não precisa convidar um usuário com o perfil de Administrador do espaço para o seu projeto porque ele já tem acesso implícito a todos os projetos no espaço.

Quando você convida um usuário para seu projeto (sem atribuir o perfil de Administrador do espaço), o usuário aparece na tabela Membros do projeto em projetos e na tabela Membros do projeto em espaços.

Para visualizar usuários e perfis em um espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Membros.

A tabela Membros do projeto mostra todos os membros que têm um perfil em um projeto.

 Tip

Se você tiver o perfil de Administrador do espaço, poderá visualizar para quais projetos foi convidado diretamente. Navegue até Configurações do projeto e selecione Meus projetos.

A tabela Administradores do espaço mostra os usuários com o perfil de Administrador do espaço. Esses usuários são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm um perfil em um projeto.

Na coluna Status, estes são valores válidos:

- CodeCatalyst Convidado — enviou o convite, mas o usuário ainda não aceitou ou recusou.

- Membro – O usuário aceitou o convite.

Tópicos

- [Convite de um usuário para um projeto](#)
- [Cancelamento de um convite](#)
- [Remoção de um usuário do projeto](#)
- [Aceitação ou recusa de um convite para um projeto](#)

Convite de um usuário para um projeto

Você pode usar o console para convidar usuários para o seu projeto. Você pode convidar membros do seu espaço ou adicionar nomes de fora do seu espaço.

Para convidar usuários para o seu projeto, você deve estar conectado com o perfil de Administrador do projeto ou Administrador do espaço.

Você não precisa convidar um usuário com o perfil de Administrador do espaço para o seu projeto porque ele já tem acesso implícito a todos os projetos no espaço.

Quando você convida um usuário para seu projeto (sem atribuir o perfil de Administrador do espaço), o usuário aparece na tabela Membros do projeto em projetos e na tabela Membros do projeto em espaços.

Para convidar um membro para o projeto na guia Configurações do projeto

1. Navegue até o projeto.

Tip

Você pode escolher qual projeto deseja visualizar na barra de navegação superior.

2. No painel de navegação, escolha Configurações do projeto.
3. Escolha a guia Membros.
4. Em Membros do projeto, selecione Convidar novo membro.
5. Digite o endereço de e-mail do novo membro, escolha o perfil para esse membro e selecione Convidar. Para ter mais informações sobre funções, consulte [Concessão de acesso com perfis de usuário](#).

Como convidar um membro para o projeto na página Visão geral do projeto

1. Navegue até o projeto.

Tip

Você pode escolher qual projeto deseja visualizar na barra de navegação superior.

2. Escolha o botão Membros +.
3. Digite o endereço de e-mail do novo membro, escolha o perfil para esse membro e selecione Convidar. Para ter mais informações sobre funções, consulte [Concessão de acesso com perfis de usuário](#).

Cancelamento de um convite

Se você enviou um convite recentemente, poderá cancelá-lo, desde que o convite ainda não tenha sido aceito.

Para gerenciar convites de projetos, você deve ter o perfil de Administrador do projeto ou Administrador do espaço.

Para cancelar um convite de membro do projeto

1. Navegue até o projeto para o qual você enviou um convite que deseja cancelar.
2. No painel de navegação, escolha Configurações do projeto.
3. Visualize a guia Membros e verifique se o membro tem o status Convidado.

Note

Você só pode cancelar um convite que ainda não tenha sido aceito.

4. Selecione a opção ao lado da linha com o membro convidado e selecione Cancelar convite.
5. Uma janela de confirmação é exibida. Selecione Cancelar convite para confirmar.

Remoção de um usuário do projeto

Você pode usar o console para remover um usuário do projeto.

Para remover um usuário de projeto, você deve estar conectado com o perfil de Administrador do projeto ou Administrador do espaço.

 Note

Remover um usuário de todos os projetos em um espaço remove automaticamente o usuário desse espaço.

Para remover um usuário de um projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço com o projeto que você deseja visualizar. Em Projetos, escolha seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Membros.
5. Escolha o seletor ao lado do perfil que você deseja remover e escolha selecione Remover.
6. Confirme que você deseja remover o usuário e selecione Remover.

Aceitação ou recusa de um convite para um projeto

Você pode receber um convite por e-mail para participar de um CodeCatalyst projeto da Amazon. Você pode aceitar ou recusar o convite.

Para aceitar ou recusar um convite

1. Abra o e-mail do convite.
2. Escolha o link do projeto no e-mail.
3. Selecione Aceitar ou Recusar.

Se você selecionar Recusar, um e-mail será enviado para a conta de gerenciamento do projeto notificando que você recusou o convite.

Permissão do acesso ao projeto usando equipes

Depois de criar um projeto, você pode criar equipes. As equipes permitem que você agrupe usuários para que eles possam compartilhar permissões e gerenciar projetos, rastreamento de problemas, funções e recursos CodeCatalyst como membros do projeto e do espaço.

Você deve ter o perfil de Administrador do projeto para gerenciar as equipes do seu projeto.

As equipes também são gerenciadas no nível do espaço em CodeCatalyst. Para saber mais sobre equipes em espaços, consulte [Permitir o acesso ao espaço usando equipes](#).

Tópicos

- [Adição de uma equipe a um projeto](#)
- [Concessão de perfis de projeto para uma equipe](#)
- [Remoção de um perfil de projeto para uma equipe](#)

Adição de uma equipe a um projeto

Você pode gerenciar equipes nas quais os membros da equipe podem acessar recursos em seu projeto.

Na página de membros para usuários do espaço e do projeto, os usuários podem ter vários perfis. Os usuários com vários perfis mostrarão um indicador quando tiverem vários perfis e serão exibidos primeiro com o perfil com mais permissões.

Para adicionar uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto. Selecione Configurações do projeto e Equipes.
3. Selecione Adicionar equipe.
4. Em Equipe, escolha uma equipe na lista de equipes disponíveis.
5. Em Função do projeto, escolha uma função na lista de funções do projeto disponíveis em CodeCatalyst.
 - Administrador do projeto: para saber detalhes, consulte [Perfil de administrador do projeto](#).
 - Colaborador: para saber detalhes, consulte [Perfil de colaborador](#).

- Revisor: para saber detalhes, consulte [Perfil de revisor](#).
- Somente leitura: para saber detalhes, consulte [Perfil somente leitura](#).

6. Selecione Adicionar equipe.

Concessão de perfis de projeto para uma equipe

Uma equipe pode ter permissões de perfil, como Usuário avançado, em um espaço. Você pode alterar o perfil do espaço de uma equipe, mas observe que todos os membros da equipe herdarão essas permissões.

Como adicionar ou alterar um perfil do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações do projeto e Equipes.
3. Para alterar um perfil, escolha o seletor ao lado da equipe nessa lista e selecione Alterar perfil. Para adicionar um perfil, selecione Adicionar perfil do projeto. Em Projeto, escolha o projeto que você deseja adicionar e, em Perfil, escolha o perfil. Escolha um dos perfis de projeto disponíveis:
 - Administrador do projeto: para saber detalhes, consulte [Perfil de administrador do projeto](#).
 - Colaborador: para saber detalhes, consulte [Perfil de colaborador](#).
 - Revisor: para saber detalhes, consulte [Perfil de revisor](#).
 - Somente leitura: para saber detalhes, consulte [Perfil somente leitura](#).
4. Escolha Salvar.

Remoção de um perfil de projeto para uma equipe

Em CodeCatalyst, você pode ver as funções do projeto para sua equipe. Você também pode visualizar os membros de uma equipe. Você pode remover o perfil do projeto de uma equipe.

Como remover um perfil do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Selecione Configurações do projeto e Equipes.
3. Escolha a guia Perfis do projeto.
4. Escolha o perfil que deseja remover.

 Important

A remoção de um perfil de uma equipe remove as permissões associadas para todos os usuários da equipe.

5. Escolha Salvar.

Permissão do acesso ao projeto para recursos da máquina

Os recursos da máquina são recursos específicos em CodeCatalyst que são concedidas permissões para projetos ou espaços em CodeCatalyst.

 Note

O termo recurso de máquina não se refere à infraestrutura em nuvem, como uma EC2 instância, mas sim a um plano ou recurso de fluxo de trabalho com permissões para um espaço ou projeto.

Um exemplo de como trabalhar com recursos de máquina em projetos inclui permitir que um recurso de esquema acesse um projeto em seu nome.

Um recurso de máquina representa sua identidade a partir do seu recurso autorizado ao acessar CodeCatalyst por meio de SSO. Os recursos da máquina são usados para conceder permissões aos recursos do projeto, como esquemas e fluxos de trabalho. Você pode visualizar os recursos da máquina no projeto e escolher entre ativar ou desativar os recursos da máquina do projeto. Por exemplo, talvez você queira desativar um recurso da máquina para gerenciar o acesso e reativá-lo posteriormente.

Essas operações estão disponíveis quando um recurso da máquina precisa ser revogado ou desativado. Por exemplo, se você suspeitar que as credenciais podem ter sido comprometidas, poderá desativar o recurso da máquina. Geralmente, essas operações não precisarão ser usadas.

Você deve ter o perfil de Administrador do espaço ou Administrador do projeto para visualizar esta página e gerenciar os recursos da máquina no nível do projeto.

Os recursos da máquina também são gerenciados no nível do espaço em CodeCatalyst. Para saber mais sobre equipes em espaços/projetos, consulte [Permissão do acesso ao espaço para recursos da máquina](#).

Tópicos

- [Visualizar o acesso ao projeto para recursos da máquina](#)
- [Desativação do acesso ao projeto para recursos da máquina](#)
- [Ativação do acesso ao projeto para recursos da máquina](#)

Visualizar o acesso ao projeto para recursos da máquina

Você pode ver uma lista dos recursos da máquina que estão em uso no seu projeto.

Você deve ter o perfil de Administrador do espaço ou Administrador do projeto.

Para visualizar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e selecione Configurações do projeto. Selecione Recursos da máquina.
3. No menu suspenso, selecione Ação de fluxo de trabalho para visualizar somente os recursos da máquina para fluxos de trabalho. Selecione Esquema para visualizar somente os recursos da máquina para esquemas.

Você também pode filtrar por um nome usando o campo Filtro.

Desativação do acesso ao projeto para recursos da máquina

Você pode optar por desativar os recursos da máquina que estão em uso no seu projeto.

Important

A desativação dos recursos da máquina removerá todas as permissões de todos os esquemas ou fluxos de trabalho associados no espaço.

Você deve ter o perfil de Administrador do espaço ou Administrador do projeto.

Para desabilitar recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e selecione Configurações do projeto. Selecione Recursos da máquina.
3. Escolha uma das opções a seguir.

Important

A desativação dos recursos da máquina removerá todas as permissões de todos os esquemas ou fluxos de trabalho associados no espaço.

- Para desabilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja desabilitar. Selecione Desabilitar e Este recurso.
- Para desabilitar todos os recursos, selecione Desabilitar e, depois, Todos os recursos.
- Para desabilitar todas as ações do fluxo de trabalho, selecione Desabilitar e, depois, Todas as ações do fluxo de trabalho.
- Para desabilitar todos os esquemas, selecione Desabilitar e, depois, Todos os esquemas.

Ativação do acesso ao projeto para recursos da máquina

Você pode optar por ativar os recursos da máquina que estão em uso no seu projeto e que foram desativados.

Você deve ter o perfil de Administrador do espaço ou Administrador do projeto.

Para habilitar recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e selecione Configurações do projeto. Selecione Recursos da máquina.
3. Escolha uma das opções a seguir.
 - Para habilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja habilitar. Selecione Habilitar e Este recurso.
 - Para habilitar todos os recursos, selecione Habilitar e, depois, Todos os recursos.
 - Para habilitar todas as ações do fluxo de trabalho, selecione Habilitar e, depois, Todas as ações do fluxo de trabalho.

- Para habilitar todos os esquemas, selecione Habilitar e, depois, Todos os esquemas.

Cotas para projetos

A tabela a seguir descreve cotas e limites para projetos na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

Número máximo de projetos por espaço	100
Número máximo de projetos aos quais um usuário pode pertencer	1.000
Número máximo de membros que podem pertencer a um projeto.	10.000
Nomes de projetos	Os nomes de projeto devem ser exclusivos em um espaço. Os nomes devem ter entre 3 e 63 caracteres. Os nomes diferenciam letras maiúsculas de minúsculas. Os nomes de projeto devem começar com um caractere alfanumérico. Caracteres válidos: A-Z, a-z, 0-9, espaços e . , _ (sublinhado) - (hífen) Os nomes de projeto não podem conter nenhum dos seguinte caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Descrições de projetos	As descrições de projeto podem ter até 200 caracteres. Caracteres válidos: A-Z, a-z, 0-9, espaços e . , _ (sublinhado) - (hífen). As descrições de projetos são opcionais.

Enviando notificações de CodeCatalyst

Você pode configurar notificações para monitorar seus projetos e recursos no CodeCatalyst. Os usuários podem escolher os eventos do projeto sobre os quais desejam receber e-mails em

qualquer projeto do qual sejam membros. Você também pode optar por configurar as notificações enviadas para uma equipe inteira em um aplicativo de mensagens de equipe, como o Slack, configurando o acesso entre um CodeCatalyst espaço e um espaço de trabalho do Slack e, em seguida, configurando as notificações para que um projeto seja enviado para um ou mais canais nesse espaço de trabalho do Slack. Depois de configurar o acesso entre um CodeCatalyst espaço e um espaço de trabalho do Slack, os membros do projeto também terão a opção de adicionar seu próprio membro do Slack para que IDs possam ser notificados diretamente sobre CodeCatalyst eventos nos espaços de trabalho e canais conectados do Slack.

Note

O conjunto de eventos do projeto que podem ser enviados para o Slack não é igual ao conjunto de eventos sobre os quais os usuários podem escolher serem notificados por e-mail.

Tópicos

- [Como as notificações funcionam?](#)
- [Conceitos básicos das notificações do Slack](#)
- [Envio de notificações do Slack e por e-mail do CodeCatalyst](#)

Como as notificações funcionam?

Você pode configurar o projeto para fornecer notificações à aplicação de mensagens da equipe, como o Slack.

Quais permissões são necessárias para notificações?

Qualquer membro do projeto pode definir, visualizar, atualizar ou excluir as configurações de notificação de um canal no CodeCatalyst. No entanto, somente usuários com o perfil de Administrador do espaço podem adicionar ou excluir espaços de trabalho do Slack. Todos os usuários podem configurar sobre quais eventos do projeto desejam receber e-mails para os projetos aos quais pertencem no CodeCatalyst.

Sobre quais eventos do CodeCatalyst posso configurar notificações?

Você pode configurar o CodeCatalyst para enviar notificações a um ou mais canais do Slack sobre eventos do fluxo de trabalho. Depois que as notificações são configuradas entre um projeto do CodeCatalyst e o Slack, os usuários do projeto podem optar por adicionar seu próprio ID de membro do Slack para receber mensagens diretas nos canais do Slack sobre eventos do CodeCatalyst. Os usuários que adicionarem IDs de membro do Slack receberão menções diretas aos IDs nos canais do Slack configurados para os projetos, ajudando a aumentar a conscientização sobre os eventos que lhes interessam.

Você também pode escolher sobre quais eventos deseja receber e-mails. Esses e-mails são enviados para o endereço de e-mail configurado para o ID do builder AWS.

Como as notificações são exibidas?

Você pode configurar o CodeCatalyst para enviar notificações a um ou mais canais do Slack. Você precisa autorizar o CodeCatalyst a conceder permissões para acessar seu espaço de trabalho do Slack. Depois que a autorização é fornecida, o CodeCatalyst pode enviar notificações aos canais do Slack configurados. Se um membro do projeto optar por adicionar o ID de membro do Slack, ele poderá receber menções sobre eventos do CodeCatalyst nos canais do Slack configurados para esse projeto.

Como configurar notificações?

As notificações por e-mail são configuradas como parte do CodeCatalyst. Os usuários do projeto podem escolher sobre quais eventos gostariam de receber e-mails na página *Minhas configurações*.

Para configurar as notificações do Slack para os recursos do projeto, você deve concluir as tarefas de alto nível a seguir.

Para configurar notificações (tarefas de alto nível)

1. No CodeCatalyst, você configura uma conexão entre o CodeCatalyst e um cliente de mensagens, como o Slack. Depois que um espaço de trabalho do Slack estiver conectado, ele estará disponível para todos os projetos no espaço.

Note

Somente usuários com o perfil de Administrador do espaço podem adicionar ou excluir um espaço de trabalho do Slack.

2. No projeto no CodeCatalyst, adicione o canal em que você deseja que sua equipe receba notificações.
3. No CodeCatalyst, você ativa as notificações para vários eventos, como falha na execução do fluxo de trabalho, e especifica o canal para o qual deseja que elas sejam enviadas.

Para saber detalhes das etapas, consulte [Conceitos básicos das notificações do Slack](#).

Depois que as notificações forem configuradas entre um espaço do CodeCatalyst e o Slack, os usuários poderão optar por adicionar seus próprios IDs de membro do Slack para receber mensagens diretas sobre os eventos do CodeCatalyst nos canais do Slack configurados para os projetos.

Conceitos básicos das notificações do Slack

Depois de criar um projeto, é possível configurar as notificações do Slack que ajudam a equipe a monitorar os recursos do projeto.

Essas etapas orientam você a configurar as notificações do Slack pela primeira vez em CodeCatalyst. Se você já configurou as notificações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

Note

O conjunto de eventos do projeto que podem ser enviados para canais de notificação não é igual ao conjunto de eventos sobre os quais os usuários podem escolher serem notificados por e-mail. Para obter mais informações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: conecte-se CodeCatalyst ao seu workspace Slack](#)
- [Etapa 2: adicione seu canal do Slack ao CodeCatalyst](#)
- [Etapa 3: testar as notificações do CodeCatalyst Slack](#)
- [Etapa 4: Próximas etapas](#)

Pré-requisitos

Antes de começar, você precisará fazer o seguinte:

- Um CodeCatalyst espaço. Para obter informações sobre como criar um CodeCatalyst espaço e fazer login pela primeira vez, consulte [Configuração e login no CodeCatalyst](#).
- Um CodeCatalyst projeto. Para obter mais informações, consulte [Como criar um projeto](#).
- Uma CodeCatalyst conta com a função de administrador do projeto ou administrador do espaço. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).
- Uma conta do Slack e um espaço de trabalho do Slack que podem ser acessados pelo CodeCatalyst
- Um canal do Slack onde CodeCatalyst enviará notificações. O canal pode ser público ou privado.

Etapa 1: conecte-se CodeCatalyst ao seu workspace Slack

Somente usuários com o perfil de Administrador do espaço podem adicionar ou excluir espaços de trabalho do Slack. A adição ou exclusão de um espaço de trabalho do Slack afeta todos os projetos no espaço. Para estabelecer a conexão entre o Slack CodeCatalyst e o Slack, CodeCatalyst realize um handshake de OAuth autenticação seguro com seu espaço de trabalho do Slack.

Use as instruções a seguir para se conectar CodeCatalyst ao seu workspace Slack.

Note

Isso só precisa ser feito uma vez para cada espaço de trabalho do Slack. Depois, você pode configurar as notificações por canal do Slack.

Para se conectar CodeCatalyst ao seu espaço de trabalho do Slack

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Notificações.
5. Selecione Configurar notificações.

6. Selecione Conectar ao espaço de trabalho do Slack.
7. Leia o conteúdo da caixa de diálogo e selecione Conectar ao espaço de trabalho do Slack.
8. Em uma mensagem do Amazon Q Developer em aplicações de chat:
 - a. No canto superior direito, escolha o espaço de trabalho do Slack que contém seu canal.
 - b. Selecione Permitir.

Você retornará ao CodeCatalyst console.

9. Avance para [Etapa 2: adicione seu canal do Slack ao CodeCatalyst](#).

Etapa 2: adicione seu canal do Slack ao CodeCatalyst

Você precisa do ID do canal do Slack para adicionar seu canal. CodeCatalyst

Como ter o ID do canal do Slack

1. Faça login no Slack. Para ter mais informações, consulte [Fazer login no Slack](#).
2. Acesse o espaço de trabalho do Slack que contém o canal para o qual você quer que as notificações sejam enviadas. Para ter mais informações, consulte [Alternar entre espaços de trabalho do Slack](#) ou [Fazer login em outros espaços de trabalho do Slack](#).
3. No painel de navegação, abra o menu de contexto (clique com o botão direito do mouse) do canal para o qual deseja que as notificações sejam enviadas e selecione Abrir detalhes do canal.

O ID do canal é exibido na parte inferior da caixa de diálogo.

4. Copie o valor de ID do canal. Você precisará dele na próxima etapa.

Usando o ID do canal que você acabou de copiar, agora você pode conectar seu canal do Slack ao CodeCatalyst

Para adicionar seu canal do Slack ao CodeCatalyst

1. Antes de começar, se o canal do Slack for privado, adicione o aplicativo Amazon Q Developer em aplicações de chat canal ao da seguinte maneira:
 - a. Na caixa de mensagem do canal do Slack, insira **@aws** e escolha o aplicação aws na caixa de diálogo.

b. Pressione Enter.

Uma mensagem do Slackbot é exibida, indicando que o Amazon Q Developer em aplicações de chat não está no canal privado.

c. Escolha Convidá-los para convidar o Amazon Q Developer em aplicações de chat para o canal.

2. No CodeCatalyst console, escolha Avançar.
3. Em ID do canal, cole o ID do canal do Slack que você obteve anteriormente.
4. Em Nome do canal, insira um nome. É recomendável usar o nome do canal do Slack.
5. Escolha Próximo.
6. Em Selecionar eventos de notificação, escolha o tipo de evento para o qual você deseja receber notificações.
7. Escolha Terminar.

Etapa 3: testar as notificações do CodeCatalyst Slack

Depois que seu projeto estiver configurado para enviar notificações sobre o status do fluxo de trabalho, você poderá ver as notificações no Slack.

Para visualizar as notificações no Slack

1. Em seu CodeCatalyst projeto, [inicie um fluxo de trabalho manualmente](#) para concluir a execução do fluxo de trabalho e receber uma notificação de status quando a execução terminar.
2. No Slack, veja o canal que você configurou para receber notificações. As notificações mostram o status mais recente de cada execução do fluxo de trabalho e se ela falhou ou foi bem-sucedida.

Etapa 4: Próximas etapas

Depois que um espaço de trabalho do Slack estiver configurado para seu CodeCatalyst espaço, você poderá adicionar outros canais do Slack aos CodeCatalyst projetos existentes e adicioná-los aos novos projetos depois de criá-los. Você também pode informar aos usuários do projeto que eles podem configurar notificações pessoais do Slack para seus membros IDs do Slack e configurar os eventos para os quais eles receberão e-mails. Para obter mais informações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

Envio de notificações do Slack e por e-mail do CodeCatalyst

Você pode configurar o CodeCatalyst para enviar notificações sobre os eventos ocorridos no projeto. O CodeCatalyst pode enviar notificações para clientes de mensagens, como canais do Slack. Fazer com que o CodeCatalyst envie mensagens para os canais do Slack ajuda a garantir que toda a sua equipe esteja ciente de eventos importantes, como falhas no fluxo de trabalho. Se desejar, você pode optar por fazer o CodeCatalyst mencionar você nas mensagens do Slack que ele envia para que você receba uma mensagem direta (DM) correspondente.

O CodeCatalyst também pode enviar notificações diretamente para você em um e-mail. Notificações por e-mail serão enviadas sobre eventos em qualquer projeto do qual você seja membro. Esses e-mails serão enviados para o endereço de e-mail configurado no ID do builder AWS.

Note

Os eventos que podem ser enviados para os canais do Slack podem ser diferentes daqueles enviados por e-mail.

Tópicos

- [Configuração de notificações de e-mail](#)
- [Envio de notificações para canais do Slack](#)
- [Configuração de mensagens diretas do Slack](#)
- [Editar as notificações para um canal de notificação](#)
- [Remoção de um canal](#)

Configuração de notificações de e-mail

Você pode optar por receber notificações por e-mail sobre eventos em qualquer projeto do qual você seja membro. Esses e-mails serão enviados para o endereço de e-mail configurado no ID do builder AWS. Por padrão, você receberá e-mails sobre todos os eventos do projeto para os quais os e-mails podem ser enviados.

Como configurar notificações por e-mail para eventos do projeto

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.

2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página Minhas configurações do CodeCatalyst é aberta.

 Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Notificações por e-mail, encontre o projeto na lista em que você deseja configurar as notificações por e-mail e selecione Editar.
4. Selecione os eventos para os quais quer receber e-mails e escolha Salvar.

Envio de notificações para canais do Slack

Você pode configurar o CodeCatalyst para enviar notificações sobre eventos do projeto para os canais do Slack da sua equipe. Ao fazer isso, você pode ajudar a garantir que toda a equipe esteja ciente de eventos importantes, como quando a execução de um fluxo de trabalho falha.

 Note

Qualquer membro de um projeto pode gerenciar as notificações enviadas aos canais desse projeto. No entanto, somente usuários com o perfil de Administrador do espaço podem adicionar ou excluir espaços de trabalho do Slack.

Use as instruções a seguir para adicionar um canal do Slack para o qual as notificações serão enviadas.

Como adicionar um canal do Slack para notificações

1. Se você estiver adicionando seu primeiro canal do Slack, consulte [Conceitos básicos das notificações do Slack](#).

Depois de configurar seu primeiro canal, retorne a este procedimento para configurar canais adicionais.

2. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
3. Navegue até o projeto.
4. No painel de navegação, escolha Configurações do projeto.

5. Escolha a guia Notificações.
6. Escolha Add channel (Adicionar canal).
7. Selecione Escolher espaço de trabalho e, depois, selecione o espaço de trabalho do Slack que contém o canal para o qual você deseja enviar notificações.

Se o seu espaço de trabalho do Slack não estiver na lista, você poderá adicioná-lo seguindo as instruções descritas em [Conceitos básicos das notificações do Slack](#).

8. Antes de inserir um ID do canal, se o canal do Slack que você deseja adicionar for privado, conclua estas etapas:
 - a. Na caixa de mensagem do canal do Slack, insira **@aws** e escolha a aplicação aws no menu pop-up.
 - b. Pressione Enter.

Uma mensagem do Slackbot é exibida, indicando que o Amazon Q Developer em aplicações de chat não está no canal privado.

- c. Escolha Convidá-los para convidar o Amazon Q Developer em aplicações de chat para o canal.
9. No campo ID do canal do CodeCatalyst, insira o ID do canal do Slack. Para encontrar o ID, acesse o Slack e, no painel de navegação, clique com o botão direito do mouse no canal e selecione Abrir detalhes do canal.

O ID do canal é exibido na parte inferior da caixa de diálogo.

10. Em Nome do canal, insira um nome. É recomendável usar o nome do canal do Slack.
11. Em Selecionar eventos de notificação, escolha o tipo de evento para o qual você deseja receber notificações.
12. Escolha Adicionar.

Configuração de mensagens diretas do Slack

Se um projeto do CodeCatalyst tiver sido configurado para [enviar notificações para um canal do Slack](#), essas notificações também poderão ser enviadas como mensagens diretas (DMs). Ter notificações enviadas diretamente para você como DMs pode ajudar a aumentar a conscientização sobre os eventos que acontecem nos projetos em que você tem um perfil. Para habilitar DMs, você deve adicionar seu ID de membro do Slack ao CodeCatalyst.

Para configurar mensagens diretas do Slack

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página Minhas configurações do CodeCatalyst é aberta.

Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Notificações pessoais do Slack, selecione Conectar ID do Slack e, depois, Conectar ao espaço de trabalho do Slack. Uma janela separada será aberta.

Tip

Essa opção não é configurável, a menos que um usuário com o perfil de Administrador do espaço tenha adicionado um espaço de trabalho do Slack ao seu espaço do CodeCatalyst. Para obter mais informações, consulte [Conceitos básicos das notificações do Slack](#) e [Envio de notificações para canais do Slack](#).

4. Na janela de solicitação de permissões, verifique se o nome do espaço de trabalho corresponde ao espaço de trabalho do Slack configurado para o espaço do CodeCatalyst. Escolha Permitir para permitir que o Amazon Q Developer em aplicações de chat acesse o espaço de trabalho. A janela será fechada e o espaço de trabalho do Slack mostrará o Status da conexão como Conectada.

Tip

Se o status da conexão não mudar, confira se ocorreu um erro ao conectar o espaço de trabalho do Slack. Talvez seja necessário rolar para cima para ver o erro.

5. Para deixar de receber notificações pessoais do Slack, escolha o espaço de trabalho conectado do Slack e, depois, selecione Desconectar ID do Slack.

Editar as notificações para um canal de notificação

Você pode alterar os canais para os quais as notificações vão e desativar completamente notificações específicas.

Como editar notificações

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Notificações.
5. Selecione Editar notificações.
6. Execute um destes procedimentos:
 - Para enviar uma notificação para um canal específico, escolha o canal na lista suspensa.
 - Para desativar uma notificação globalmente, escolha o botão ao lado da notificação.
 - Para parar de enviar uma notificação para um canal específico, selecione o X no canal.
7. Escolha Salvar.

Remoção de um canal

Você pode remover um canal do Slack do Amazon CodeCatalyst. Ao remover um canal do Slack, as notificações sobre um projeto do CodeCatalyst escolhido não serão mais enviadas para o canal.

Para remover um canal

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto. No painel de navegação, escolha Configurações do projeto.
3. Na página Configurações do projeto, escolha a guia Notificações.
4. Escolha o indicador ao lado do canal que deseja remover e selecione Remover canal. Na janela de confirmação, escolha OK.

Configure CodeCatalyst projetos com plantas

Os blueprints são geradores de código arbitrários que representam um componente arquitetônico de um projeto. CodeCatalyst O componente pode consistir em qualquer coisa, desde um fluxo de trabalho em um único arquivo até o projeto completo com código de amostra. Os esquemas usam um conjunto arbitrário de opções para gerar um conjunto arbitrário de código de saída que é encaminhado para um projeto. Conforme o esquema é atualizado com as práticas recomendadas mais recentes ou novas opções, ele pode gerar novamente as partes relevantes da sua base de código em projetos que contenham esse esquema.

Você pode usar um CodeCatalyst plano da Amazon para criar um projeto completo com um repositório de origem, exemplo de código-fonte, CI/CD fluxos de trabalho, relatórios de criação e teste e ferramentas integradas de rastreamento de problemas. Um CodeCatalyst blueprint gera recursos e código-fonte com base no conjunto de parâmetros de configuração. Ao usar um blueprint CodeCatalyst gerenciado, o blueprint que você escolhe determina quais recursos são adicionados ao seu projeto, bem como as ferramentas que ele CodeCatalyst cria ou configura, para que você possa acompanhar e usar os recursos do seu projeto. Como usuário do blueprint, você pode criar um projeto com um blueprint ou adicioná-lo a um projeto existente CodeCatalyst . É possível adicionar vários esquemas ao projeto, e cada um pode ser aplicado como um componente independente. Por exemplo, é possível ter um projeto que foi criado com um esquema de aplicação web e adicionar um esquema de segurança posteriormente. Quando um dos esquemas é atualizado, é possível incorporar as alterações ou as correções ao projeto por meio do gerenciamento do ciclo de vida. Para obter mais informações, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#) e [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#).

Como autor do blueprint, você também pode criar e publicar blueprints personalizados para os membros do seu CodeCatalyst espaço usarem os recursos do seu projeto. Os esquemas personalizados podem ser desenvolvidos para atender às necessidades específicas dos projetos do espaço. Depois de adicionar um esquema personalizado ao catálogo do espaço, você pode gerenciar o esquema e continuar fazendo atualizações para que os projetos do espaço permaneçam atualizados com as práticas recomendadas mais recentes. Para obter mais informações, consulte [Padronizando projetos com plantas personalizadas em CodeCatalyst](#). [Para ver o SDK de blueprints e exemplos de blueprints, consulte o repositório de código aberto. GitHub](#)

Talvez você já tenha a padronização e as práticas recomendadas implementadas. Em vez de criar e desenvolver um esquema personalizado do zero, você pode optar por converter um repositório de

origem existente com código-fonte em um esquema personalizado. Para obter mais informações, consulte [Converter repositórios de origem em esquemas personalizados](#).

Tópicos

- [Criar um projeto com um esquema](#)
- [Adicionar um esquema a um projeto para integrar recursos](#)
- [Desassociar um esquema de um projeto para interromper as atualizações](#)
- [Alteração das versões do esquema em um projeto](#)
- [Edição de uma descrição para um esquema em um projeto](#)
- [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#)
- [Criação de um projeto abrangente com CodeCatalyst plantas](#)
- [Padronizando projetos com plantas personalizadas em CodeCatalyst](#)
- [Cotas para plantas em CodeCatalyst](#)

Criar um projeto com um esquema

Você pode criar rapidamente um projeto usando um blueprint do catálogo de CodeCatalyst blueprints da Amazon ou do catálogo espacial da sua equipe com blueprints personalizados. Dependendo do esquema, seu projeto é criado com recursos específicos. Você também pode colaborar com o Amazon Q, um assistente de IA generativa, ao criar novos projetos ou adicionar componentes a projetos existentes. Você pode fornecer ao Amazon Q os requisitos do projeto interagindo com ele em uma interface semelhante a um bate-papo. Com base em seus requisitos, o Amazon Q sugere um esquema e também descreve os requisitos que não podem ser atendidos. Você pode continuar com a sugestão do Amazon Q se estiver satisfeito, e ele criará os recursos necessários, como um repositório de origem com código para seu requisito. Para obter mais informações, consulte [Criar um projeto com um esquema](#), [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#) e [Criação de um projeto abrangente com CodeCatalyst plantas](#).

Depois de criar um projeto, você pode adicionar outras plantas ao seu CodeCatalyst projeto a partir do CodeCatalyst catálogo ou do catálogo do seu espaço com plantas personalizadas. Os esquemas representam componentes arquitetônicos, portanto, vários esquemas podem ser usados juntos em seu projeto para incorporar as práticas recomendadas da equipe. Isso também permite garantir que o projeto esteja atualizado com as alterações mais recentes nos componentes em evolução.

Para saber mais sobre como trabalhar com esquemas em seu projeto, consulte [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#).

Adicionar um esquema a um projeto para integrar recursos

É possível adicionar vários esquemas a um projeto para incorporar componentes funcionais, recursos e governança. Seus projetos podem comportar vários elementos que são gerenciados de forma independente em esquemas separados. Adicionar esquemas a um projeto reduz a necessidade de criar recursos manualmente e tornar os componentes de software funcionais. Seus projetos também podem se manter atualizados conforme os requisitos evoluem. Para saber mais sobre como adicionar esquemas ao projeto, consulte [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#).

Ao configurar os detalhes de um esquema, você também pode optar por armazenar o código-fonte do esquema em um repositório de terceiros preferencial, onde ainda é possível gerenciar o esquema e utilizar os recursos de gerenciamento do ciclo de vida para manter o projeto atualizado. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#).

Important

Para adicionar um blueprint ao seu CodeCatalyst projeto, você deve estar conectado com uma conta que tenha a função de administrador do espaço, usuário avançado ou administrador do projeto no espaço.

Tip

Depois de adicionar um esquema ao projeto, você pode configurar as notificações por e-mail e do Slack para fornecer atualizações sobre as alterações mais recentes no esquema. Para obter mais informações, consulte [Enviando notificações de CodeCatalyst](#).

Como adicionar um esquema ao projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço e escolha o projeto ao qual você deseja adicionar um blueprint.

3. No painel de navegação, escolha Esquemas e Adicionar esquema.

 Tip

Você pode optar por adicionar um esquema fornecendo ao Amazon Q os requisitos do projeto para que o Amazon Q sugira um esquema. Para obter mais informações, consulte [Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade](#) e [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#). Esse recurso está disponível somente na região do Oeste dos EUA (Oregon).

Essa funcionalidade exige que recursos de IA generativa estejam habilitados para o espaço. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

4. Escolha um blueprint na guia CodeCatalyst Blueprints ou um blueprint personalizado na guia Space blueprints e, em seguida, escolha Avançar.
5. Em Detalhes do esquema, escolha uma versão do esquema no menu suspenso Versão de destino. A versão mais recente do catálogo é selecionada automaticamente.
6. (Opcional) Por padrão, o código-fonte criado pelo blueprint é armazenado em um CodeCatalyst repositório. Também é possível armazenar o código-fonte do esquema em um repositório de terceiros. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você deseja usar:

- GitHub repositórios: Conecte uma GitHub conta.

Escolha o menu suspenso Avançado, escolha GitHub como provedor do repositório e, em seguida, escolha a GitHub conta na qual você deseja armazenar o código-fonte criado pelo blueprint.

 Note

Se você estiver usando uma conexão com uma GitHub conta, deverá criar uma conexão pessoal para estabelecer o mapeamento de identidade entre sua CodeCatalyst identidade e sua GitHub identidade. Para obter mais informações, consulte [Conexões pessoais](#) e [Acessando GitHub recursos com conexões pessoais](#).

- Repositórios do Bitbucket: conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, selecione Bitbucket como provedor do repositório e, depois, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo esquema.

- GitLab repositórios: Conecte um GitLab usuário.

Escolha o menu suspenso Avançado, escolha GitLab como provedor do repositório e, em seguida, escolha o GitLab usuário no qual você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Configurar esquema, configure os parâmetros do esquema. Dependendo do esquema, você pode ter a opção de nomear o repositório de origem.
8. Revise as diferenças entre a versão atual do esquema e sua versão atualizada. A diferença exibida em uma solicitação pull mostra as alterações entre a versão atual e a mais recente, que é a versão desejada no momento em que a solicitação pull foi criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão tanto para a atual quanto para a versão desejada.
9. Quando a solicitação pull tiver o código e as alterações que você deseja revisar, escolha Adicionar esquema. Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos, bem como à solicitação pull geral. Você pode adicionar links a recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

Note

O esquema não será aplicado até que a solicitação pull seja aprovada e mesclada. Para obter mais informações, consulte [Revisão de uma solicitação pull](#) e [Mesclar uma solicitação pull](#).

Os autores do esquema também podem adicionar um esquema personalizado a projetos em espaços específicos que não tenham o esquema disponível para criar projetos ou adicionar a projetos existentes. Para obter mais informações, consulte [Publicação e adição de um esquema personalizado em espaços e projetos especificados](#).

Se você não quiser mais receber atualizações para um esquema, desassocie o esquema do projeto. Para obter mais informações, consulte [Desassociar um esquema de um projeto para interromper as atualizações](#).

Desassociar um esquema de um projeto para interromper as atualizações

Se você não quiser novas atualizações de um esquema, desassocie o esquema do seu projeto. Os recursos e componentes funcionais de software adicionados ao seu projeto a partir do esquema permanecerão em seu projeto.

Important

Para desassociar um blueprint do seu CodeCatalyst projeto, você deve estar conectado com uma conta que tenha a função de administrador do espaço, usuário avançado ou administrador do projeto no espaço.

Para desassociar um esquema do seu projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço e escolha o projeto em que você deseja desassociar um blueprint.
3. No painel de navegação, escolha Esquemas.
4. Escolha o esquema com os recursos que você deseja desassociar, escolha o menu suspenso Ações e, depois, selecione Desassociar esquema.
5. Insira `confirm` para confirmar a desassociação.
6. Escolha Confirmar.

Alteração das versões do esquema em um projeto

Se você criou um projeto com um esquema ou adicionou um esquema a um projeto existente, você será notificado sobre novas versões do esquema. Antes que a versão do esquema seja atualizada por meio de uma solicitação pull aprovada, você pode visualizar as alterações no código e os ambientes afetados. O gerenciamento do ciclo de vida permite que você altere as versões de um

ou mais esquemas aplicados em seu projeto, para que cada versão do esquema possa ser alterada sem afetar outras áreas do seu projeto. Você também pode substituir as atualizações do esquema. Para obter mais informações, consulte [Trabalho com o gerenciamento do ciclo de vida como usuário do esquema](#).

 Important

Para alterar a versão de um blueprint em seu CodeCatalyst projeto, você deve estar conectado com uma conta que tenha a função de administrador do espaço, usuário avançado ou administrador do projeto no espaço.

 Tip

Depois de adicionar um esquema ao projeto, você pode configurar as notificações por e-mail e do Slack para fornecer atualizações sobre as alterações mais recentes no esquema. Para obter mais informações, consulte [Enviando notificações de CodeCatalyst](#).

Para atualizar um esquema para a versão mais recente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja atualizar a versão de um blueprint.
3. No painel do espaço, escolha o projeto com o esquema que deseja atualizar.
4. No painel de navegação, escolha Esquemas.
5. Na coluna Status, escolha o link para alterar a versão do catálogo (por exemplo, Alterar versão 0.3.109 do catálogo).
6. Escolha o menu suspenso Ações e selecione Atualizar versão. A versão mais recente é selecionada automaticamente.
7. (Opcional) Em Configurar esquema, configure os parâmetros do esquema.
8. (Opcional) Na guia Alterações de código, revise as diferenças entre a versão atual do esquema e a versão atualizada. A diferença exibida em uma solicitação pull são as alterações entre a versão atual e a versão mais recente, que é a versão desejada no momento em que a solicitação pull é criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão tanto para a atual quanto para a versão desejada.

- Quando a solicitação pull tiver o código e as alterações que você deseja revisar, selecione Aplicar atualização. Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos e à solicitação pull geral. Você pode adicionar links a recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

 Note

O esquema não será atualizado até que a solicitação pull seja aprovada e mesclada. Para obter mais informações, consulte [Revisão de uma solicitação pull](#) e [Mesclar uma solicitação pull](#).

 Note

Se você tiver solicitações pull existentes abertas para atualizar uma versão do esquema, feche as solicitações pull anteriores antes de criar outra. Ao selecionar Atualizar versão, você será direcionado para a lista de solicitações pull pendentes para o esquema. Você também pode visualizar solicitações pull pendentes na guia Esquemas nas Configurações do projeto e na página de resumo do projeto. Para obter mais informações, consulte [Visualizar solicitações pull](#).

Para alterar a versão de um esquema

- Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
- No CodeCatalyst console, navegue até o espaço em que você deseja atualizar a versão de um blueprint.
- No painel do espaço, escolha o projeto com o esquema que deseja atualizar.
- No painel de navegação, selecione Esquemas e selecione o botão de seleção do esquema que deseja atualizar.
- Escolha o menu suspenso Ações e selecione Configurar esquema.
- No menu suspenso Versão de destino, selecione a versão que você deseja usar. A versão mais recente é selecionada automaticamente.
- (Opcional) Em Configurar esquema, configure os parâmetros do esquema.

8. (Opcional) Na guia Alterações de código, revise as diferenças entre a versão atual do esquema e a versão atualizada. A diferença exibida em uma solicitação pull são as alterações entre a versão atual e a versão mais recente, que é a versão desejada no momento em que a solicitação pull é criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão tanto para a atual quanto para a versão desejada.
9. Quando a solicitação pull tiver o código e as alterações que você deseja revisar, selecione Aplicar atualização. Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos e à solicitação pull geral. Você pode adicionar links a recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

 Note

O esquema não será atualizado até que a solicitação pull seja aprovada e mesclada. Para obter mais informações, consulte [Revisão de uma solicitação pull](#) e [Mesclar uma solicitação pull](#).

 Note

Se você tiver solicitações pull existentes abertas para atualizar uma versão do esquema, feche as solicitações pull anteriores antes de criar outra. Ao selecionar Atualizar versão, você será direcionado para a lista de solicitações pull pendentes para o esquema. Você também pode visualizar solicitações pull pendentes na guia Esquemas nas Configurações do projeto e na página de resumo do projeto. Para obter mais informações, consulte [Visualizar solicitações pull](#).

Edição de uma descrição para um esquema em um projeto

Você pode editar a descrição de um esquema usado para criar um projeto ou aplicativo após a criação de um projeto. Um esquema pode ser usado mais de uma vez em um projeto. Para diferenciar a finalidade dos esquemas em seu projeto, você pode usar descrições para eles. As descrições também podem ser usadas para identificar os componentes que você está adicionando de um esquema específico.

 Important

Para editar a descrição de um esquema personalizado em seu espaço, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço, Usuário avançado ou Administrador do projeto no espaço.

Para editar a descrição de um esquema em seu projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até seu espaço e escolha o projeto com as configurações do blueprint que você deseja atualizar.
3. No painel de navegação, escolha Esquemas.
4. Selecione o esquema com a descrição que deseja atualizar, escolha o menu suspenso Ações e selecione Configurações.
5. No campo de entrada de texto Descrição do esquema, insira uma descrição para identificar o esquema em seu projeto.
6. Escolha Salvar.

Trabalho com o gerenciamento do ciclo de vida como usuário do esquema

O gerenciamento do ciclo de vida é a capacidade de gerar novamente uma base de código a partir de opções ou versões atualizadas de um esquema. Isso permite que um autor do esquema gerencie centralmente o ciclo de vida de desenvolvimento de software de cada projeto que contém um esquema específico. Por exemplo, enviar uma correção de segurança para um esquema de aplicação web permitirá que cada projeto contendo ou criado a partir do esquema de aplicação web receba essa correção automaticamente. Essa mesma estrutura de gerenciamento também permite que você, como usuário do esquema, altere as opções do esquema após elas terem sido selecionadas.

Tópicos

- [Usar o gerenciamento do ciclo de vida em projetos existentes](#)
- [Usar o gerenciamento do ciclo de vida em vários esquemas em um projeto](#)
- [Trabalho com conflitos em solicitações pull do ciclo de vida](#)

- [Cancelamento das alterações no gerenciamento do ciclo de vida](#)
- [Como ignorar o gerenciamento do ciclo de vida de um esquema em um projeto](#)

Usar o gerenciamento do ciclo de vida em projetos existentes

Você pode usar o gerenciamento do ciclo de vida para projetos criados a partir de esquemas ou em projetos existentes não associados a nenhum esquema. Por exemplo, você pode adicionar um blueprint de práticas de segurança padrão em um aplicativo five-year-old Java que nunca foi criado a partir de um blueprint. O esquema gera um fluxo de trabalho de verificação de segurança e outros códigos relacionados. Essa parte da base de código na aplicação Java agora será atualizada automaticamente com as práticas recomendadas da sua equipe sempre que forem feitas alterações no esquema.

Usar o gerenciamento do ciclo de vida em vários esquemas em um projeto

Como os esquemas representam componentes arquitetônicos, vários esquemas geralmente podem ser usados juntos no mesmo projeto. Por exemplo, um projeto pode ser composto por um esquema central de API da Web criado por um engenheiro de plataforma da empresa, junto com um esquema de verificação de lançamento criado pela equipe de segurança da aplicação. Cada um desses esquemas pode ser atualizado de forma independente e lembrará as resoluções de mesclagem aplicadas a eles no passado.

Note

Como componentes arquitetônicos arbitrários, nem todos os esquemas fazem sentido juntos ou funcionarão logicamente juntos, mesmo que ainda tentem se fundir.

Trabalho com conflitos em solicitações pull do ciclo de vida

Ocasionalmente, solicitações pull do ciclo de vida podem gerar conflitos de mesclagem. Eles podem ser resolvidos manualmente. As resoluções são lembradas nas atualizações subsequentes do esquema.

Cancelamento das alterações no gerenciamento do ciclo de vida

Os usuários podem remover um esquema de um projeto para desassociar todas as referências ao esquema e optar por não receber atualizações do ciclo de vida. Por motivos de segurança, isso

não remove nem afeta nenhum código ou recursos do projeto, incluindo o que foi adicionado do esquema. Para obter mais informações, consulte [Desassociar um esquema de um projeto para interromper as atualizações](#).

Como ignorar o gerenciamento do ciclo de vida de um esquema em um projeto

Se você quiser substituir as atualizações de um blueprint em arquivos específicos em seu projeto, você pode incluir um arquivo de propriedade em seu repositório. GitLabA especificação [do Code Owners](#) é a diretriz recomendada. O esquema sempre respeita o arquivo do proprietário do código acima de tudo e pode gerar uma amostra como a seguinte:

```
new BlueprintOwnershipFile(sourceRepo, {
  resynthesis: {
    strategies: [
      {
        identifier: 'dont-override-sample-code',
        description: 'This strategy is applied accross all sample code. The
blueprint will create sample code, but skip attempting to update it.',
        strategy: MergeStrategies.neverUpdate,
        globs: [
          '**/src/**',
          '**/css/**',
        ],
      },
    ],
  },
});
```

Isso gera um `.ownership-file` com o seguinte conteúdo:

```
[dont-override-sample-code] @amazon-codecatalyst/blueprints.import-from-git
# This strategy is applied accross all sample code. The blueprint will create sample
code, but skip attempting to update it.
# Internal merge strategy: neverUpdate
**/src/**
**/css/**
```

Criação de um projeto abrangente com CodeCatalyst plantas

Ao criar um projeto usando um blueprint, CodeCatalyst cria um projeto completo com um repositório de origem, exemplo de código-fonte, CI/CD fluxos de trabalho, relatórios de criação e teste e ferramentas integradas de controle de problemas. Um esquema de projeto usa código para provisionar infraestrutura de nuvem, recursos e exemplos de artefatos de origem para diferentes tipos de aplicação e estruturas.

Para obter mais informações, consulte [Como criar um projeto](#). Você deve ser administrador do espaço para criar um projeto.

Tópicos

- [Esquemas disponíveis](#)
- [Descoberta de informações sobre o esquema do projeto](#)

Esquemas disponíveis

Nome do esquema	Descrição do esquema
API Web do ASP.NET Core	Esse esquema cria uma aplicação de API Web do .NET 6 ASP.NET Core. O blueprint usa a ferramenta de AWS implantação para .NET e fornece uma opção para configurar o Amazon Elastic Container Service ou AWS Elastic Beanstalk como um destino de implantação. AWS App Runner
ETL do AWS Glue	Esse esquema cria um exemplo de implementação de referência de carga de transformação de extração (ETL) usando AWS CDK, Glue AWS, AWS Lambda e Amazon Athena para converter valores separados por vírgula (,) em Apache Parquet. CSVs
DevOps pipeline de implantação	Esse esquema cria um pipeline de implantação usando a Arquitetura de Referência do AWS Pipeline de Implantação, que implanta

Nome do esquema	Descrição do esquema
	um aplicativo de referência AWS em vários estágios.
API Java com AWS Fargate	Esse esquema cria um projeto de serviço Web em contêineres. O projeto usa a CLI da AWS Copilot para criar e implantar um serviço Web Java Spring Boot em contêineres apoiado pelo Amazon DynamoDB no Amazon ECS. O projeto implanta um aplicativo em contêiner em um cluster do Amazon ECS em computação sem servidor. AWS Fargate A aplicação armazena os dados em uma tabela do DynamoDB. Depois que o fluxo de trabalho for executado, o serviço Web de amostra estará disponível publicamente por meio do Application Load Balancer.
Aplicação web moderna de três níveis	Esse esquema gera código em Python para a camada da aplicação e a estrutura frontend do Vue para criar e implantar uma aplicação web moderna de três níveis bem arquitetada.
Aplicação .NET com tecnologia sem servidor	Esse blueprint cria AWS Lambda funções usando as ferramentas Lambda da CLI do.NET. O blueprint fornece opções para as AWS Lambda funções, incluindo a opção de C# ou F#.

Nome do esquema	Descrição do esquema
API Node.js com AWS Fargate	Esse esquema cria um projeto de serviço Web em contêineres. O projeto usa a CLI do AWS Copilot para criar e implantar um serviço Web Express/Node.js em contêineres no Amazon Elastic Container Service. O projeto implanta um aplicativo em contêiner em um cluster do Amazon ECS em computação sem servidor. AWS Fargate Depois que o fluxo de trabalho for executado, o serviço Web de amostra estará disponível publicamente por meio do Application Load Balancer.
Modelo de aplicação com tecnologia sem servidor (SAM)	Esse esquema cria um projeto que usa um modelo de aplicação com tecnologia sem servidor (SAM) para criar e implantar uma API. Você pode escolher SDK for Java TypeScript ou SDK for Python como linguagem de programação.
Microserviço sem servidor RESTful	Esse blueprint cria uma API REST que usa AWS Lambda e Amazon API Gateway com uma referência de serviço de tarefas pendentes . Você pode escolher SDK for Java TypeScript ou SDK for Python como linguagem de programação.
Aplicação de página única	Esse esquema cria uma aplicação de página única (SPA) que usa estruturas React, Vue e Angular. Para hospedagem, escolha entre AWS Amplify Hosting ou Amazon CloudFront Amazon S3.

Nome do esquema	Descrição do esquema
Site estático	Esse esquema cria um site estático usando os geradores de sites estáticos Hugo ou Jekyll . Os geradores de sites estáticos usam arquivos de entrada de texto (como Markdown) para gerar páginas da Web estáticas. Eles são ideais para conteúdo informativo que raramente muda, como páginas de produtos, documentação e blogs. O blueprint usa o AWS CDK para implantar páginas da web estáticas em um AWS Amplify ou no Amazon CloudFront S3 +.
Aplicação web para tarefas	Esse esquema cria uma aplicação web para tarefas sem servidor com componentes de frontend e de backend. Você pode escolher SDK for Java TypeScript ou SDK for Python como linguagem de programação.
Inscriver-se em um esquema externo	Esse esquema cria um fluxo de trabalho para cada pacote importado. Esses fluxos de trabalho são executados uma vez por dia para verificar se há novas versões dos pacotes no NPM. Se existir uma nova versão, o fluxo de trabalho tentará adicioná-la ao seu CodeCatalyst espaço como um blueprint personalizado. A ação falhará se um pacote não puder ser encontrado ou não for um esquema. O pacote de destino deve estar no NPM e o pacote deve ser um esquema. O espaço deve ser inscrito em uma camada que ofereça suporte a esquemas personalizados.

Nome do esquema	Descrição do esquema
Chatbot de IA generativa do Bedrock	Esse esquema cria um chatbot de IA generativa com o Amazon Bedrock e o Claude da Anthropic . Com esse esquema, você pode criar e implantar seu próprio playground de LLM seguro e protegido por login, que pode ser personalizado de acordo com seus dados. Para ter mais informações, consulte a Documentação do Chatbot de IA generativa do Bedrock .

Descoberta de informações sobre o esquema do projeto

Vários esquemas de projetos estão disponíveis em CodeCatalyst. Para cada esquema, há um resumo e um arquivo README que o acompanham. O resumo descreve os recursos que são instalados pelo esquema, enquanto o arquivo README explica o esquema em detalhes e fornece instruções sobre como usá-lo.

Padronizando projetos com plantas personalizadas em CodeCatalyst

Você pode padronizar o desenvolvimento e as melhores práticas para os projetos do seu CodeCatalyst espaço com plantas personalizadas. Planos personalizados podem ser usados para definir vários aspectos de um CodeCatalyst projeto, como definições de fluxo de trabalho e código do aplicativo. Depois que um esquema personalizado é usado para criar um projeto ou aplicado a projetos existentes, todas as alterações no esquema ficam disponíveis para esses projetos como atualizações de solicitação pull. Como autor do esquema, você pode ver detalhes sobre quais projetos estão usando seus esquemas em todo o espaço, para que você possa ver como os padrões estão sendo aplicados em todos os projetos. O gerenciamento do ciclo de vida de um esquema permite que você gerencie centralmente o ciclo de vida de desenvolvimento de software de cada projeto, permitindo que os projetos em seu espaço continuem seguindo as práticas recomendadas com as alterações ou correções mais recentes. Para obter mais informações, consulte [Trabalho com o gerenciamento do ciclo de vida como autor do esquema](#).

Os esquemas personalizados permitem atualizar as versões do esquema em relação ao projeto anterior por meio de ressíntese. A ressíntese é o processo de executar novamente a síntese do

esquema com versões atualizadas ou a capacidade de incorporar correções e alterações em projetos existentes. Para obter mais informações, consulte [Conceitos de esquemas personalizados](#).

Talvez você já tenha a padronização e as práticas recomendadas implementadas. Em vez de criar e desenvolver um esquema personalizado do zero, você pode optar por converter um repositório de origem existente com código-fonte em um esquema personalizado. Para obter mais informações, consulte [Converter repositórios de origem em esquemas personalizados](#).

[Para ver o SDK de blueprints e exemplos de blueprints, consulte o repositório de código aberto. GitHub](#)

Tópicos

- [Conceitos de esquemas personalizados](#)
- [Conceitos básicos de esquemas personalizados](#)
- [Tutorial: Criar e atualizar uma aplicação React](#)
- [Converter repositórios de origem em esquemas personalizados](#)
- [Trabalho com o gerenciamento do ciclo de vida como autor do esquema](#)
- [Desenvolvimento de um esquema personalizado para satisfazer os requisitos do projeto](#)
- [Publicação de um esquema personalizado em um espaço](#)
- [Definição de permissões de publicação para um esquema personalizado](#)
- [Adicionar um esquema personalizado a um catálogo de esquemas de espaço](#)
- [Alterar as versões do catálogo para um esquema personalizado](#)
- [Visualização de detalhes, versões e projetos de um esquema personalizado](#)
- [Remoção de um esquema personalizado de um catálogo de esquemas de espaço](#)
- [Exclusão de um esquema ou uma versão personalizada publicada](#)
- [Como lidar com dependências, incompatibilidades e ferramentas](#)
- [Contribuir](#)

Conceitos de esquemas personalizados

Confira a seguir alguns conceitos e termos que você precisa conhecer ao trabalhar com esquemas personalizados no CodeCatalyst.

Tópicos

- [Projeto do esquema](#)

- [Esquemas de espaço](#)
- [Catálogo de esquemas de espaço](#)
- [Síntese](#)
- [Ressíntese](#)
- [Opções parciais](#)
- [Projen](#)

Projeto do esquema

Um projeto de esquema permite desenvolver e publicar esquemas em seu espaço. Um repositório de origem é criado durante o processo de criação do projeto, e o nome do repositório é aquele que você escolheu ao inserir os detalhes dos Recursos do projeto. Durante o processo de criação do esquema, se você optar por gerar uma versão do fluxo de trabalho, um fluxo de trabalho de publicação será criado no esquema com o Criador de esquema. O fluxo de trabalho publica automaticamente sua versão mais recente.

Esquemas de espaço

Você pode visualizar e gerenciar todos os esquemas na tabela Esquemas de espaço ao navegar até a seção Esquemas do seu espaço. Depois que são publicados em seu espaço, os esquemas são disponibilizados como um esquema de espaço para serem adicionados e removidos do catálogo de esquemas do seu espaço. Você também pode gerenciar as permissões de publicação e excluir esquemas na seção Esquemas do seu espaço. Para obter mais informações, consulte [Visualização de detalhes, versões e projetos de um esquema personalizado](#).

Catálogo de esquemas de espaço

Você pode ver todos os esquemas personalizados adicionados no catálogo de esquemas de um espaço. É aqui que um membro do espaço pode escolher seu esquema personalizado para criar um projeto. Esse catálogo é diferente do catálogo do CodeCatalyst, que já tem esquemas disponíveis para todos os membros do espaço. Para obter mais informações, consulte [Criação de um projeto abrangente com CodeCatalyst plantas](#).

Síntese

A síntese é o processo de gerar um pacote de projeto do CodeCatalyst que representa o código-fonte, a configuração e os recursos em um projeto. O pacote é usado pelas operações da API de implantação do CodeCatalyst para implantar em um projeto. O processo pode ser executado

localmente durante o desenvolvimento do esquema personalizado para emular a criação do projeto sem precisar criar um projeto no CodeCatalyst. Os comandos a seguir podem ser usados para realizar a síntese:

```
yarn blueprint:synth          # fast mode
yarn blueprint:synth --cache  # wizard emulation mode
```

O esquema é iniciado chamando a classe `blueprint.ts` principal com essa opção mesclada em `defaults.json`. Um novo pacote de projeto é gerado na pasta `synth/synth.[options-name]/proposed-bundle/`. A saída inclui o pacote de projeto que um esquema personalizado gera, dadas as opções definidas, incluindo as [opções parciais](#) que você pode ter configurado.

Ressíntese

A ressíntese é o processo de nova geração de um esquema com diferentes opções de esquema ou versões de esquema de projetos existentes. Como autor do esquema, você pode definir estratégias de mesclagem personalizadas no código do esquema personalizado. Você também pode definir limites de propriedade em um `.ownership-file` para especificar em quais partes da base de código um esquema pode ser atualizado. Embora o esquema personalizado possa propor atualizações para o `.ownership-file`, os desenvolvedores de projeto que usam o esquema personalizado podem determinar os limites de propriedade dos projetos. Você pode executar a ressíntese localmente, testar e atualizar antes de publicar seu esquema personalizado. Use os comandos a seguir para executar a ressíntese:

```
yarn blueprint:resynth       # fast mode
yarn blueprint:resynth --cache # wizard emulation mode
```

O esquema é iniciado chamando a classe `blueprint.ts` principal com essa opção mesclada em `defaults.json`. Um novo pacote de projeto é gerado na pasta `synth/resynth.[options-name]/`. A saída inclui o pacote de projeto que um esquema personalizado gera, dadas as opções definidas, incluindo as [opções parciais](#) que você pode ter configurado.

Os seguintes conteúdos são criados após os processos de síntese e ressíntese:

- `proposed-bundle` - A saída da síntese quando ela é executada com novas opções para a versão do esquema de destino.
- `existing-bundle` - Uma simulação do projeto existente. Se não houver nada nessa pasta, ela será gerada com a mesma saída que `proposed-bundle`.

- **ancestor-bundle** - Uma simulação do que seu esquema geraria quando executado com uma versão anterior, opções anteriores ou uma combinação de ambas. Se não houver nada nessa pasta, ela será gerada com a mesma saída que **proposed-bundle**.
- **resolved-bundle** - O pacote sempre é gerado novamente e o padrão é uma mesclagem de três vias entre **proposed-bundle**, **existing-bundle** e **ancestor-bundle**. Esse pacote fornece uma emulação do que uma ressíntese produziria localmente.

Para saber mais sobre os pacotes de saída do esquema, consulte [Gerar arquivos com ressíntese](#).

Opções parciais

Você pode adicionar variações de opções em `src/wizard-configuration/` que não precisem enumerar toda a interface de `Options`, e as opções são mescladas no arquivo `defaults.json`. Isso permite que você personalize os casos de teste em opções específicas.

Exemplo:

Interface de `Options`:

```
{
  language: "Python" | "Java" | "Typescript",
  repositoryName: string
  ...
}
```

`defaults.json`Arquivo :

```
{
  language: "Python",
  repositoryName: "Myrepo"
  ...
}
```

Testes de configuração adicionais:

- `#wizard-config-typescript-test.json`

```
{
  language: "Typescript",
}
```

- ```
#wizard-config-java-test.json
{
 language: "Java",
}
```

## Projen

O Projen é uma ferramenta de código aberto que esquemas personalizados usam para se manterem atualizados e consistentes. Os esquemas vêm como pacotes do Projen porque essa estrutura permite criar, agrupar e publicar projetos, e você pode usar a interface para gerenciar as configurações de um projeto.

Você pode usar o Projen para atualizar os esquemas em grande escala, mesmo depois de criados. A ferramenta Projen é a tecnologia subjacente por trás da síntese do esquema que gera um pacote de projeto. O Projen possui a configuração de um projeto e isso não deve afetar você como autor do esquema. Você pode executar `yarn projen` para gerar novamente a configuração do projeto depois de adicionar dependências ou pode alterar as opções no arquivo `projenrc.ts`. O Projen também é a ferramenta de geração subjacente para esquemas personalizados para sintetizar um projeto. Para ter mais informações, consulte a [página Projen no GitHub](#). Para saber mais sobre como trabalhar com o Projen, consulte a [documentação do Projen](#).

## Conceitos básicos de esquemas personalizados

Durante o processo de criação de um esquema, você pode configurar o esquema e gerar uma prévia dos recursos do projeto. Cada esquema personalizado é gerenciado por um projeto do CodeCatalyst, que contém um fluxo de trabalho por padrão para publicação no catálogo de esquemas do espaço.

Ao configurar os detalhes do esquema personalizado, você também pode optar por armazenar o código-fonte do esquema em um repositório de terceiros, onde ainda é possível gerenciar o esquema personalizado e utilizar os recursos de gerenciamento do ciclo de vida para manter os projetos do seu espaço sincronizados quando o esquema personalizado é modificado. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Trabalho com o gerenciamento do ciclo de vida como autor do esquema](#).

Se você já tem um repositório de origem com padronização e práticas recomendadas em vigor, pode optar por converter esse repositório de origem em um esquema personalizado. Para obter mais informações, consulte [Converter repositórios de origem em esquemas personalizados](#).

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um esquema personalizado no CodeCatalyst](#)
- [Etapa 2: desenvolver um esquema personalizado com componentes](#)
- [Etapa 3: visualizar um esquema personalizado](#)
- [\(Opcional\) Etapa 4: publicar uma versão prévia do esquema personalizado](#)

## Pré-requisitos

Antes de criar um esquema personalizado, pense nos seguintes requisitos:

- Seu espaço no CodeCatalyst deve ser do nível Enterprise. Para ter mais informações, consulte [Managing billing](#) no Guia do administrador do Amazon CodeCatalyst.
- Você precisa ter o perfil Administrador do espaço ou Usuário avançado para criar esquemas personalizados. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).

## Etapa 1: criar um esquema personalizado no CodeCatalyst

Quando você cria um esquema personalizado a partir das configurações do seu espaço, um repositório é criado. O repositório inclui todos os recursos necessários para desenvolver seu esquema antes de publicá-lo no catálogo de esquemas do espaço.

Para criar um esquema personalizado

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço onde você deseja criar um esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Escolha Criar esquema.
5. Em Nomear seu esquema, insira o nome que você deseja atribuir ao seu projeto e os nomes dos recursos associados. O nome deve ser exclusivo em seu espaço.
6. (Opcional) Por padrão, o código-fonte criado pelo esquema é armazenado em um repositório do CodeCatalyst. Também é possível armazenar o código-fonte do esquema em um repositório

de terceiros. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você deseja usar:

- Repositórios do GitHub: conecte uma conta do GitHub.

Escolha o menu suspenso Avançado, selecione GitHub como provedor do repositório e, depois, selecione a conta do GitHub na qual você deseja armazenar o código-fonte criado pelo esquema.

 Note

Se você estiver usando uma conexão com uma conta do GitHub, deverá criar uma conexão pessoal para estabelecer o mapeamento entre sua identidade do CodeCatalyst e sua identidade do GitHub. Para obter mais informações, consulte [Conexões pessoais](#) e [Acessando GitHub recursos com conexões pessoais](#).

- Repositórios do Bitbucket: conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, selecione Bitbucket como provedor do repositório e, depois, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo esquema.

- Repositórios do GitLab: conecte um usuário do GitLab.

Escolha o menu suspenso Avançado, selecione GitLab como provedor do repositório e, depois, escolha o usuário do GitLab no qual você deseja armazenar o código-fonte criado pelo esquema.

7. Em Detalhes do esquema, faça o seguinte:

- a. No campo de entrada de texto Nome de exibição do esquema, insira um nome que aparecerá no catálogo de esquemas do espaço.
- b. No campo de entrada de texto Descrição, insira uma descrição para seu esquema personalizado.
- c. No campo de entrada de texto Nome do autor, insira um nome de autor para seu esquema personalizado.
- d. (Opcional) Escolha as Configurações avançadas.

- i. Selecione + Adicionar para adicionar tags que são adicionadas ao arquivo `package.json`.
  - ii. Escolha o menu suspenso Licença e, depois, escolha uma licença para seu esquema personalizado.
  - iii. No campo de entrada de texto Nome do pacote do esquema, insira um nome para identificar seu pacote do esquema.
  - iv. Por padrão, um fluxo de trabalho de lançamento é gerado usando um esquema de publicação em seu projeto chamado Blueprint Builder. O fluxo de trabalho publica a versão mais recente do esquema em seu espaço quando você envia alterações, já que as permissões de publicação são habilitadas pelo fluxo de trabalho de lançamento. Para desativar a geração do fluxo de trabalho, desmarque a caixa de seleção Fluxo de trabalho de lançamento.
8. (Opcional) Um projeto de esquema vem com código predefinido para apoiar a publicação do esquema no catálogo do espaço. Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, selecione Visualizar código ou Visualizar fluxo de trabalho em Gerar visualização do esquema.
9. Escolha Criar esquema.

Se você não desativou a geração do fluxo de trabalho para seu esquema personalizado, o fluxo de trabalho começa a ser executado automaticamente quando o esquema é criado. Quando a execução do fluxo de trabalho estiver concluída, seu esquema personalizado estará disponível para ser adicionado ao catálogo de esquemas do espaço por padrão. Você pode desativar as permissões de publicação se não quiser que a versão mais recente do esquema seja publicada automaticamente no espaço. Para obter mais informações, consulte [Definição de permissões de publicação para um esquema personalizado](#) e [Executar um fluxo de trabalho](#).

Como o fluxo de trabalho de publicação chamado `blueprint-release` é criado usando um esquema, o esquema pode ser encontrado como um esquema aplicado em seu projeto. Para obter mais informações, consulte [Adicionar um esquema a um projeto para integrar recursos](#) e [Desassociar um esquema de um projeto para interromper as atualizações](#).

## Etapa 2: desenvolver um esquema personalizado com componentes

Um assistente de esquema é gerado quando você cria um esquema personalizado e pode ser modificado com componentes ao desenvolver o esquema personalizado. Você pode atualizar os arquivos `src/blueprints.js` e `src/defaults.json` para modificar o assistente.

 Important

Se você quiser usar pacotes de esquema de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos esquemas personalizados que adiciona ao seu espaço e pelo código que eles geram.

Crie um Ambiente de Desenvolvimento em seu projeto do CodeCatalyst com um ambiente de desenvolvimento integrado (IDE) compatível antes de configurar o código do esquema. É necessário um Ambiente de Desenvolvimento para trabalhar com as ferramentas e os pacotes necessários.

Como criar um ambiente de desenvolvimento

1. No painel de navegação, execute uma das seguintes opções:
  - a. Selecione Visão geral e navegue até a seção Meus ambientes de desenvolvimento.
  - b. Selecione Código e, depois, escolha Ambientes de Desenvolvimento.
  - c. Selecione Código, selecione Repositórios de origem e escolha o repositório que você criou ao criar o esquema.
2. Escolha Criar ambiente de desenvolvimento.
3. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para ter mais informações.
4. Selecione Trabalhar na ramificação existente e, no menu suspenso Ramificação existente, escolha a ramificação do recurso que você criou.
5. (Opcional) No campo de entrada de texto Alias - opcional, insira um alias para identificar o Ambiente de Desenvolvimento.
6. Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento.

Para obter mais informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

Para desenvolver seu esquema personalizado

1. Em um terminal em funcionamento, use o seguinte comando yarn para instalar dependências:



```
yarn
```

As ferramentas e os pacotes necessários são disponibilizados por meio do Ambiente de Desenvolvimento do CodeCatalyst, incluindo o Yarn. Se você estiver trabalhando em um esquema personalizado sem um Ambiente de Desenvolvimento, instale primeiro o Yarn em seu sistema. Para ter mais informações, consulte a [documentação de instalação do Yarn](#).

- Desenvolva seu esquema personalizado para que ele seja configurado de acordo com suas preferências. Você pode modificar o assistente do esquema adicionando componentes. Para ter mais informações, consulte [Desenvolvimento de um esquema personalizado para satisfazer os requisitos do projeto](#), [Modificar os recursos do esquema com um assistente de frontend](#) e [Publicação de um esquema personalizado em um espaço](#).

### Etapa 3: visualizar um esquema personalizado

Depois de configurar e desenvolver seu esquema personalizado, você pode visualizar e publicar a versão prévia do esquema no espaço. Uma versão prévia permite verificar se o esquema é o que você deseja antes de ser usado para criar novos projetos ou aplicado a projetos existentes.

Para visualizar um esquema personalizado

- Em um terminal em funcionamento, use o seguinte comando yarn:

```
yarn blueprint:preview
```

- Navegue até o link `See this blueprint at:` fornecido para visualizar seu esquema personalizado.
- Confira se a interface do usuário, inclusive o texto, aparece conforme o esperado com base na configuração. Se quiser alterar o esquema personalizado, você pode editar o arquivo `blueprint.ts`, resintetizar o esquema e, depois, publicar uma versão prévia novamente. Para obter mais informações, consulte [Ressíntese](#).

### (Opcional) Etapa 4: publicar uma versão prévia do esquema personalizado

Você pode publicar uma versão prévia do esquema personalizado no espaço se quiser adicioná-la ao catálogo de esquemas do espaço. Isso permite que você visualize o esquema como usuário antes de adicionar a versão sem pré-visualização ao catálogo. A versão prévia permite que você

publique sem usar uma versão real. Por exemplo, se você trabalha em uma versão `0.0.1`, pode publicar e adicionar uma versão prévia, para que novas atualizações para uma segunda versão possam ser publicadas e adicionadas como `0.0.2`.

Para publicar uma versão prévia de um esquema personalizado

Navegue até o link `Enable version [version number]` at: fornecido para habilitar seu esquema personalizado. Esse link é fornecido ao executar o comando `yarn` em [Etapa 3: visualizar um esquema personalizado](#).

Depois de criar, desenvolver, visualizar e publicar seu esquema personalizado, você pode publicar e adicionar a versão final do esquema ao catálogo do espaço. Para obter mais informações, consulte [Publicação de um esquema personalizado em um espaço](#) e [Adicionar um esquema personalizado a um catálogo de esquemas de espaço](#).

## Tutorial: Criar e atualizar uma aplicação React

Como autor de esquemas, você pode desenvolver e adicionar esquemas personalizados ao catálogo de esquemas do espaço. Esses esquemas podem ser usados pelos membros do espaço para criar projetos ou adicioná-los a projetos existentes. Você pode continuar fazendo alterações nos esquemas, que são disponibilizadas como atualizações por meio de solicitações pull.

Este tutorial fornece um passo a passo do ponto de vista do autor e do usuário do esquema. O tutorial mostra como criar um esquema de aplicação web de página única do React. O esquema é usado para criar um projeto. Quando o esquema é atualizado com alterações, o projeto criado a partir do esquema incorpora essas alterações por meio de uma solicitação pull.

### Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um esquema personalizado](#)
- [Etapa 2: visualizar o fluxo de trabalho de lançamento](#)
- [Etapa 3: adicionar esquema ao catálogo](#)
- [Etapa 4: criar projeto com esquema](#)
- [Etapa 5: atualizar o esquema](#)
- [Etapa 6: atualizar a versão publicada do catálogo do esquema para a nova versão](#)
- [Etapa 7: atualizar projeto com a nova versão do esquema](#)
- [Etapa 8: visualizar as alterações no projeto](#)

## Pré-requisitos

Para criar e atualizar um esquema personalizado, você deve ter concluído as tarefas descritas em [Configuração e login no CodeCatalyst](#) da seguinte forma:

- Tenha um ID do builder AWS para fazer login no CodeCatalyst.
- Você deve pertencer a um espaço e ter o perfil Administrador do espaço ou Usuário avançado atribuído a você nesse espaço. Para ter mais informações, consulte [Criar um espaço](#), [Concessão de permissões de espaço aos usuários](#) e [Perfil de administrador do espaço](#).

## Etapa 1: criar um esquema personalizado

Quando você cria um esquema personalizado, é criado um projeto do CodeCatalyst que contém o código-fonte e as ferramentas e os recursos de desenvolvimento do esquema. Você desenvolverá, testará e publicará o esquema no projeto.

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, acesse o espaço onde você deseja criar um esquema.
3. Escolha Configurações para navegar até as configurações de espaço.
4. Na guia Configurações de espaço, escolha Esquemas e Criar esquema.
5. Atualize os campos no assistente de criação de esquema com os seguintes valores:
  - Em Nome do esquema, insira `react-app-blueprint`.
  - Em Nome de exibição do esquema, insira `react-app-blueprint`.
6. Se desejar, escolha Exibir código para visualizar o código-fonte do esquema. Da mesma forma, escolha Exibir fluxo de trabalho para visualizar o fluxo de trabalho que será criado no projeto que cria e publica o esquema.
7. Escolha Criar esquema.
8. Depois que o esquema é criado, você é levado ao projeto do esquema. Esse projeto contém o código-fonte do esquema, junto com as ferramentas e os recursos necessários para desenvolver, testar e publicar o esquema. Um fluxo de trabalho de lançamento foi gerado e publicou automaticamente seu esquema no espaço.
9. Agora que o esquema e o projeto do esquema foram criados, a próxima etapa é configurá-los atualizando o código-fonte. Você pode usar Ambientes de Desenvolvimento para abrir e editar o repositório de origem diretamente no navegador.

No painel de navegação, escolha Código e Ambientes de Desenvolvimento.

10. Escolha Criar ambiente de desenvolvimento e AWS Cloud9 (no navegador).
11. Mantenha o restante das configurações padrão e escolha Criar.
12. No terminal do AWS Cloud9, acesse o diretório do projeto do esquema e execute o seguinte comando:

```
cd react-app-blueprint
```

13. Uma pasta `static-assets` é criada e preenchida automaticamente quando um esquema é criado. Neste tutorial, você excluirá a pasta padrão e gerará uma nova para um esquema da aplicação react.

Exclua a pasta `static-assets` executando o seguinte comando:

```
rm -r static-assets
```

O AWS Cloud9 é desenvolvido em uma plataforma baseada em Linux. Se estiver usando um sistema operacional Windows, você poderá usar o seguinte comando:

```
rmdir /s /q static-assets
```

14. Agora que a pasta padrão foi excluída, crie uma pasta `static-assets` para um esquema da aplicação react executando o seguinte comando:

```
npx create-react-app static-assets
```

Se solicitado, insira `y` para continuar.

Uma nova aplicação react foi criada na pasta `static-assets` com os pacotes necessários. As alterações precisam ser enviadas ao repositório de origem remoto do CodeCatalyst.

15. Verifique se você tem as alterações mais recentes e, depois, confirme e envie as alterações para o repositório de origem do CodeCatalyst do esquema executando os seguintes comandos:

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Quando uma alteração é enviada ao repositório de origem do esquema, o fluxo de trabalho de lançamento é iniciado automaticamente. Esse fluxo de trabalho aumenta a versão do esquema, cria o esquema e o publica no espaço. Na próxima etapa, você acessará a execução do fluxo de trabalho da versão para ver como está.

## Etapa 2: visualizar o fluxo de trabalho de lançamento

1. No console do CodeCatalyst, no painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha o fluxo de trabalho blueprint-release.
3. O fluxo de trabalho tem ações para criar e publicar o esquema.
4. Em Última execução, escolha o link de execução do fluxo de trabalho para visualizar a execução a partir da alteração de código que você fez.
5. Depois que a execução for concluída, a nova versão do esquema será publicada. As versões publicadas do esquema podem ser vistas nas Configurações do espaço, mas não podem ser usadas em projetos até serem adicionadas ao catálogo de esquemas do espaço. Na próxima etapa, você adicionará o esquema ao catálogo.

## Etapa 3: adicionar esquema ao catálogo

Adicionar um esquema ao catálogo do espaço o torna disponível para uso em todos os projetos em um espaço. Os membros do espaço podem usar o esquema para criar novos projetos ou adicioná-lo aos projetos existentes.

1. No console do CodeCatalyst, navegue de volta ao espaço.
2. Escolha Configurações e Esquemas.
3. Escolha react-app-blueprint e Adicionar ao catálogo.
4. Escolha Salvar.

## Etapa 4: criar projeto com esquema

Agora que o esquema foi adicionado ao catálogo, ele pode ser usado em projetos. Nesta etapa, você criará um projeto com o esquema criado. Em uma etapa posterior, você atualizará esse projeto atualizando e publicando uma nova versão do esquema.

1. Escolha a guia Projetos e Criar projeto.
2. Escolha Esquemas do espaço e react-app-blueprint.

### Note

Depois que o esquema for escolhido, você poderá ver o conteúdo do arquivo `README.md` do esquema.

3. Escolha Próximo.

4.

### Note

O conteúdo desse assistente de criação de projeto pode ser configurado no esquema.

Insira o nome do projeto como usuário do esquema. Para este tutorial, insira `react-app-project`. Para obter mais informações, consulte [Desenvolvimento de um esquema personalizado para satisfazer os requisitos do projeto](#).

Depois, você atualizará o esquema e adicionará a nova versão ao catálogo, que será usada para atualizar esse projeto.

## Etapa 5: atualizar o esquema

Depois que um esquema é usado para criar um projeto ou aplicado a projetos existentes, você pode continuar fazendo atualizações como autor do esquema. Nesta etapa, você fará alterações no esquema e publicará automaticamente uma nova versão no espaço. A nova versão pode ser adicionada como a versão do catálogo.

1. Navegue até o projeto `react-app-blueprint` criado em [Tutorial: Criar e atualizar uma aplicação React](#).
2. Abra o Ambiente de Desenvolvimento criado em [Tutorial: Criar e atualizar uma aplicação React](#).

- a. No painel de navegação, escolha Código e Ambientes de desenvolvimento.
  - b. Na tabela, localize o Ambiente de Desenvolvimento e escolha Abrir n AWS Cloud9 (no navegador).
3. Quando o fluxo de trabalho de lançamento do esquema foi executado, ele aumentou a versão do esquema atualizando o arquivo `package.json`. Faça essa alteração executando o seguinte comando no terminal do AWS Cloud9:

```
git pull
```

4. Navegue até a pasta `static-assets` executando o seguinte comando:

```
cd /projects/react-app-blueprint/static-assets
```

5. Crie um arquivo `hello-world.txt` na pasta `static-assets` executando o seguinte comando:

```
touch hello-world.txt
```

O AWS Cloud9 é desenvolvido em uma plataforma baseada em Linux. Se estiver usando um sistema operacional Windows, você poderá usar o seguinte comando:

```
echo > hello-world.txt
```

6. No painel de navegação à esquerda, clique duas vezes no arquivo `hello-world.txt` para abri-lo no editor e adicione o seguinte conteúdo:

```
Hello, world!
```

Salve o arquivo.

7. Verifique se você tem as alterações mais recentes e, depois, confirme e envie as alterações para o repositório de origem do CodeCatalyst do esquema executando os seguintes comandos:

```
git pull
```

```
git add .
```

```
git commit -m "prettier setup"
```

```
git push
```

O envio das alterações iniciou o fluxo de trabalho de lançamento, que publicará automaticamente a nova versão do esquema no espaço.

## Etapa 6: atualizar a versão publicada do catálogo do esquema para a nova versão

Depois que um esquema é usado para criar um projeto ou aplicado a projetos existentes, você ainda pode atualizar o esquema como autor. Nesta etapa, você fará alterações no esquema e alterará a versão do catálogo do esquema.

1. No console do CodeCatalyst, navegue de volta ao espaço.
2. Escolha Configurações e Esquemas.
3. Escolha react-app-blueprint e Gerenciar versão do catálogo.
4. Escolha a nova versão e Salvar.

## Etapa 7: atualizar projeto com a nova versão do esquema

Uma nova versão já está disponível no catálogo de esquemas do espaço. Como usuário do esquema, você pode atualizar a versão do projeto criado em [Etapa 4: criar projeto com esquema](#). Isso garante que você tenha as alterações e correções mais recentes necessárias para atender às práticas recomendadas.

1. No console do CodeCatalyst, navegue até o projeto react-app-project criado em [Etapa 4: criar projeto com esquema](#).
2. No painel de navegação, escolha Esquemas.
3. Escolha Atualizar esquema na caixa de informações.
4. No painel Alterações de código do lado direito, você pode ver as atualizações `hello-world.txt` e `package.json`.
5. Escolha Aplicar atualização.



Escolher Aplicar atualização cria uma solicitação pull no projeto com as alterações da versão atualizada do esquema. Para fazer as atualizações no projeto, você deve mesclar a solicitação pull. Para obter mais informações, consulte [Revisão de uma solicitação pull](#) e [Mesclar uma solicitação pull](#).

1. Na tabela Esquemas, encontre o esquema. Na coluna Status, escolha Solicitação pull pendente e, depois, escolha o link para a solicitação pull aberta.
2. Revise a solicitação pull e escolha Mesclar.
3. Escolha Mesclagem rápida para manter os valores padrão e, depois, escolha Mesclar.

## Etapa 8: visualizar as alterações no projeto

As alterações no esquema agora estão disponíveis em seu projeto depois de [Etapa 7: atualizar projeto com a nova versão do esquema](#). Como usuário do esquema, você pode visualizar as alterações no repositório de origem.

1. No painel de navegação, escolha Repositórios de origem e escolha o nome do repositório de origem criado quando o projeto foi criado.
2. Em Arquivos, você pode visualizar o arquivo `hello-world.txt` que foi criado em [Etapa 5: atualizar o esquema](#).
3. Escolha o `hello-world.txt` para visualizar o conteúdo do arquivo.

O gerenciamento do ciclo de vida fornece aos autores do esquema a capacidade de gerenciar centralmente o ciclo de vida de desenvolvimento de software de cada projeto que contém um esquema específico. Conforme visto neste tutorial, você pode enviar atualizações para o esquema que podem ser incorporadas por projetos que usaram o esquema para criar um projeto ou aplicá-lo a um projeto existente. Para obter mais informações, consulte [Trabalho com o gerenciamento do ciclo de vida como autor do esquema](#).

## Converter repositórios de origem em esquemas personalizados

Os esquemas personalizados permitem que você incorpore as práticas recomendadas ou novas opções em vários projetos em um espaço do CodeCatalyst. Embora você possa criar e desenvolver um novo esquema personalizado do zero, também é possível converter um repositório de origem do CodeCatalyst existente ou de terceiros com código e práticas recomendadas estabelecidas em um projeto de esquema. Você pode evitar copiar artefatos relevantes desse repositório existente em um

projeto de esquema. Depois de converter um repositório de origem em um esquema personalizado, você pode atualizar, publicar e adicionar o esquema como qualquer outro esquema personalizado.

Depois de converter um repositório de origem em um esquema personalizado, o repositório de origem é reestruturado para ser um projeto de esquema, o conteúdo do repositório é movido para uma pasta `static-assets` dentro do repositório e os ativos relevantes necessários para um esquema são adicionados ao repositório. Quando o esquema personalizado é usado para criar projetos ou adicionado a um projeto existente, as definições de fluxo de trabalho armazenadas no repositório de origem convertido também são adicionadas aos projetos pelo esquema.

#### Note

Recursos como ambientes e segredos não são incluídos na conversão de um repositório de origem em um esquema personalizado. Você precisa copiar ou adicionar manualmente esses recursos depois de converter um repositório de origem em um esquema personalizado.

#### Important

Para converter um repositório de origem em um esquema personalizado, você deve estar conectado com uma conta que tenha o perfil de Administrador do projeto, Administrador do espaço ou Usuário avançado no espaço.

Para converter um repositório de origem em um esquema personalizado da lista de repositórios de origem

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço e escolha o projeto em que você deseja converter um repositório de origem em um esquema personalizado.
3. No painel de navegação, selecione Código, selecione Repositórios de origem e, depois, escolha o botão de opção do repositório de origem que você deseja converter em um esquema personalizado.
4. Selecione Converter em esquema para converter seu repositório de origem em um esquema personalizado.

Você também pode converter repositórios do CodeCatalyst em esquemas personalizados na página do repositório de origem no console do CodeCatalyst.

Para converter um repositório de origem em um esquema personalizado na página do repositório de origem

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço e escolha o projeto em que você deseja converter um repositório de origem em um esquema personalizado.
3. No painel de navegação, selecione Código, selecione Repositórios de origem e, depois, escolha o nome do repositório de origem do CodeCatalyst que você deseja converter em um esquema personalizado.
4. Escolha o menu suspenso Mais e, depois, selecione Converter em esquema para converter o repositório de origem em um esquema personalizado.

Depois o seu repositório de origem é convertido em um esquema personalizado, um fluxo de trabalho de versão é executado automaticamente. Depois que a execução for concluída, o esquema personalizado será publicado na lista de esquemas personalizados do seu espaço. A partir daí, você pode adicionar o esquema personalizado convertido ao catálogo do espaço para criar novos projetos ou ser adicionado a projetos existentes. Para obter mais informações, consulte [Publicação de um esquema personalizado em um espaço](#) e [Adicionar um esquema personalizado a um catálogo de esquemas de espaço](#).

#### Important

Para adicionar um esquema personalizado ao catálogo de esquemas do seu espaço, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

## Trabalho com o gerenciamento do ciclo de vida como autor do esquema

O gerenciamento do ciclo de vida permite que você mantenha um grande número de projetos sincronizados a partir de uma única fonte comum de práticas recomendadas. Isso escala a propagação de correções e a manutenção de qualquer número de projetos em todo o ciclo de vida de desenvolvimento de software. O gerenciamento do ciclo de vida simplifica campanhas internas, correções de segurança, auditorias, atualizações de runtime, alterações nas práticas

recomendadas e outras práticas de manutenção porque esses padrões são definidos em um só lugar e automaticamente atualizados de modo centralizado quando novos padrões são publicados.

Quando uma nova versão do esquema é publicada, todos os projetos que contêm esse esquema são solicitados a atualizar para a versão mais recente. Como autor do esquema, você também pode ver a versão de um determinado esquema que cada projeto contém para fins de conformidade. Quando há conflitos em um repositório de origem existente, o gerenciamento do ciclo de vida cria solicitações pull. Para todos os outros recursos, como o Ambiente de Desenvolvimento, todas as atualizações de gerenciamento do ciclo de vida criam estritamente novos recursos. Os usuários são livres para mesclar ou não essas solicitações pull. Quando as solicitações pull pendentes são mescladas, a versão do esquema, incluindo as opções, usada no projeto é atualizada. Para saber mais sobre como trabalhar com o gerenciamento do ciclo de vida como usuário do esquema, consulte [Usar o gerenciamento do ciclo de vida em projetos existentes](#) e [Usar o gerenciamento do ciclo de vida em vários esquemas em um projeto](#).

## Tópicos

- [Teste do gerenciamento do ciclo de vida para saídas de pacotes e conflitos de mesclagem](#)
- [Uso de estratégias de mesclagem para gerar pacotes e especificar arquivos](#)
- [Acesso a objetos de contexto para saber detalhes do projeto](#)

## Teste do gerenciamento do ciclo de vida para saídas de pacotes e conflitos de mesclagem

Você pode testar localmente o gerenciamento do ciclo de vida do seu esquema e mesclar a resolução de conflitos. Uma série de pacotes no diretório synth/ que representam as várias fases de uma atualização do ciclo de vida é gerada. Para testar o gerenciamento do ciclo de vida, execute o seguinte comando yarn em seu esquema: `yarn blueprint: resynth`. Para saber mais sobre ressíntese e pacotes, consulte e. [Ressíntese Gerar arquivos com ressíntese](#)

## Uso de estratégias de mesclagem para gerar pacotes e especificar arquivos

Você pode usar estratégias de mesclagem para gerar pacotes com ressíntese e especificar arquivos para atualizações de gerenciamento do ciclo de vida de esquemas personalizados. Ao aproveitar as estratégias de ressíntese e mesclagem, você pode gerenciar atualizações e controlar quais arquivos são atualizados durante as implantações. Você também pode escrever suas próprias estratégias para controlar como as alterações são mescladas aos projetos existentes do CodeCatalyst.

## Tópicos

- [Gerar arquivos com ressíntese](#)
- [Usar estratégias de mesclagem](#)
- [Especificação de arquivos para atualizações de gerenciamento do ciclo de vida](#)
- [Escrita de estratégias de mesclagem](#)

## Gerar arquivos com ressíntese

A ressíntese pode mesclar o código-fonte produzido por um esquema com o código-fonte que foi gerado anteriormente pelo mesmo esquema, permitindo que as alterações em um esquema sejam propagadas para projetos existentes. As mesclagens são executadas a partir da função `resynth()` nos pacotes de saída do esquema. A ressíntese primeiro gera três pacotes que representam diferentes aspectos do esquema e do estado do projeto. Ela pode ser executada manualmente no local com o comando `yarn blueprint:resynth`, que criará os pacotes se eles ainda não existirem. Trabalhar manualmente com os pacotes permitirá que você simule e teste o comportamento da ressíntese localmente. Por padrão, os esquemas só executam a ressíntese nos repositórios em `src/*`, já que somente essa parte do pacote normalmente está sob controle de origem. Para obter mais informações, consulte [Ressíntese](#).

- `existing-bundle` - Esse pacote é uma representação do estado existente do projeto. Ele é construído artificialmente pela computação de síntese para fornecer ao esquema um contexto sobre o que está no projeto em que está sendo implantado (se houver). Se algo já existir nesse local ao executar a ressíntese localmente, ele será redefinido e respeitado como uma simulação. Caso contrário, ele será definido como o conteúdo do `ancestor-bundle`.
- `ancestor-bundle` - Esse é o pacote que representa a saída do esquema se tiver sido sintetizado com algumas opções e/ou versões anteriores. Se for a primeira vez que esse esquema está sendo adicionado a um projeto, o ancestral não existe, e ele está definido com o mesmo conteúdo do `existing-bundle`. Localmente, se esse pacote já existir, ele será respeitado como uma simulação.
- `proposed-bundle` - Esse é o pacote que simula o esquema se tiver sido sintetizado com algumas novas opções e/ou versões. Esse é o mesmo pacote que seria produzido pela função `synth()`. Localmente, esse pacote sempre é substituído.

Cada pacote é criado durante uma fase de ressíntese que pode ser acessada a partir da classe de esquema em `this.context.resynthesisPhase`.

- **resolved-bundle** - Esse é o pacote final, que é uma representação do que é empacotado e implantado em um projeto do CodeCatalyst. Você pode ver quais arquivos e diffs são enviados para os mecanismos de implantação. Essa é a saída da função `resynth()` que resolve mesclagens entre os outros três pacotes.

A mesclagem de três vias é aplicada tomando a diferença entre o `ancestor-bundle` e `proposed-bundle` e adicionando-a ao `existing-bundle` para gerar o `resolved-bundle`. Todas as estratégias de mesclagem resolvem arquivos para o `resolved-bundle`. A ressíntese resolve o alcance desses pacotes com as estratégias de mesclagem do esquema durante `resynth()` e produz o pacote resolvido no resultado.

### Usar estratégias de mesclagem

Você pode usar uma estratégia de mesclagem oferecida pela biblioteca de esquemas. Essas estratégias fornecem maneiras de resolver saídas de arquivos e conflitos para arquivos mencionados na seção [Gerar arquivos com ressíntese](#).

- `alwaysUpdate` - Uma estratégia que sempre se resolve com o arquivo proposto.
- `neverUpdate` - Uma estratégia que sempre se resolve com o arquivo existente.
- `onlyAdd` - Uma estratégia que se resolve com o arquivo proposto quando um arquivo existente ainda não existe. Caso contrário, se resolve com o arquivo existente.
- `threeWayMerge` - Uma estratégia que executa uma fusão tripla entre os arquivos ancestrais existentes, propostos e comuns. O arquivo resolvido pode conter marcadores de conflito se os arquivos não puderem ser mesclados de forma limpa. O conteúdo dos arquivos fornecidos deve ser codificado em UTF-8 para que a estratégia produza uma saída significativa. A estratégia tenta detectar se os arquivos de entrada são binários. Se a estratégia detectar um conflito de mesclagem em um arquivo binário, ela sempre retornará o arquivo proposto.
- `preferProposed` - Uma estratégia que executa uma fusão tripla entre os arquivos ancestrais existentes, propostos e comuns. Essa estratégia resolve conflitos selecionando o lado do arquivo proposto de cada conflito.
- `preferExisting` - Uma estratégia que executa uma fusão tripla entre os arquivos ancestrais existentes, propostos e comuns. Essa estratégia resolve conflitos selecionando o lado do arquivo existente de cada conflito.

Para ver o código-fonte das estratégias de mesclagem, consulte o [repositório de código aberto do GitHub](#).

## Especificação de arquivos para atualizações de gerenciamento do ciclo de vida

Durante a ressíntese, os esquemas controlam como as alterações são mescladas em um repositório de origem existente. No entanto, você talvez não queira enviar atualizações para todos os arquivos do seu esquema. Por exemplo, códigos de exemplo, como folhas de estilo CSS, devem ser específicos do projeto. A estratégia de mesclagem de três vias é a opção padrão se você não especificar outra estratégia. Os esquemas podem especificar quais arquivos possuem e quais não possuem, especificando estratégias de mesclagem no próprio constructo do repositório. Os esquemas podem atualizar as estratégias de mesclagem, e as estratégias mais recentes podem ser usadas durante a ressíntese.

```
const sourceRepo = new SourceRepository(this, {
 title: 'my-repo',
});
sourceRepo.setResynthStrategies([
 {
 identifier: 'dont-override-sample-code',
 description: 'This strategy is applied accross all sample code. The blueprint
will create sample code, but skip attempting to update it.',
 strategy: MergeStrategies.neverUpdate,
 globs: [
 '**/src/**',
 '**/css/**',
],
 },
]);
```

Várias estratégias de mesclagem podem ser especificadas, e a última estratégia tem precedência. Os arquivos descobertos usam como padrão uma mesclagem de três vias semelhante ao Git. Existem várias estratégias de mesclagem fornecidas por meio do constructo `MergeStrategies`, mas você pode escrever a sua própria. As estratégias fornecidas seguem o driver de [estratégia de mesclagem do Git](#).

### Escrita de estratégias de mesclagem

Além de usar uma das estratégias de mescla de compilações fornecidas, você também pode escrever suas próprias estratégias. As estratégias devem seguir uma interface padrão. Você deve escrever uma função de estratégia que pegue versões de um arquivo do `existing-bundle`, do `proposed-bundle` e do `ancestor-bundle`, e mesclá-las em um único arquivo resolvido. Por exemplo:

```

type StrategyFunction = (
 /**
 * file from the ancestor bundle (if it exists)
 */
 commonAncestorFile: ContextFile | undefined,
 /**
 * file from the existing bundle (if it exists)
 */
 existingFile: ContextFile | undefined,
 /**
 * file from the proposed bundle (if it exists)
 */
 proposedFile: ContextFile | undefined,
 options?: {})
 /**
 * Return: file you'd like in the resolved bundle
 * passing undefined will delete the file from the resolved bundle
 */
=> ContextFile | undefined;

```

Se os arquivos não existirem (forem indefinidos), esse caminho de arquivo não existirá nesse pacote de localização específico.

Exemplo de:

```

strategies: [
 {
 identifier: 'dont-override-sample-code',
 description: 'This strategy is applied across all sample code. The
 blueprint will create sample code, but skip attempting to update it.',
 strategy: (ancestor, existing, proposed) => {
 const resolvedfile = ...
 ...
 // do something
 ...
 return resolvedfile
 },
 globs: [
 '**/src/**',
 '**/css/**',
],
 },
],

```



```
],
```

## Acesso a objetos de contexto para saber detalhes do projeto

Como autor do esquema, você pode acessar o contexto do projeto do esquema durante a síntese para ter informações, como espaço e nomes de projetos, ou arquivos existentes no repositório de origem de um projeto. Você também pode saber detalhes, como a fase de ressíntese que o esquema está gerando. Por exemplo, você pode acessar o contexto para saber se está ressintetizando para gerar um pacote ancestral ou um pacote proposto. O contexto de código existente pode ser usado para transformar o código em seu repositório. Por exemplo, você pode escrever sua própria estratégia de ressíntese para definir padrões de código específicos. A estratégia pode ser adicionada ao arquivo `blueprint.ts` para pequenos esquemas, ou você pode criar um arquivo separado para estratégias.

O exemplo a seguir mostra como você pode encontrar arquivos no contexto de um projeto, definir um criador de fluxo de trabalho e definir uma estratégia de ressíntese fornecida por um esquema para um arquivo específico:

```
const contextFiles = this.context.project.src.findAll({
 fileGlobs: ['**/package.json'],
});

// const workflows = this.context.project.src.findAll({
// fileGlobs: ['**/.codecatalyst/**/*.yaml'],
// });

const security = new WorkflowBuilder(this, {
 Name: 'security-workflow',
});
new Workflow(this, repo, security.getDefinition());
repo.setResynthStrategies([
 {
 identifier: 'force-security',
 globs: ['**/.codecatalyst/security-workflow.yaml'],
 strategy: MergeStrategies.alwaysUpdate,
 },
]);

for (const contextFile of contextFiles) {
 const packageObject = JSON.parse(contextFile.buffer.toString());
```

```
new SourceFile(internalRepo, contextFile.path, JSON.stringify({
 ...packageObject,
 }, null, 2));
}
```

## Desenvolvimento de um esquema personalizado para satisfazer os requisitos do projeto

Antes de publicar um esquema personalizado, você pode desenvolver o esquema para satisfazer requisitos específicos. Você pode desenvolver seu esquema personalizado e testar o esquema criando um projeto durante a pré-visualização. Você pode desenvolver o esquema personalizado para incluir componentes do projeto, como código-fonte específico, conexões de conta, fluxos de trabalho, problemas ou qualquer outro componente que possa ser criado no CodeCatalyst.

### Important

Se você quiser usar pacotes de esquema de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos esquemas personalizados que adiciona ao seu espaço e pelo código que eles geram.

### Important

Para desenvolver um esquema personalizado ao espaço do CodeCatalyst, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

Para desenvolver ou atualizar um esquema personalizado

1. Retome o Ambiente de Desenvolvimento. Para obter mais informações, consulte [Retomar um Ambiente de Desenvolvimento](#).

Caso você não tenha um Ambiente de Desenvolvimento, primeiro será necessário criar um. Para obter mais informações, consulte [Criar um Ambiente de Desenvolvimento](#).

2. Abra um terminal funcional no Ambiente de Desenvolvimento.

3. Se você optou por um fluxo de trabalho de lançamento ao criar seu esquema, a versão mais recente do esquema será publicada automaticamente. Faça as alterações para garantir que o arquivo `package.json` tenha a versão incrementada. Use o seguinte comando:

```
git pull
```

4. No arquivo `src/blueprint.ts`, edite as opções do esquema personalizado. A interface `Options` é interpretada dinamicamente pelo assistente do CodeCatalyst para gerar uma interface de usuário (UI) de seleção. Você pode desenvolver seu esquema personalizado adicionando componentes e tags compatíveis. Para obter mais informações, consulte [Modificar os recursos do esquema com um assistente de frontend](#), [Adicionar componentes do ambiente a um esquema](#), [Adição de componentes da região a um esquema](#), [Adicionar componentes de repositório e código-fonte a um esquema](#), [Adição de componentes de fluxo de trabalho a um esquema](#) e [Adicionar componentes do Ambiente de Desenvolvimento a um esquema](#).

Você também pode visualizar o SDK de esquemas e exemplos de esquema para ter suporte adicional ao desenvolver seu esquema personalizado. Para ter mais informações, consulte o [repositório de código aberto do GitHub](#).

Esquemas personalizados fornecem pacotes de pré-visualização como resultado de uma síntese bem-sucedida. O pacote do projeto representa o código-fonte, a configuração e os recursos em um projeto e é usado pelas operações da API de implantação do CodeCatalyst para implantar em um projeto. Se você quiser continuar desenvolvendo seu esquema personalizado, execute novamente o processo de síntese do esquema. Para obter mais informações, consulte [Conceitos de esquemas personalizados](#).

## Modificar os recursos do esquema com um assistente de frontend

Um assistente de seleção de esquema no CodeCatalyst é gerado automaticamente pela interface de `Options` no arquivo `blueprint.ts`. O assistente de front-end é compatível com modificações e recursos de `Options` do esquema usando [comentários e tags no estilo JSDOC](#). Use comentários e tags no estilo JSDOC para realizar tarefas. Por exemplo, você pode selecionar o texto exibido acima de uma opção, habilitar recursos, como validação de entrada, ou tornar uma opção recolhível. O assistente funciona interpretando uma árvore de sintaxe abstrata (AST) gerada a partir do tipo TypeScript da interface `Options`. O assistente é configurado automaticamente para o tipo descrito da melhor maneira possível. Nem todos os tipos são compatíveis. Outros tipos compatíveis incluem o seletor de região e o seletor de ambiente.

Veja a seguir um exemplo de um assistente que usa comentários e tags JSDOC com Options do esquema:

```
export interface Options {
 /**
 * What do you want to call your new blueprint?
 * @validationRegex /^[a-zA-Z0-9_]+$/
 * @validationMessage Must contain only upper and lowercase letters, numbers and
underscores
 */
 blueprintName: string;

 /**
 * Add a description for your new blueprint.
 */
 description?: string;

 /**
 * Tags for your Blueprint:
 * @collapsed true
 */
 tags?: string[];
}
```

O nome de exibição de cada opção da interface de Options aparece em camelCase por padrão. O texto simples no comentário no estilo JSDOC é exibido como texto acima da opção no assistente.

## Tópicos

- [Tags compatíveis](#)
- [Tipos de TypeScript compatíveis](#)
- [Comunicação com o usuário durante a síntese](#)

## Tags compatíveis

As seguintes tags JSDOC são compatíveis com Options do esquema personalizado no assistente de front-end.

@inlinePolicy ./path/to/policy/file.json

- Requer: opção de ser um tipo Role.

- **Uso** – Permite comunicar as políticas em linha necessárias para um perfil. Espera-se que o caminho `policy.json` esteja no código-fonte. Use essa tag quando precisar de uma política personalizada para um perfil.
- **Dependências:** `blueprint-cli 0.1.12` e posterior
- **Exemplo:** `@inlinePolicy ./deployment-policy.json`

```
environment: EnvironmentDefinition{
 awsAccountConnection: AccountConnection{
 /**
 * @inlinePolicy ./path/to/deployment-policy.json
 */
 cdkRole: Role[];
 };
};
```

`@trustPolicy ./path/to/policy/file.json`

- **Requer:** opção de ser um tipo `Role`.
- **Uso:** permite comunicar as políticas de confiança necessárias para um perfil. Espera-se que o caminho `policy.json` esteja no código-fonte. Use essa tag quando precisar de uma política personalizada para um perfil.
- **Dependências:** `blueprint-cli 0.1.12` e posterior
- **Exemplo:** `@trustPolicy ./trust-policy.json`

```
environment: EnvironmentDefinition{
 awsAccountConnection: AccountConnection{
 /**
 * @trustPolicy ./path/to/trust-policy.json
 */
 cdkRole: Role[];
 };
};
```

**Expressão Regex** `@validationRegex`

- **Requer** – Opção de ser uma string.

- Uso – Executa a validação de entrada na opção usando a expressão regex e exibições @validationMessage fornecidas.
- Exemplo: @validationRegex /^[a-zA-Z0-9\_]+\$
- Recomendação – Use com @validationMessage. A mensagem de validação está vazia por padrão.

string @validationMessage

- Requer – @validationRegex ou outros erros para revisar o uso.
- Uso – Exibe mensagem de validação quando houver falha em @validation\*.
- Exemplo – @validationMessage Must contain only upper and lowercase letters, numbers, and underscores.
- Recomendação – Use com @validationMessage. A mensagem de validação está vazia por padrão.

booleano @collapsed (opcional)

- Requer – N/A
- Uso – Booleano que permite que uma subopção seja recolhível. Se a anotação recolhida estiver presente, seu valor padrão será verdadeiro. Definir o valor como @collapsed false cria uma seção recolhível que é inicialmente aberta.
- Exemplo: @collapsed true

string @displayName

- Requer – N/A
- Uso – Altera o nome de exibição da opção. Permite formatos diferentes de camelCase para o nome de exibição.
- Exemplo: @displayName Blueprint Name

string @displayName

- Requer – N/A

- Uso – Altera o nome de exibição da opção. Permite formatos diferentes de [camelCase](#) para o nome de exibição.
- Exemplo: `@displayName Blueprint Name`

número `@defaultEntropy`

- Requer – Opção de ser uma string.
- Uso – Anexa uma string alfanumérica randomizada de um comprimento especificado à opção.
- Exemplo: `@defaultEntropy 5`

string `@placeholder` (opcional)

- Requer – N/A
- Uso – Altera o espaço reservado padrão para o campo de texto.
- Exemplo: `@placeholder type project name here`

número `@textArea` (opcional)

- Requer – N/A
- Uso – Converte a entrada de string em um componente de área de texto para seções maiores de texto. Adicionar um número define o número de linhas. O padrão é cinco linhas.
- Exemplo: `@textArea 10`

booleano `@hidden` (opcional)

- Requer – N/A
- Uso – Oculta o arquivo do usuário, a menos que a verificação de validação falhe. O valor padrão é verdadeiro.
- Exemplo: `@hidden`

booleano `@button` (opcional)

- Requer – N/A

- Uso – A anotação deve estar em uma propriedade booleana. Adiciona um botão que será sintetizado como verdadeiro quando selecionado. Não é uma alternância.
- Exemplo: `buttonExample: boolean;`

```
/**
 * @button
 */
buttonExample: boolean;
```

booleano `@showName` (opcional)

- Requer – N/A
- Uso – Só pode ser usado em um tipo de conexão de conta. Mostra a entrada de nome oculto. O padrão é `default_environment`.
- Exemplo: `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
 ...
}>;
```

booleano `@showEnvironmentType` (opcional)

- Requer – N/A
- Uso – Só pode ser usado em um tipo de conexão de conta. Mostra o menu suspenso do tipo de ambiente oculto. Todas as conexões são padronizadas para `production`. As opções são `Não produção` ou `Produção`.
- Exemplo: `@showEnvironmentType true`

```
/**
 * @showEnvironmentType true
 */
accountConnection: AccountConnection<{
 ...
}>;
```



## booleano @forceDefault (opcional)

- Requer – N/A
- Uso – Usa o valor padrão fornecido pelo autor do esquema em vez do valor usado anteriormente pelo usuário.
- Exemplo: `forceDefaultExample: any;`

```
/**
 * @forceDefault
 */
forceDefaultExample: any;
```

## @requires blueprintName

- Requer – Anota a interface de `Options`.
- Uso – Avisa o usuário para adicionar o `blueprintName` especificado ao projeto como um requisito para o esquema atual.
- Exemplo: `@requires '@amazon-codecatalyst/blueprints.blueprint-builder'`

```
/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
 ...
}
```

## regex @filter

- Requer – Anota a interface de `Selector` ou `MultiSelect`.
- Uso – Lista suspensa de filtros no assistente para opções que correspondem ao regex especificado.
- Exemplo: `@filter /blueprintPackageName/`

```
/**
 * @filter /myPackageName/
 */
blueprintInstantiation?: Selector<BlueprintInstantiation>;
```

...

## Tipos de TypeScript compatíveis

Os seguintes tipos TypeScript são compatíveis com `Options` do esquema personalizado no assistente de front-end.

### Número

- Requer: opção de ser um tipo `number`.
- Uso – Gere um campo de entrada numérica.
- Exemplo: `age: number`

```
{
 age: number
 ...
}
```

### String

- Requer: opção de ser um tipo `string`.
- Uso – Gere uma string de entrada numérica.
- Exemplo: `name: string`

```
{
 age: string
 ...
}
```

### Lista de strings

- Requer – Opção de ser uma array do tipo `string`.
- Uso – Gerar uma entrada de lista de strings.
- Exemplo: `isProduction: boolean`

```
{
```

```
isProduction: boolean
...
}
```

## Checkbox

- Requer – Opção de ser um boolean.
- Uso – Gere uma caixa de seleção.
- Exemplo: `isProduction: boolean`

```
{
 isProduction: boolean
 ...
}
```

## Seleção

- Requer – Opção de ser uma união de três ou menos strings.
- Uso – Gere um botão de seleção selecionado.

### Note

Quando há quatro ou mais itens, esse tipo é renderizado como uma lista suspensa.

- Exemplo: `color: 'red' | 'blue' | 'green'`

```
{
 color: 'red' | 'blue' | 'green'
 ...
}
```

## Dropdown

- Requer – Opção de ser uma união de quatro ou mais strings.
- Uso – Gere uma lista suspensa.
- Exemplo: `runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'`

```
{
 runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'
 ...
}
```

## Seção expansível

- Requer – Opção de ser um objeto.
- Uso – Gere uma seção expansível. As opções no objeto serão aninhadas dentro da seção expansível do assistente.
- Exemplo –

```
{
 expandableSectionTitle: {
 nestedString: string;
 nestedNumber: number;
 }
}
```

## Tupla

- Requer – Opção de ser do tipo Tuple.
- Uso – Gere uma entrada paga de valor-chave.
- Exemplo: tuple: Tuple[string, string]>

```
{
 tuple: Tuple[string, string]>;
 ...
}
```

## Lista de tuplas

- Requer – Opção de ser uma array do tipo Tuple.
- Uso – Gere uma entrada de lista de tuplas.
- Exemplo: tupleList: Tuple[string, string]>[]

```
{
 tupleList: Tuple[string, string]>[];
 ...
}
```

## Seletor

- Requer – Opção de ser do tipo `Selector`.
- Uso – Gere uma lista suspensa de repositórios de origem ou esquemas aplicados a um projeto.
- Exemplo: `sourceRepo: Selector<SourceRepository>`

```
{
 sourceRepo: Selector<SourceRepository>;
 sourceRepoOrAdd: Selector<SourceRepository | string>;
 blueprintInstantiation: Selector<BlueprintInstantiation>;
 ...
}
```

## Seleção múltipla

- Requer – Opção de ser do tipo `Selector`.
- Uso – Gere uma entrada de seleção múltipla.
- Exemplo: `multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>`

```
{
 multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>;
 ...
}
```

## Comunicação com o usuário durante a síntese

Como autor do esquema, você pode se comunicar com os usuários além das mensagens de validação. Por exemplo, um membro do espaço pode visualizar uma combinação de opções que produz um esquema que não está claro. Os esquemas personalizados são compatíveis com a capacidade de comunicar mensagens de erro aos usuários invocando a síntese. O esquema básico implementa um perfil de `throwSynthesisError(...)` que espera uma mensagem de erro clara. Você pode invocar a mensagem usando o seguinte:

```
//blueprint.ts
this.throwSynthesisError({
 name: BlueprintSynthesisErrorTypes.BlueprintSynthesisError,
 message: 'hello from the blueprint! This is a custom error communicated to the
user.'
})
```

## Geração de entradas e nova renderização de os elementos do assistente de frontend

Você pode gerar entradas de assistente com o `DynamicKVInput` e criar dinamicamente elementos de assistente de frontend para o esquema personalizado.

### Tópicos

- [Criar Ambientes de Desenvolvimento](#)
- [Criar dinamicamente elementos de assistente](#)

### Criar Ambientes de Desenvolvimento

O tipo `DynamicKVInput` pode ser usado para gerar entradas de assistente de frontend usando os padrões do esquema personalizado. Para ver o esquema mais atualizado, consulte a [definição de `DynamicKVInput`](#).

O exemplo a seguir mostra como é possível usar `Options` para moldar um objeto:

```
import { DynamicKVInput } from '@amazon-codecatalyst/blueprints.blueprint';

export interface Options extends ParentOptions {

 parameters: DynamicKVInput[];

}
```

O exemplo a seguir mostra como é possível configurar parâmetros padrão com várias propriedades:

```
{
 "parameters": [
 {
 "key": "AWS_REGION",
 "value": "us-west-2",
 "displayType": "region",
```

```
 "possibleValues": [
 "us-west-1",
 "us-west-2",
 "us-east-1",
 "us-east-2"
],
 "displayName": "AWS Region",
 "description": "AWS Region to deploy the solution to."
 },
 {
 "key": "SchedulingActive",
 "value": "Yes",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Scheduling Active",
 "description": "Activate or deactivate scheduling."
 },
 {
 "key": "ScheduledServices",
 "value": "Both",
 "displayType": "dropdown",
 "possibleValues": [
 "EC2",
 "RDS",
 "Both"
],
 "displayName": "Scheduled Services",
 "description": "Services to schedule."
 },
 {
 "key": "ScheduleRdsClusters",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Schedule RDS Clusters",
 "description": "Enable scheduling of Aurora clusters for RDS service."
 },
 {
```

```

 "key": "CreateRdsSnapshot",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Create RDS Snapshot",
 "description": "Create snapshot before stopping RDS instances (does not
apply to Aurora Clusters).",
 },
 {
 "key": "MemorySize",
 "value": "128",
 "displayType": "dropdown",
 "possibleValues": [
 "128",
 "384",
 "512",
 "640",
 "768",
 "896",
 "1024",
 "1152",
 "1280",
 "1408",
 "1536"
],
 "displayName": "Memory Size",
 "description": "Size of the Lambda function running the scheduler, increase
size when processing large numbers of instances.",
 },
 {
 "key": "UseCloudWatchMetrics",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Use CloudWatch Metrics",
 "description": "Collect instance scheduling data using CloudWatch metrics."
 },
 {

```



```
 "key": "LogRetentionDays",
 "value": "30",
 "displayType": "dropdown",
 "possibleValues": [
 "1",
 "3",
 "5",
 "7",
 "14",
 "30",
 "60",
 "90",
 "120",
 "150",
 "180",
 "365",
 "400",
 "545",
 "731",
 "1827",
 "3653"
],
 "displayName": "Log Retention Days",
 "description": "Retention days for scheduler logs."
 },
 {
 "key": "Trace",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Trace",
 "description": "Enable debug-level logging in CloudWatch logs."
 },
 {
 "key": "EnableSSMMaintenanceWindows",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
]
 }
],
```

```

 "displayName": "Enable SSM Maintenance Windows",
 "description": "Enable the solution to load SSM Maintenance Windows, so
that they can be used for EC2 instance Scheduling."
 },
 {
 "key": "DefaultTimezone",
 "value": "UTC",
 "displayType": "string",
 "displayName": "Default Timezone",
 "description": "Default timezone to use for scheduling."
 },
 {
 "key": "Regions",
 "value": "us-west-2",
 "displayType": "string",
 "displayName": "Regions",
 "description": "List of regions in which instances should be scheduled,
leave blank for current region only."
 },
 {
 "key": "UsingAWSOrganizations",
 "value": "No",
 "displayType": "dropdown",
 "possibleValues": [
 "Yes",
 "No"
],
 "displayName": "Using AWS Organizations",
 "description": "Use AWS Organizations to automate spoke account
registration."
 },
 {
 "key": "Principals",
 "displayType": "string",
 "optional": false,
 "displayName": "Principals",
 "description": "(Required) If using AWS Organizations, provide the
Organization ID. Eg. o-xxxxyyy. Else, provide a comma separated list of spoke account
ids to schedule. Eg.: 1111111111, 2222222222 or {param: ssm-param-name}"
 },
 {
 "key": "Namespace",
 "value": "Default",
 "displayType": "string",

```

```

 "displayName": "Namespace",
 "description": "Provide unique identifier to differentiate between multiple
solution deployments (No Spaces). Example: Dev"
 },
 {
 "key": "SchedulerFrequency",
 "value": 5,
 "displayType": "number",
 "displayName": "Scheduler Frequency",
 "description": "Scheduler running frequency in minutes."
 }
]
}

```

## Criar dinamicamente elementos de assistente

O mesmo esquema da criação de entradas do assistente pode ser usado para renderizar dinamicamente um assistente durante a síntese. Isso pode ser usado para responder às perguntas de acompanhamento de um usuário quando necessário.

```

//blueprint.ts

export interface Options extends ParentOptions {
 ...
 dynamicOptions: OptionsSchemaDefinition<'optionsIdentifier', KVSchema>;
}

```

O assistente pode ser configurado durante o período de síntese usando um componente de Options.

```

import {
 OptionsSchemaDefinition,
 OptionsSchema,
} from '@amazon-codecatalyst/blueprints.blueprint';

...

// dynamically renders a number in the place where 'optionsIdentifier' was set in the
original options type.
new OptionsSchema<KVSchema>(this, 'optionsIdentifier', [
 {
 "key": "SchedulerFrequency",

```

```
 "value": 5,
 "displayType": "number",
 "displayName": "Scheduler Frequency",
 "description": "Scheduler running frequency in minutes."
 }
]);
```

## Adicionar componentes do ambiente a um esquema

O assistente de esquema personalizado é gerado dinamicamente na interface `Options` exposta por meio do assistente. Os esquemas oferecem suporte à geração de componentes de interface de usuário (UI) a partir de tipos expostos.

Para importar componentes de ambiente de esquema do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

### Tópicos

- [Criar Ambientes de Desenvolvimento](#)
- [Lista de ambientes](#)
- [Simular exemplos de interface](#)

## Criar Ambientes de Desenvolvimento

O exemplo a seguir mostra como implantar a aplicação na nuvem:

```
export interface Options extends ParentOptions {
 ...
 myNewEnvironment: EnvironmentDefinition{
 thisIsMyFirstAccountConnection: AccountConnection{
 thisIsARole: Role['lambda', 's3', 'dynamo'];
 };
 };
}
```

A interface gera um componente de interface que solicita um novo ambiente (`myNewEnvironment`) com uma única conexão de conta (`thisIsMyFirstAccountConnection`). Um perfil na conexão da conta (`thisIsARole`) também é gerado com `['lambda', 's3', 'dynamo']` como os

recursos de perfil mínimos necessários. Nem todos os usuários têm conexões de conta, então você deve verificar o caso em que um usuário não conecta uma conta ou não conecta uma conta com um perfil. Os perfis também podem ser anotados com `@inlinePolicies`. Para obter mais informações, consulte [@inlinePolicy ./path/to/policy/file.json](#).

O componente de ambiente requer um `name` e `environmentType`. O código a seguir é a forma padrão mínima exigida:

```
{
 ...
 "myNewEnvironment": {
 "name": "myProductionEnvironment",
 "environmentType": "PRODUCTION"
 },
}
```

Em seguida, o componente de interface solicita vários campos. Conforme você preenche os campos, o esquema ganha uma forma totalmente expandida. Pode ser útil incluir a simulação completa no arquivo `defaults.json` para fins de teste e desenvolvimento.

### Lista de ambientes

Especificar uma matriz do tipo `EnvironmentDefinition` gerará uma lista de ambientes na interface do assistente.

```
export interface Options extends ParentOptions {
 ...
 /**
 * @showName readOnly
 */
 myEnvironments: EnvironmentDefinition<{
 thisIsMyFirstAccountConnection: AccountConnection<{
 thisIsARole: Role<['lambda', 's3', 'dynamo']>;
 }>;
 }>[];
}
```

O exemplo a seguir mostra os padrões de uma lista de ambientes:

```
{
```

```

...
"myEnvironments": [
{
 "name": "myProductionEnvironment",
 "environmentType": "PRODUCTION"
},
{
 "name": "myDevelopmentEnvironment",
 "environmentType": "DEVELOPMENT"
},
]
}

```

## Simular exemplos de interface

### Interface simulada simples

```

{
 ...
 "thisIsMyEnvironment": {
 "name": "myProductionEnvironment",
 "environmentType": "PRODUCTION",
 "thisIsMySecondAccountConnection": {
 "id": "12345678910",
 "name": "my-account-connection-name",
 "secondAdminRole": {
 "arn": "arn:aws:iam::12345678910:role/ConnectedQuokkaRole",
 "name": "ConnectedQuokkaRole",
 "capabilities": [
 "lambda",
 "s3",
 "dynamo"
]
 }
 }
 }
}

```

### Interface simulada complexa

```

export interface Options extends ParentOptions {
 /**
 * The name of an environment

```

```

 * @displayName This is a Environment Name
 * @collapsed
 */
thisIsMyEnvironment: EnvironmentDefinition{
 /**
 * comments about the account that is being deployed into
 * @displayName This account connection has an overridden name
 * @collapsed
 */
 thisIsMyFirstAccountConnection: AccountConnection{
 /**
 * Blah blah some information about the role that I expect
 * e.g. here's a copy-pastable policy: [to a link]
 * @displayName This role has an overridden name
 */
 adminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
 /**
 * Blah blah some information about the second role that I expect
 * e.g. here's a copy-pastable policy: [to a link]
 */
 lambdaRole: Role['lambda', 's3'];
 };
 /**
 * comments about the account that is being deployed into
 */
 thisIsMySecondAccountConnection: AccountConnection{
 /**
 * Blah blah some information about the role that I expect
 * e.g. here's a copy-pastable policy: [to a link]
 */
 secondAdminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
 /**
 * Blah blah some information about the second role that I expect
 * e.g. here's a copy-pastable policy: [to a link]
 */
 secondLambdaRole: Role['lambda', 's3'];
 };
};
}

```

## Interface simulada completa

```
{
```

```
...
"thisIsMyEnvironment": {
 "name": "my-production-environment",
 "environmentType": "PRODUCTION",
 "thisIsMySecondAccountConnection": {
 "id": "12345678910",
 "name": "my-connected-account",
 "secondAdminRole": {
 "name": "LambdaQuokkaRole",
 "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
 "capabilities": [
 "admin",
 "lambda",
 "s3",
 "cloudfront"
]
 },
 "secondLambdaRole": {
 "name": "LambdaQuokkaRole",
 "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
 "capabilities": [
 "lambda",
 "s3"
]
 }
 },
 "thisIsMyFirstAccountConnection": {
 "id": "12345678910",
 "name": "my-connected-account",
 "adminRole": {
 "name": "LambdaQuokkaRole",
 "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
 "capabilities": [
 "admin",
 "lambda",
 "s3",
 "cloudfront"
]
 },
 "lambdaRole": {
 "name": "LambdaQuokkaRole",
 "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
 "capabilities": [
 "lambda",
```



```
 "s3"
]
 }
 },
},
}
```

## Adicionar componentes de segredos a um esquema

Os segredos podem ser usados no CodeCatalyst para armazenar dados confidenciais que podem ser referenciados em fluxos de trabalho. Você pode adicionar um segredo ao seu esquema personalizado e referenciá-lo em seu fluxo de trabalho. Para obter mais informações, consulte [Mascarar dados usando segredos](#).

Como importar o tipo de região de esquemas do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-
component.secrets'
```

## Tópicos

- [Criar um segredo](#)
- [Fazer referência a um segredo em um fluxo de trabalho](#)

## Criar um segredo

O exemplo a seguir cria um componente de interface do usuário que solicita que o usuário insira um valor de segredo e uma descrição opcional:

```
export interface Options extends ParentOptions {
 ...
 mySecret: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
 constructor(options_: Options) {
 new Secret(this, options.secret);
 }
}
```

```
}
```

O componente secreto requer um name. O código a seguir é a forma padrão mínima exigida:

```
{
 ...
 "secret": {
 "name": "secretName"
 },
}
```

### Fazer referência a um segredo em um fluxo de trabalho

O esquema de exemplo a seguir cria um segredo e um fluxo de trabalho que faz referência ao valor do segredo. Para obter mais informações, consulte [Fazer referência a um segredo em um fluxo de trabalho](#).

```
export interface Options extends ParentOptions {
 ...
 /**
 *
 * @validationRegex /^\\w+$/
 */
 username: string;

 password: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
 constructor(options_: Options) {
 const password = new Secret(this, options_.password);

 const workflowBuilder = new WorkflowBuilder(this, {
 Name: 'my_workflow',
 });

 workflowBuilder.addBuildAction({
 actionName: 'download_files',
```

```
 input: {
 Sources: ['WorkflowSource'],
 },
 output: {
 Artifacts: [{ Name: 'download', Files: ['file1'] }],
 },
 steps: [
 `curl -u ${options_.username}:${password.reference} https://example.com`,
],
 });

 new Workflow(
 this,
 repo,
 workflowBuilder.getDefinition(),
);
}
```

Para saber mais sobre como usar segredos no CodeCatalyst, consulte [Mascarar dados usando segredos](#).

## Adição de componentes da região a um esquema

O tipo de região pode ser adicionado à interface de Options do seu esquema personalizado para gerar um componente no assistente de esquema; é possível inserir uma ou mais regiões da AWS. O tipo de região pode ser importado do seu esquema base em seu arquivo `blueprint.ts`. Para ter mais informações, consulte [Regiões da AWS](#).

Como importar o tipo de região de esquemas do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import { Region } from '@amazon-codecatalyst/blueprints.blueprint'
```

O parâmetro de tipo de região é uma matriz de códigos de região da AWS para escolher, ou você pode usar `*` para incluir todas as regiões da AWS compatíveis.

### Tópicos

- [Anotações](#)
- [Exemplos de componentes de região](#)

## Anotações

As tags JSDoc podem ser adicionadas a cada campo na interface de `Options` para personalizar como um campo aparece e se comporta no assistente. Para o tipo de região, as seguintes tags são compatíveis:

- A anotação `@displayName` pode ser usada para alterar o rótulo do campo no assistente.

Exemplo: `@displayName AWS Region`

- A anotação `@placeholder` pode ser usada para alterar o espaço reservado do componente de seleção/seleção múltipla.

Exemplo: `@placeholder Choose AWS Region`

## Exemplos de componentes de região

### Escolha de uma região de uma lista especificada

```
export interface Options extends ParentOptions {
 ...
 /**
 * @displayName Region
 */
 region: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>;
}
```

### Escolha de uma ou mais regiões de uma lista especificada

```
export interface Options extends ParentOptions {
 ...
 /**
 * @displayName Regions
 */
 multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

### Escolha de uma região da AWS

```
export interface Options extends ParentOptions {
 ...
 /**
```

```
* @displayName Region
*/
region: Region<['*']>;
}
```

Escolha de uma ou mais regiões de uma lista especificada

```
export interface Options extends ParentOptions {
 ...
 /**
 * @displayName Regions
 */
 multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

## Adicionar componentes de repositório e código-fonte a um esquema

Um repositório é usado pelo Amazon CodeCatalyst para armazenar código. O repositório usa um nome como entrada. A maioria dos componentes é armazenada em um repositório, como arquivos de código-fonte, fluxos de trabalho e outros componentes, como ambientes de desenvolvimento gerenciado (MDE). O componente do repositório de origem também exporta componentes usados para gerenciar arquivos e ativos estáticos. Os repositórios têm restrições de nome. Para obter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).

```
const repository = new SourceRepository(this, {
 title: 'my-new-repository-title',
});
```

Para importar os componentes de código-fonte e repositório de esquemas do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

## Tópicos

- [Adicionar um arquivo](#)
- [Adicionar um arquivo genérico](#)
- [Copiar arquivos](#)
- [Selecionar vários arquivos como destino](#)

- [Criação de um novo repositório e adição de arquivos](#)

## Adicionar um arquivo

Você pode gravar um arquivo de texto em um repositório com o constructo `SourceFile`. A operação é um dos casos de uso mais comuns e usa um repositório, um caminho de arquivo e conteúdo de texto. Se o caminho do arquivo não existir em um repositório, o componente criará todas as pastas necessárias.

```
new SourceFile(repository, `path/to/my/file/in/repo/file.txt`, 'my file contents');
```

### Note

Se você gravar dois arquivos no mesmo local no mesmo repositório, a implementação mais recente substituirá a anterior. Você pode usar o recurso para criar camadas de código gerado, o que é especialmente útil para estender o código que os esquemas personalizados podem ter gerado.

## Adicionar um arquivo genérico

Você pode gravar bits arbitrários no repositório. Você pode ler de um buffer e usar o constructo `File`.

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

## Copiar arquivos

Você pode começar com o código gerado copiando e colando o código inicial e, depois, gerando mais código sobre essa base. Coloque o código dentro do diretório `static-assets` e, depois, direcione esse código com o constructo `StaticAsset`. Nesse caso, o caminho sempre começa na raiz do diretório `static-assets`.

```
const starterCode = new StaticAsset('path/to/file/file.txt')
const starterCodeText = new StaticAsset('path/to/file/file.txt').toString()
const starterCodeRawContent = new StaticAsset('path/to/image/hello.png').content()
```

```
const starterCodePath = new StaticAsset('path/to/image/hello.png').path()
// starterCodePath is equal to 'path/to/image/hello.png'
```

Uma subclasse de `StaticAsset` é `SubstitutionAsset`. A subclasse funciona exatamente da mesma forma, mas, em vez disso, você pode executar uma substituição de mustache no arquivo. Isso pode ser útil para realizar a geração de estilos de copiar e substituir.

A substituição estática de ativos usa um mecanismo de modelos de mustache para renderizar os arquivos estáticos que são implantados no repositório de origem gerado. As regras de modelos de mustache são aplicadas durante a renderização, o que significa que todos os valores são codificados em HTML por padrão. Para renderizar HTML sem escape, use a sintaxe de mustache triplo `{{{name}}}`. Para ter mais informações, consulte as [regras de modelos de mustache](#).

#### Note

Executar um substituto em arquivos que não são interpretáveis por texto pode produzir erros.

```
const starterCodeText = new SubstitutionAsset('path/to/file/file.txt').substitute({
 'my_variable': 'subbed value1',
 'another_variable': 'subbed value2'
})
```

## Selecionar vários arquivos como destino

Os ativos estáticos oferecem suporte à segmentação global por meio de uma função estática em `StaticAsset` e das subclasses chamadas `findAll(...)`, o que retorna uma lista de ativos estáticos pré-carregados com caminhos, conteúdos e muito mais. Você pode encadear a lista com constructos `File` para copiar e colar conteúdo no diretório `static-assets`.

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

## Criação de um novo repositório e adição de arquivos

Você pode usar um componente de repositório para criar um repositório em um projeto gerado. Depois, você pode adicionar arquivos ou fluxos de trabalho ao repositório criado.

```
import { SourceRepository } from '@amazon-codecatalyst/codecatalyst-source-repositories';
...
const repository = new SourceRepository(this, { title: 'myRepo' });
```

O exemplo a seguir mostra como adicionar arquivos e fluxos de trabalho a um repositório existente:

```
import { SourceFile } from '@amazon-codecatalyst/codecatalyst-source-repositories';
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows';
...
new SourceFile(repository, 'README.md', 'This is the content of my readme');
new Workflow(this, repository, {/**...workflowDefinition...**/});
```

A combinação das duas partes do código gera um único repositório chamado myRepo com um arquivo de origem README.md e um fluxo de trabalho do CodeCatalyst na raiz.

## Adição de componentes de fluxo de trabalho a um esquema

Um fluxo de trabalho é usado pelos projetos do Amazon CodeCatalyst para executar ações com base em gatilhos. Você pode usar componentes de fluxo de trabalho para criar e reunir arquivos YAML do fluxo de trabalho. Para obter mais informações, consulte [Definição do YAML do fluxo de trabalho](#).

Para importar componentes de fluxos de trabalho de esquema do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

### Tópicos

- [Exemplos de componentes de fluxo de trabalho](#)
- [Conexão a um ambiente](#)

## Exemplos de componentes de fluxo de trabalho

### Componente WorkflowBuilder

Você pode usar uma classe para criar uma definição de fluxo de trabalho. A definição pode ser dada a um componente do fluxo de trabalho para renderização em um repositório.



```
import { WorkflowBuilder } from '@amazon-codecatalyst/codecatalyst-workflows'

const workflowBuilder = new WorkflowBuilder({} as Blueprint, {
 Name: 'my_workflow',
});

// trigger the workflow on pushes to branch 'main'
workflowBuilder.addBranchTrigger(['main']);

// add a build action
workflowBuilder.addAction({
 // give the action a name
 actionName: 'build_and_do_some_other_stuff',

 // the action pulls from source code
 input: {
 Sources: ['WorkflowSource'],
 },

 // the output attempts to autodiscover test reports, but not in the node modules
 output: {
 AutoDiscoverReports: {
 Enabled: true,
 ReportNamePrefix: AutoDiscovered,
 IncludePaths: ['**/*'],
 ExcludePaths: ['*/node_modules/**/*'],
 },
 },
});

// execute some arbitrary steps
steps: [
 'npm install',
 'npm run myscript',
 'echo hello-world',
],
// add an account connection to the workflow
environment: convertToWorkflowEnvironment(myEnv),
});
```

## Componente Projen de fluxo de trabalho

O exemplo a seguir mostra como um componente Projen pode ser usado para gravar um YAML de fluxo de trabalho em um repositório:

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'

...

const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// can also pass in any object and have it rendered as a yaml. This is unsafe and may
 not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

## Conexão a um ambiente

Muitos fluxos de trabalho precisam ser executados em uma conexão de conta da AWS. Os fluxos de trabalho lidam com isso permitindo que as ações se conectem a ambientes com especificações de nome de conta e perfil.

```
import { convertToWorkflowEnvironment } from '@amazon-codecatalyst/codecatalyst-
workflows'

const myEnv = new Environment(...);

// can be passed into a workflow constructor
const workflowEnvironment = convertToWorkflowEnvironment(myEnv);

// add a build action
workflowBuilder.addAction({
 ...
 // add an account connection to the workflow
 environment: convertToWorkflowEnvironment(myEnv),
});
```

## Adicionar componentes do Ambiente de Desenvolvimento a um esquema

Ambientes de desenvolvimento gerenciado (MDE) são usados para criar e configurar espaços de trabalho de MDE no CodeCatalyst. O componente gera um arquivo `devfile.yaml`. Para ter mais

informações, consulte [Introdução a arquivos de desenvolvimento](#) e [Edição do repositório devfile para um Ambiente de Desenvolvimento](#).

```
new Workspace(this, repository, SampleWorkspaces.default);
```

Como importar componentes de espaços de trabalho de esquema do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

## Adição de componentes de problema a um esquema

No CodeCatalyst, você pode monitorar recursos, tarefas, bugs e qualquer outro trabalho envolvido no projeto. Cada parte do trabalho é mantida em um registro distinto, chamado de problema. Cada problema pode ter uma descrição, um responsável, status e outras propriedades, que você pode pesquisar, agrupar e filtrar. Você pode visualizar os problemas usando as visualizações padrão ou criar suas próprias visualizações com filtragem, classificação ou agrupamento personalizados. Para ter mais informações sobre conceitos relacionados a problemas, consulte [Conceitos de problemas](#) e [Cotas para edições em CodeCatalyst](#).

O componente de problema gera uma representação JSON de um problema. O componente usa um campo de ID e a definição do problema como entrada.

Como importar componentes de problema de esquema do Amazon CodeCatalyst

No arquivo `blueprint.ts`, adicione:

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

## Tópicos

- [Exemplos de componentes de problema](#)

## Exemplos de componentes de problema

### Criar um problema

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
```

```
title: 'myFirstIssue',
content: 'This is an example issue.',
});
```

## Criar um problema de alta prioridade

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
...
const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// Creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// Can also pass in any object and have it rendered as a yaml. This is unsafe and may
not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

## Criar um problema de baixa prioridade com rótulos

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myThirdIssue', {
 title: 'myThirdIssue',
 content: 'This is an example of a low priority issue with a label.',
 priority: 'LOW',
 labels: ['exampleLabel'],
});
```

## Trabalhar com ferramentas e CLI do esquema

A [CLI do esquema](#) fornece ferramentas para gerenciar e trabalhar com esquemas personalizados.

### Tópicos

- [Trabalhar com ferramentas de esquema](#)
- [Ferramenta de upload de imagens](#)

## Trabalhar com ferramentas de esquema

### Como trabalhar com as ferramentas do esquema

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Retorne o Ambiente de Desenvolvimento. Para obter mais informações, consulte [Retomar um Ambiente de Desenvolvimento](#).

Caso você não tenha um Ambiente de Desenvolvimento, primeiro será necessário criar um. Para obter mais informações, consulte [Criar um Ambiente de Desenvolvimento](#).

3. Em um terminal em funcionamento, execute o seguinte comando para instalar a CLI do esquema:

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. No arquivo `blueprint.ts`, importe as ferramentas que você deseja usar no seguinte formato:

```
import { <tooling-function-name> } from '@amazon-codecatalyst/blueprint-util.cli/lib/<tooling-folder-name>/<tooling-file-name>;
```

#### Tip

É possível acessar o [CodeCatalyst blueprints GitHub repository](#) para encontrar o nome da ferramenta que deseja usar.

Se quiser usar a ferramenta de upload de imagens, adicione o seguinte ao script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

## Exemplos

- Se quiser usar a função de publicação, adicione o seguinte ao script:

```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/publish';
```

- Se quiser usar a ferramenta de upload de imagens, adicione o seguinte ao script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

## 5. Chame a função.

Examples:

- Se quiser usar a função de publicação, adicione o seguinte ao script:

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Se quiser usar a ferramenta de upload de imagens, adicione o seguinte ao script:

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/image');
```

### Ferramenta de upload de imagens

A ferramenta de upload de imagens permite fazer upload da própria imagem em um bucket do S3 na conta da AWS e, depois, distribuir essa imagem publicamente por trás do CloudFront. A ferramenta usa um caminho de imagem no armazenamento local (e o nome opcional do bucket) como entrada e exibe o URL para a imagem que está disponível publicamente. Para ter mais informações, consulte [O que é o Amazon CloudFront?](#) e [O que é o Amazon S3?](#)

### Como trabalhar com a ferramenta de upload de imagens

1. Clone o [repositório de esquemas de código aberto no GitHub](#) que concede acesso ao SDK e a exemplos de esquema. Em um terminal em funcionamento, execute o seguinte comando:

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Execute o seguinte comando para navegar até o repositório de esquemas no GitHub:

```
cd codecatalyst-blueprints
```

3. Execute o seguinte comando para instalar dependências:

```
yarn && yarn build
```

4. Execute o seguinte comando para garantir que a versão mais recente da CLI do esquema esteja instalada:

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Faça login na conta da AWS com o bucket do S3 no qual você deseja fazer upload da imagem. Para ter mais informações, consulte [Configure the AWS CLI](#) e [Sign in through the AWS Command Line Interface](#).
6. Execute o seguinte comando na raiz do repositório do CodeCatalyst para navegar até o diretório com a CLI do esquema:

```
cd packages/utils/blueprint-cli
```

7. Execute o seguinte comando para fazer upload da imagem em um bucket S3:

```
yarn blueprint upload-image-public <./path/to/your/image>
 <optional:optional-bucket-name>
```

Um URL para a imagem é gerado. O URL não estará disponível imediatamente, pois requer algum tempo para que a distribuição do CloudFront seja implantada. Confira o status da distribuição para saber o status de implantação mais recente. Para ter mais informações, consulte [Trabalhar com distribuições](#).

## Avaliar as alterações na interface com testes de snapshot

Testes de snapshots gerados em várias configurações do seu esquema são compatíveis.

Os esquemas são compatíveis com o [teste de snapshot](#) em configurações fornecidas por você como autor do esquema. As configurações são substituições parciais, mescladas sobre o arquivo `defaults.json` na raiz de um esquema. Quando o teste de snapshot é ativado e configurado, o processo de compilação e teste sintetiza as configurações fornecidas e verifica se as saídas sintetizadas não foram alteradas em relação ao snapshot de referência. Para visualizar o código de teste de snapshot, consulte o [repositório GitHub de esquemas do CodeCatalyst](#).

### Como habilitar o teste de snapshot

1. No arquivo `.projenrc.ts`, atualize o objeto de entrada para `ProjenBlueprint` com os arquivos que você deseja capturar. Por exemplo:

```
{

 blueprintSnapshotConfiguration: {
 snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],
 },
}
```

```
}
```

2. Sintetize novamente o esquema para criar arquivos TypeScript em seu projeto de esquema. Não edite os arquivos de origem, pois eles são mantidos e regenerados pelo Projen. Use o seguinte comando:

```
yarn projen
```

3. Navegue até o diretório `src/snapshot-configurations` para ver o arquivo `default-config.json` com um objeto vazio. Atualize ou substitua o arquivo por uma ou mais de suas próprias configurações de teste. Cada configuração de teste é então mesclada com o arquivo `defaults.json` do projeto, sintetizada e comparada aos instantâneos durante o teste. Use o comando a seguir para testar:

```
yarn test
```

A seguinte mensagem é exibida na primeira vez que você usa um comando de teste: `Snapshot Summary > NN snapshots written from 1 test suite`. Os testes subsequentes verificam se a saída sintetizada não foi alterada nos instantâneos e exibem a seguinte mensagem: `Snapshots: NN passed, NN total`.

Se você alterar intencionalmente seu esquema para produzir uma saída diferente, execute o comando a seguir para atualizar os snapshots de referência:

```
yarn test:update
```

Os snapshots esperam que as saídas sintetizadas sejam constantes entre cada execução. Se o seu esquema gerar arquivos que variam, você deve excluir esses arquivos do teste de snapshot. Atualize o objeto `blueprintSnapshotConfiguration` do seu objeto de entrada `ProjenBlueprint` para adicionar a propriedade `snapshotGlobs`. A propriedade `snapshotGlobs` é uma matriz de [globs](#) que determina quais arquivos são incluídos ou excluídos do snapshot.

#### Note

Há uma lista padrão de globs. Se você especificar sua própria lista, talvez seja necessário trazer de volta explicitamente as entradas padrão.



## Publicação de um esquema personalizado em um espaço

Antes de adicionar um esquema personalizado ao catálogo de esquemas do seu espaço, você deve publicá-lo no espaço. Você também pode visualizar o esquema no console do CodeCatalyst antes da publicação. Você pode publicar uma versão prévia ou uma versão normal do seu esquema.

### Important

Se você quiser usar pacotes de esquema de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos esquemas personalizados que adiciona ao seu espaço e pelo código que eles geram.

### Important

Para publicar um esquema personalizado no espaço do CodeCatalyst, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

### Tópicos

- [Visualização e publicação de uma versão prévia de um esquema personalizado](#)
- [Visualização e publicação de uma versão normal de um esquema personalizado](#)
- [Publicação e adição de um esquema personalizado em espaços e projetos especificados](#)

## Visualização e publicação de uma versão prévia de um esquema personalizado

Você pode publicar uma versão prévia do esquema personalizado no espaço se quiser adicioná-la ao catálogo de esquemas do espaço. Isso permite que você visualize o esquema como usuário antes de adicionar a versão sem pré-visualização ao catálogo. A versão prévia permite que você publique sem usar uma versão real. Por exemplo, se você trabalha em uma versão 0.0.1, pode publicar e adicionar uma versão prévia, para que novas atualizações para uma segunda versão possam ser publicadas e adicionadas como 0.0.2.

Depois de fazer as alterações, reconstrua o pacote do esquema personalizado executando o arquivo `package.json` e visualize as alterações.

Para visualizar e publicar uma versão prévia de um esquema personalizado

1. Retome o Ambiente de Desenvolvimento. Para obter mais informações, consulte [Retomar um Ambiente de Desenvolvimento](#).
2. Abra um terminal funcional no Ambiente de Desenvolvimento.
3. (Opcional) Em um terminal em funcionamento, instale as dependências necessárias do projeto, caso ainda não as tenha instalado. Use o seguinte comando:

```
yarn
```

4. (Opcional) Se você fez alterações no arquivo `.projenrc.ts`, gere novamente a configuração do projeto antes de criar e visualizar seu esquema. Use o seguinte comando:

```
yarn projen
```

5. Reconstrua e visualize seu esquema personalizado usando o comando a seguir. Use o seguinte comando:

```
yarn blueprint:preview
```

Navegue até o link `See this blueprint at:` fornecido para visualizar seu esquema personalizado. Confira se a interface do usuário, inclusive o texto, aparece conforme o esperado com base na configuração. Se quiser alterar o esquema personalizado, você pode editar o arquivo `blueprint.ts`, resintetizar o esquema e, depois, publicar uma versão prévia novamente. Para obter mais informações, consulte [Ressíntese](#).

6. (Opcional) Você pode publicar uma versão prévia do esquema personalizado, que pode ser adicionada ao catálogo de esquemas do espaço. Navegue até o link `Enable version [preview version number] at:` para publicar uma versão prévia em seu espaço.

Você pode emular a criação do projeto sem precisar criar um projeto no CodeCatalyst. Use o comando a seguir para sintetizar seu projeto:

```
yarn blueprint:synth
```

Um esquema é gerado na pasta `synth/synth.[options-name]/proposed-bundle/`. Para obter mais informações, consulte [Síntese](#).

Se você estiver atualizando o esquema personalizado, use o comando a seguir para resintetizar seu projeto:

```
yarn blueprint:resynth
```

Um esquema é gerado na pasta `synth/synth`. `[options-name]/proposed-bundle/`. Para obter mais informações, consulte [Ressíntese](#).

Depois de publicar a versão prévia, você pode adicionar o esquema para que os membros do espaço possam usá-lo para criar novos projetos ou adicionar projetos existentes. Para obter mais informações, consulte [Adicionar um esquema personalizado a um catálogo de esquemas de espaço](#).

## Visualização e publicação de uma versão normal de um esquema personalizado

Quando terminar de desenvolver e visualizar o esquema personalizado, você poderá visualizar e publicar a nova versão que deseja adicionar ao catálogo de esquemas do seu espaço. O fluxo de trabalho de lançamento gerado ao criar um projeto publica automaticamente as alterações que são enviadas por push. Se você desativou a geração do fluxo de trabalho ao criar o esquema, seu esquema não será automaticamente disponibilizado para ser adicionado ao catálogo de esquemas do espaço. Você ainda pode publicar o esquema personalizado no espaço depois de executar um comando `yarn`.

Para visualizar e publicar um esquema personalizado

1. Retome o Ambiente de Desenvolvimento. Para obter mais informações, consulte . [Retomar um Ambiente de Desenvolvimento](#)
2. Abra um terminal funcional no Ambiente de Desenvolvimento.
3. • Se você optou por não gerar o fluxo de trabalho de lançamento ao criar o esquema, use o seguinte comando:

```
yarn blueprint:release
```

Você ainda pode navegar até o link `See this blueprint at:` fornecido para ver seu esquema personalizado.

Publique a versão atualizada do esquema personalizado, que pode ser adicionada ao catálogo de esquemas do espaço. Navegue até o link `Enable version [release version number] at:` para publicar a versão mais recente em seu espaço.

- Se você optou por um fluxo de trabalho de lançamento ao criar seu esquema, a versão mais recente do esquema será publicada automaticamente quando as alterações forem enviadas. Use os seguintes comandos:

```
git add .
```

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

```
git push
```

Depois de publicar a versão normal, você pode adicionar o esquema para que os membros do espaço possam usá-lo para criar novos projetos ou adicionar projetos existentes. Para obter mais informações, consulte [Adicionar um esquema personalizado a um catálogo de esquemas de espaço](#).

## Publicação e adição de um esquema personalizado em espaços e projetos especificados

Por padrão, os comandos `blueprint:preview` e `blueprint:release` são publicados no espaço do CodeCatalyst em que você criou o esquema. Se você tiver vários espaços corporativos, também poderá visualizar e publicar o mesmo esquema nesses espaços. Você também pode adicionar um esquema a um projeto existente de outro espaço.

Para publicar ou adicionar um esquema personalizado em um espaço especificado

1. Retome o Ambiente de Desenvolvimento. Para obter mais informações, consulte [Retomar um Ambiente de Desenvolvimento](#).
2. Abra um terminal funcional no Ambiente de Desenvolvimento.
3. (Opcional) Instale as dependências necessárias do projeto, caso ainda não as tenha instalado. Use o seguinte comando:

```
yarn
```

4. Use a tag `--space` para publicar uma versão prévia ou normal em um espaço especificado. Por exemplo:

- ```
yarn blueprint:preview --space my-awesome-space # publishes under a "preview" version tag to 'my-awesome-space'
```

Resultado do exemplo:

```
Enable version 0.0.1-preview.0 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
```

```
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.0/projects/create
```

- ```
yarn blueprint:release --space my-awesome-space # publishes normal version to 'my-awesome-space'
```

### Resultado do exemplo:

```
Enable version 0.0.1 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
```

```
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1/projects/create
```

Use o `--project` para adicionar uma versão prévia de um esquema personalizado a um projeto existente em um espaço especificado. Por exemplo:

```
yarn blueprint:preview --space my-awesome-space --project my-project # previews blueprint application to an existing project
```

### Resultado do exemplo:

```
Enable version 0.0.1-preview.1 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
```

```
Blueprint applied to [my-project]: https://codecatalyst.aws/spaces/my-awesome-space/projects/my-project/blueprints/%40amazon-codecatalyst%2FmySpace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.1/add
```

## Definição de permissões de publicação para um esquema personalizado

Por padrão, a permissão de um esquema personalizado será habilitada se uma versão do fluxo de trabalho for gerada durante a criação do projeto. Quando as permissões de publicação estão habilitadas, o esquema pode ser publicado no espaço. Você pode desabilitar a permissão para que o esquema não possa ser publicado. Quando a permissão é desabilitada, o fluxo de trabalho

de lançamento gerado durante a criação do esquema não é executado. Novas alterações em um esquema não podem ser publicadas, a menos que as permissões do esquema estejam habilitadas.

#### Important

Para habilitar ou desabilitar as permissões de publicação de um projeto de esquema personalizado, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

Para definir as permissões de publicação de um projeto de esquema

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço em que você deseja gerenciar as permissões de publicação de um esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Escolha a guia Permissões de publicação do projeto para ver as permissões de publicação de todos os esquemas do espaço.
5. Escolha o esquema que você deseja gerenciar e, depois, selecione Habilitar ou Desabilitar para alterar as permissões de publicação. Se você estiver habilitando as permissões, revise os detalhes da alteração de permissão e selecione Habilitar publicação de esquema para confirmar a alteração.

## Adicionar um esquema personalizado a um catálogo de esquemas de espaço

Depois que você publicar um esquema personalizado no espaço, poderá adicioná-lo ao catálogo de esquemas do espaço. Se você adicionar um esquema personalizado ao catálogo do espaço do CodeCatalyst, o esquema estará disponível para todos os membros do espaço usarem ao criar um projeto ou adicioná-lo a um projeto existente. Antes de adicionar um esquema personalizado ao catálogo de esquemas do espaço, a permissão de publicação do esquema deve estar habilitada. Se você optou pela geração da versão do fluxo de trabalho, as permissões de publicação estão habilitadas por padrão. Para obter mais informações, consulte [Definição de permissões de publicação para um esquema personalizado](#) e [Publicação de um esquema personalizado em um espaço](#).

 Important

Para adicionar um esquema personalizado ao catálogo de esquemas do espaço do CodeCatalyst, você deve se conectar com uma conta que tenha o perfil Administrador do espaço ou Usuário avançado no espaço.

## Como adicionar um esquema ao catálogo de esquemas do espaço

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. O esquema só pode ser adicionado pela ramificação padrão do repositório de origem. Se você desenvolveu o esquema em uma ramificação de recurso, mescle a ramificação de recurso com as alterações na ramificação padrão. Crie uma solicitação pull para mesclar as alterações na ramificação padrão. Para obter mais informações, consulte [Análise de código com pull requests na Amazon CodeCatalyst](#).
3. No console do CodeCatalyst, acesse o painel do espaço com o esquema personalizado.
4. No painel do espaço, escolha a guia Configurações e Esquemas.
5. Escolha o nome do esquema que você deseja adicionar e, depois, selecione Adicionar ao catálogo. Se você tiver mais de uma versão, escolha uma delas no menu suspenso Versão do catálogo.
6. Escolha Salvar.

## Alterar as versões do catálogo para um esquema personalizado

Como autor do esquema, você pode gerenciar a versão que deseja publicar no catálogo de esquemas do espaço. Alterar a versão do catálogo de um esquema não afeta os projetos que estão usando uma versão diferente do esquema.

 Important

Para alterar as versões de catálogo de um esquema personalizado no catálogo de esquemas do espaço do Amazon CodeCatalyst, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

## Como gerenciar uma versão do esquema personalizado

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço em que você deseja alterar uma versão do esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Na tabela Esquemas do espaço, selecione o botão de opção do esquema personalizado que você deseja gerenciar.
5. Selecione Criar versão do catálogo e, depois, escolha a versão no menu suspenso Versão do catálogo.
6. Escolha Salvar.

## Visualização de detalhes, versões e projetos de um esquema personalizado

Você pode visualizar os esquemas personalizados publicados do seu espaço, incluindo os detalhes, as versões e os projetos criados com o esquema ou adicionados ao esquema.

### Tópicos

- [Visualização dos esquemas personalizados de um espaço](#)
- [Visualização de projetos criados com ou adicionados esquemas personalizados](#)

## Visualização dos esquemas personalizados de um espaço

Para visualizar os esquemas personalizados de um espaço

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço onde você deseja visualizar um esquema personalizado.
3. No painel do espaço, selecione a guia Configurações e selecione Esquemas para visualizar os Esquemas do espaço. Os detalhes a seguir são exibidos na tabela:
  - Nome – Nome do esquema personalizado.
  - Status do catálogo – Se o esquema personalizado é publicado no catálogo de esquemas do espaço.
  - Versão mais recente – A versão mais recente do esquema personalizado.
  - Última modificação – A data em que o esquema do espaço foi atualizado pela última vez.



## Visualização de projetos criados com ou adicionados esquemas personalizados

Para visualizar projetos criados com ou adicionados esquemas personalizados

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço onde você deseja visualizar um esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Na tabela de Esquemas do espaço, selecione o nome de um esquema personalizado para visualizar as tabelas de Projetos quem usam esquema e Projetos que não usam esquema.

## Remoção de um esquema personalizado de um catálogo de esquemas de espaço

Um esquema personalizado pode ser removido do catálogo de esquemas do seu espaço se você não quiser mais que ele seja usado para criar novos projetos ou aplicado a projetos existentes.

### Note

Se você remover um esquema personalizado do catálogo de esquemas de um espaço, isso não afetará um projeto criado a partir do esquema ou um projeto que aplicou o esquema. Os recursos do esquema não são removidos do projeto.

### Important

Para remover um esquema personalizado do catálogo de esquemas do seu espaço do CodeCatalyst, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

Para remover um esquema personalizado de um catálogo de esquemas de espaço

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o painel do espaço com seu esquema personalizado
3. No painel do espaço, escolha a guia Configurações e Esquemas.

4. Escolha o nome do esquema que você deseja remover e, depois, escolha Remover esquema do catálogo.

## Exclusão de um esquema ou uma versão personalizada publicada

Quando você exclui a versão de um esquema personalizado ou o próprio esquema do espaço do Amazon CodeCatalyst, todo o seu acesso é removido dos recursos do projeto ou da versão do esquema. Quando você excluir uma versão do esquema ou o esquema, os membros do projeto não conseguirão acessar os recursos do projeto e todos os fluxos de trabalho solicitados por repositórios de origem de terceiros serão interrompidos.

### Note

Se você excluir um esquema, isso não afetará um projeto ao qual o esquema foi aplicado. Os recursos do esquema não são removidos do projeto.

Se uma versão do esquema for publicada no catálogo de esquemas do espaço, escolha uma nova versão para o catálogo antes de excluir a versão publicada.

### Important

Para excluir um esquema personalizado publicado ou a versão do catálogo de um esquema personalizado do seu espaço, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço ou Usuário avançado no espaço.

Para excluir a versão do catálogo de um esquema personalizado

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço em que você deseja excluir a versão do catálogo de um esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Escolha o nome do esquema com a versão do catálogo que deseja excluir.
5. Escolha o botão de opção da versão do catálogo que deseja excluir e, depois, selecione Excluir versão.

6. Revise os detalhes e, depois, escolha outra versão do esquema no menu suspenso Escolher uma nova versão do catálogo de esquemas.
7. Insira `delete` para confirmar a exclusão da versão do catálogo de esquemas.
8. Escolha Excluir.

Se uma versão do esquema não estiver no catálogo de esquemas do espaço, você poderá excluí-la sem escolher uma nova versão.

Para excluir uma versão do esquema personalizado

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço em que você deseja excluir uma versão do esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Escolha o nome do esquema com a versão que deseja excluir.
5. Escolha o botão de opção da versão que deseja excluir e, depois, selecione Excluir versão.
6. Insira `delete` para confirmar a exclusão da versão do esquema.
7. Escolha Excluir.

A exclusão de um esquema do catálogo de esquemas do espaço exclui todas as versões do esquema. Os projetos do espaço que estão usando o esquema não são afetados pela exclusão.

Para excluir uma versão do esquema personalizado

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. No console do CodeCatalyst, navegue até o espaço em que você deseja excluir um esquema personalizado.
3. No painel do espaço, escolha a guia Configurações e Esquemas.
4. Na tabela Esquemas do espaço, escolha o botão de opção do esquema personalizado que você deseja excluir e, depois, selecione Excluir esquema.
5. Insira `delete` para confirmar a exclusão do esquema personalizado.
6. Escolha Excluir.

## Como lidar com dependências, incompatibilidades e ferramentas

O gerenciamento incorreto de dependências pode levar a falhas de criação e problemas de runtime para aqueles que usam seus esquemas personalizados. Ferramentas e componentes desatualizados podem impedir que os usuários do esquema acessem os recursos e as correções de erros mais recentes. Você pode gerenciar dependências, lidar com incompatibilidades de dependências e atualizar ferramentas e componentes para garantir que todas as dependências usem as mesmas versões de componentes e que os componentes sejam sincronizados.

### Tópicos

- [Adicionar dependências](#)
- [Como lidar com incompatibilidades de tipos de dependência](#)
- [Uso de yarn e npm](#)
- [Atualização de ferramentas e componentes](#)

### Adicionar dependências

Como autor do esquema, você talvez precise adicionar pacotes ao seu esquema, como `@amazon-codecatalyst/blueprint-component.environments`. Você precisa atualizar o arquivo `projen.ts` com esse pacote e, depois, gerar novamente a configuração do seu projeto com o [Projen](#). O Projen atua como modelo de projeto para cada base de código do esquema, o que permite enviar atualizações de ferramentas compatíveis com versões anteriores alterando a forma como o modelo renderiza os arquivos de configuração. O arquivo `package.json` é um arquivo parcialmente pertencente ao modelo do Projen. O Projen reconhece as versões de dependência incluídas no arquivo `package.json`, mas outras opções precisam se originar do modelo.

Para adicionar uma dependência e atualizar um arquivo **`projenrc.ts`**

1. No arquivo `projen.ts`, navegue até a seção `deps`.
2. Adicione a dependência que deseja usar no esquema.
3. Use o comando a seguir para gerar novamente a configuração do projeto:

```
yarn projen && yarn
```

## Como lidar com incompatibilidades de tipos de dependência

Depois de uma atualização do [Yarn](#), você pode receber o seguinte erro em relação a um parâmetro do repositório:

```
Type 'SourceRepository' is missing the following properties from type
'SourceRepository': synthesisSteps, addSynthesisStep
```

O erro ocorre devido a uma incompatibilidade de dependências que acontece quando um componente depende de uma versão mais recente de outro componente, mas o componente dependente é fixado em uma versão mais antiga. O erro pode ser corrigido fazendo com que todos os seus componentes dependam da mesma versão para que a versão seja sincronizada entre eles. É melhor manter todos os pacotes fornecidos pelo esquema na mesma versão mais recente (0.0.x), a menos que você tenha certeza de como está lidando com as versões. O exemplo a seguir mostra como o arquivo `package.json` pode ser configurado para que todas as dependências dependam da mesma versão:

```
...
"@caws-blueprint-component/caws-environments": "^0.1.12345",
"@caws-blueprint-component/caws-source-repositories": "^0.1.12345",
"@caws-blueprint-component/caws-workflows": "^0.1.12345",
"@caws-blueprint-component/caws-workspaces": "^0.1.12345",
"@caws-blueprint-util/blueprint-utils": "^0.1.12345",
...
"@caws-blueprint/blueprints.blueprint": "*",
```

Depois de configurar as versões para todas as dependências, use o seguinte comando:

```
yarn install
```

## Uso de yarn e npm

Os esquemas usam [Yarn](#) para ferramentas. Usar [npm](#) e Yarn causará problemas de ferramentas porque a forma como as árvores de dependência são resolvidas por cada uma é diferente. Para evitar esses problemas, é melhor usar somente o Yarn.

Se você instalou dependências acidentalmente usando o npm, poderá remover o arquivo `package-lock.json` gerado e garantir que seu arquivo `.projenrc.ts` esteja atualizado com as dependências necessárias. Você gera novamente a configuração do projeto com o Projén.

Use o seguinte para gerar novamente a partir do modelo:

```
yarn projen
```

Depois de verificar se o arquivo `.projenrc.ts` está atualizado com as dependências necessárias, use o seguinte comando:

```
yarn
```

## Atualização de ferramentas e componentes

Ocasionalmente, você talvez queira atualizar suas ferramentas e componentes para trazer novos recursos disponíveis. É recomendável manter todos os componentes na mesma versão, a menos que você tenha certeza de como está lidando com as versões. As versões são sincronizadas entre os componentes, portanto, as mesmas versões para todos os componentes garantem a dependência adequada entre eles.

Usar o monorepo do espaço de trabalho do Yarn

Use o seguinte comando para atualizar utilitários e componentes da raiz do repositório de um esquema personalizado:

```
yarn upgrade @amazon-codecatalyst/*
```

Use o comando a seguir se não estiver usando um monorepo:

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Outras opções que você pode usar para atualizar ferramentas e componentes:

- Use a visualização `npm @caws-blueprint-component/<some-component>` para ter a versão mais recente.
- Aumente manualmente para a versão mais recente definindo a versão no arquivo `package.json` e usando o seguinte comando: `yarn`. Todos os componentes e utilitários devem ter a mesma versão.

## Contribuir

O kit de desenvolvimento de software (SDK) de esquemas é uma biblioteca de código aberto com a qual você pode contribuir. Como colaborador, considere as orientações de contribuição, o feedback e os defeitos. Para ter mais informações, consulte o [repositório GitHub de esquemas](#).

## Cotas para plantas em CodeCatalyst

A tabela a seguir descreve as cotas e os limites dos blueprints na Amazon. CodeCatalyst Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

|                                                             |     |
|-------------------------------------------------------------|-----|
| Número máximo de plantas aplicadas por CodeCatalyst projeto | 100 |
|-------------------------------------------------------------|-----|

# Armazene e colabore no código com repositórios de origem no CodeCatalyst

CodeCatalyst repositórios de origem são repositórios Git hospedados na Amazon. CodeCatalyst Você pode usar repositórios de origem CodeCatalyst para armazenar, criar versões e gerenciar ativos de um projeto com segurança.

Os ativos em um CodeCatalyst repositório podem incluir:

- documentos
- código-fonte
- arquivos binários

CodeCatalyst também usa o repositório de origem de um projeto para armazenar informações de configuração do seu projeto, como arquivos de configuração do fluxo de trabalho.

Você pode ter mais de um repositório de origem em um CodeCatalyst projeto. Por exemplo, você pode querer ter repositórios de origem separados para código-fonte front-end, código-fonte back-end, utilitários e documentação.

Aqui está um fluxo de trabalho possível para trabalhar com código em repositórios de origem, pull requests e ambientes de desenvolvimento em: CodeCatalyst

Mary Major cria um projeto de aplicativo web CodeCatalyst usando um blueprint, que cria um repositório de origem com código de amostra nele. Ela convida seus amigos Li Juan, Saanvi Sarkar e Jorge Souza para trabalhar no projeto com ela. Li Juan analisa o código de amostra no repositório de origem e decide fazer algumas mudanças rápidas para adicionar um teste ao código. Li cria um ambiente de desenvolvimento, escolhe AWS Cloud9 como IDE e especifica uma nova ramificação, *test-code*. O Ambiente de Desenvolvimento é aberto. Li adiciona rapidamente o código, depois confirma e envia a ramificação com as alterações no repositório de origem. CodeCatalyst Depois, Li cria uma solicitação pull. Como parte da criação dessa solicitação pull, Li adiciona Jorge Souza e Saanvi Sarkar como revisores para garantir que o código seja revisado.

Ao revisar o código, Jorge Souza lembra que ele tem seu próprio repositório de projetos GitHub que contém um protótipo do aplicativo em que eles estão trabalhando. Ele pede que Mary Major instale e configure a extensão que lhe permitirá vincular o GitHub repositório ao projeto como um repositório



de origem adicional. Mary revisa o repositório GitHub e trabalha com Jorge para configurar a GitHub extensão para que ele possa vincular o GitHub repositório como um repositório de origem adicional para o projeto.

CodeCatalyst os repositórios de origem oferecem suporte à funcionalidade padrão do Git e funcionam com suas ferramentas existentes baseadas no Git. Você pode criar e usar tokens de acesso pessoal (PATs) como uma senha específica do aplicativo ao clonar e trabalhar com repositórios de origem de um cliente Git ou ambientes de desenvolvimento integrados (). IDEs Eles PATs estão associados à sua identidade de CodeCatalyst usuário. Para obter mais informações, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).

CodeCatalyst repositórios de origem oferecem suporte a pull requests. Essa é uma maneira simples de você e outros membros do projeto revisarem e comentarem as alterações no código antes de mesclá-las de uma ramificação para outra. Você pode ver as alterações no CodeCatalyst console e comentar nas linhas de código.

Os envios para ramificações em um repositório de CodeCatalyst origem podem iniciar automaticamente uma execução em um fluxo de trabalho, onde as alterações podem ser criadas, testadas e implantadas. Se seu repositório de origem foi criado como parte de um projeto usando um modelo de projeto, um ou mais fluxos de trabalho serão configurados para você como parte do projeto. Você pode acrescentar fluxos de trabalho adicionais para repositórios a qualquer momento. Os arquivos de configuração YAML para fluxos de trabalho em um projeto são armazenados nos repositórios de origem configurados na ação de origem desses fluxos de trabalho. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#).

## Tópicos

- [Conceitos de repositório de origem](#)
- [Configuração para trabalhar com repositórios de origem](#)
- [Introdução aos repositórios CodeCatalyst de origem e ao blueprint do aplicativo de página única](#)
- [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#)
- [Organizando seu trabalho de código-fonte com filiais na Amazon CodeCatalyst](#)
- [Gerenciando arquivos de código-fonte na Amazon CodeCatalyst](#)
- [Análise de código com pull requests na Amazon CodeCatalyst](#)
- [Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst](#)
- [Cotas para repositórios de origem em CodeCatalyst](#)

# Conceitos de repositório de origem

Aqui estão alguns conceitos que você deve conhecer ao trabalhar com repositórios CodeCatalyst de origem.

## Tópicos

- [Projetos](#)
- [Repositórios de origem](#)
- [Ambientes de Desenvolvimento](#)
- [Tokens de acesso pessoal \(PATs\)](#)
- [Ramificações](#)
- [Ramificações padrão](#)
- [Confirmações](#)
- [Solicitações pull](#)
- [Revisões](#)
- [Fluxos de trabalho](#)

## Projetos

Um projeto representa um esforço colaborativo CodeCatalyst que apoia equipes e tarefas de desenvolvimento. Depois que tiver um projeto, você poderá adicionar, atualizar ou remover usuários e recursos, personalizar o painel do projeto e monitorar o andamento do trabalho da equipe. Você pode ter vários projetos em um espaço.

Os repositórios de origem são específicos para os projetos onde você os cria ou os vincula em um espaço. Não é possível compartilhar um repositório entre projetos e vincular um repositório a mais de um projeto em um espaço. Os usuários com o perfil de Colaborador ou Administrador do projeto em um projeto podem interagir com os repositórios de origem associados a esse projeto de acordo com as permissões concedidas a esses perfis. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).

## Repositórios de origem

Em um repositório de origem, você armazena com segurança códigos e arquivos do seu projeto. Ele também armazena o histórico de versões dos seus arquivos. Por padrão, um repositório de origem é compartilhado com os outros usuários em seu CodeCatalyst projeto. É possível ter mais

de um repositório de origem para um projeto. Você pode criar repositórios de origem para projetos em CodeCatalyst, ou você pode optar por vincular um repositório de origem existente hospedado por outro serviço, se esse serviço for suportado por uma extensão instalada. Por exemplo, você pode vincular um GitHub repositório a um projeto depois de instalar a extensão GitHub Repositories. Para obter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#) e [Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst](#).

## Ambientes de Desenvolvimento

Um ambiente de desenvolvimento é um ambiente de desenvolvimento baseado em nuvem que você pode usar CodeCatalyst para trabalhar rapidamente no código armazenado nos repositórios de origem do seu projeto. As ferramentas do projeto e as bibliotecas de aplicações incluídas no Ambiente de Desenvolvimento são definidas por um arquivo de desenvolvimento no repositório de origem do projeto. Se você não tiver um arquivo de desenvolvimento no repositório de origem, um arquivo padrão será aplicado automaticamente. O arquivo de desenvolvimento padrão inclui ferramentas para as linguagens e estruturas de programação usadas com mais frequência. Por padrão, um Ambiente de Desenvolvimento é configurado para ter um processador de 2 núcleos, 4 GB de RAM e 16 GiB de armazenamento persistente.

Você pode optar por clonar uma ramificação existente do seu repositório de origem em seu Ambiente de Desenvolvimento ou criar uma ramificação como parte da criação do Ambiente de Desenvolvimento.

## Tokens de acesso pessoal (PATs)

Um token de acesso pessoal (PAT) é semelhante a uma senha. Ele está associado à sua identidade de usuário para uso em todos os espaços e projetos em CodeCatalyst. Você usa PATs para acessar CodeCatalyst recursos que incluem ambientes de desenvolvimento integrados (IDEs) e repositórios de origem baseados em Git. PATs representam você CodeCatalyst e você pode gerenciá-los em suas configurações de usuário. Um usuário pode ter mais de um PAT. Os tokens de acesso pessoal são exibidos apenas uma vez. Como prática recomendada, armazene-os com segurança em seu computador local. Por padrão, PATs expira após um ano.

Ao trabalhar com ambientes de desenvolvimento integrados (IDEs), PATs são equivalentes a uma senha do Git. Forneça o PAT quando for solicitada uma senha ao configurar seu IDE para trabalhar com um repositório Git. Para ter mais informações sobre como conectar o IDE a um repositório baseado em Git, consulte a documentação do IDE.

## Ramificações

Uma ramificação é um ponteiro ou referência para um commit no Git e no CodeCatalyst. Você pode usar ramificações para organizar seu trabalho. Por exemplo, você pode usar ramificações para trabalhar em uma versão nova ou diferente de arquivos sem afetar o trabalho em outras ramificações. Elas também podem ser usadas para desenvolver novos recursos, armazenar uma versão específica do seu projeto e muito mais. Um repositório de origem pode ter uma ramificação ou várias ramificações. Quando você cria um projeto usando um modelo, o repositório de origem criado para o projeto contém arquivos de amostra em uma ramificação chamada principal. Essa ramificação principal é a ramificação padrão do seu repositório.

### Ramificações padrão

Os repositórios de origem CodeCatalyst têm uma ramificação padrão, independentemente de como você os cria. Se você optar por criar um projeto usando um modelo, o repositório de origem criado para esse projeto incluirá um arquivo README.md, além de código de exemplo, definições de fluxo de trabalho e outros recursos. Se você criar um repositório de origem sem usar um modelo, um arquivo README.md será adicionado como uma primeira confirmação e uma ramificação padrão será criada como parte da criação do repositório. Essa ramificação padrão é chamada principal. Essa ramificação padrão é aquela usada como base ou ramificação padrão em repositórios locais quando os usuários clonam o repositório. Você pode alterar qual ramificação é usada como ramificação padrão. Para obter mais informações, consulte [Gerenciar a ramificação padrão de um repositório](#).

Você não pode excluir a ramificação padrão de um repositório de origem. Os resultados da pesquisa incluem somente os resultados da ramificação padrão.

### Confirmações

Uma confirmação é uma alteração em um arquivo ou conjunto de arquivos. No CodeCatalyst console da Amazon, um commit salva suas alterações e as envia para um repositório de origem. A confirmação inclui informações sobre a alteração, incluindo a identidade do usuário que fez a alteração, a hora e a data da alteração, o título da confirmação e qualquer mensagem incluída sobre a alteração. Para obter mais informações, consulte [Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst](#).

No contexto de um repositório de origem em CodeCatalyst, os commits são instantâneos do conteúdo e das alterações no conteúdo do seu repositório. Você também pode adicionar tags Git a confirmações para identificar confirmações específicas.

## Solicitações pull

Uma solicitação pull é a principal maneira pela qual você e outros usuários revisam, comentam e mesclam alterações de código de uma ramificação para outra em um repositório de origem. Você pode usar solicitações pull para revisar alterações de código de maneira colaborativa em caso de alterações ou correções relativamente pequenas, adições significativas de recursos ou novas versões do seu software lançado. Em uma solicitação pull, você pode revisar as alterações entre as ramificações de origem e de destino ou as diferenças entre as revisões dessas ramificações. Você pode adicionar comentários às alterações de linhas de código individuais, bem como comentários sobre a solicitação pull em geral.

### Tip

Enquanto você cria uma solicitação pull, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a solicitação pull for criada, a diferença exibida será entre a revisão da solicitação pull escolhida e a confirmação que foi a ponta da ramificação de destino quando você criou a solicitação pull. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

## Revisões

Revisão é uma versão atualizada de uma solicitação pull. Cada envio para a ramificação de origem de uma solicitação pull cria uma revisão que contém as alterações feitas nas confirmações incluídas nesse envio. Você pode visualizar as diferenças entre as revisões de uma solicitação pull, além das diferenças entre as ramificações de origem e destino. Para obter mais informações, consulte [Análise de código com pull requests na Amazon CodeCatalyst](#).

## Fluxos de trabalho

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

 Tip

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Um repositório de origem também pode armazenar os arquivos de configuração e outras informações para fluxos de trabalho, notificações, problemas e outras informações de configuração do projeto. Os arquivos de configuração são criados e armazenados no repositório de origem quando você cria recursos que exigem arquivos de configuração ou quando você especifica o repositório como uma ação de origem para um fluxo de trabalho. Se você criar um projeto a partir de um esquema, terá arquivos de configuração já armazenados no repositório de origem criado para você como parte do projeto. Essas informações de configuração são armazenadas em uma pasta chamada `.codecatalyst` na ramificação padrão do seu repositório. Sempre que você cria uma ramificação da ramificação padrão, você cria uma cópia dessa pasta e de sua configuração, além de todos os outros arquivos e pastas dessa ramificação.

## Configuração para trabalhar com repositórios de origem

Ao trabalhar com repositórios de origem na Amazon CodeCatalyst em sua máquina local, você pode usar o Git sozinho ou em um ambiente de desenvolvimento integrado (IDE) compatível para fazer alterações no código e enviar e extrair seu código. Como prática recomendada, use as versões mais recentes do Git e de outros softwares.

 Note

Se você usa Ambientes de Desenvolvimento, não precisa instalar o Git. Uma versão recente do Git está incluída no Ambiente de Desenvolvimento.

Informações de compatibilidade de versão para CodeCatalyst

| Componente | Versão       |
|------------|--------------|
| Git        | mais recente |

## Instale o Git

Para trabalhar com arquivos, confirmações, ramificações e outras informações nos repositórios de origem de um cliente Git sem um IDE, instale o Git na sua máquina local.

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

## Criar um token de acesso pessoal

Para clonar repositórios de origem em sua máquina local ou em seu IDE preferido, crie um token de acesso pessoal (PAT).

Para criar um token de acesso pessoal (PAT)

1. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações.

### Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

2. Em Nome do PAT, insira um nome descritivo para o seu PAT.
3. Em Data de expiração, mantenha a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de vencimento é padronizada para um ano a contar da data atual.
4. Escolha Criar.

Você também pode criar esse token ao escolher Clonar repositório para um repositório de origem.

5. Salve o segredo do PAT em um local seguro.

### Important

O segredo do PAT só é exibido uma vez. Não é possível recuperá-lo depois de fechar a janela.

# Introdução aos repositórios CodeCatalyst de origem e ao blueprint do aplicativo de página única

Siga as etapas deste tutorial para aprender a trabalhar com repositórios de origem na Amazon CodeCatalyst.

A maneira mais rápida de começar a trabalhar com repositórios de origem na Amazon CodeCatalyst é criar um projeto usando um modelo. Quando você cria um projeto usando um modelo, recursos são criados para você, incluindo um repositório de origem que inclui código de amostra. Você pode usar esse exemplo de repositório e código para saber como:

- Visualizar os repositórios de origem de um projeto e navegar por seus conteúdos
- Criar um Ambiente de Desenvolvimento com uma nova ramificação onde você pode trabalhar no código
- Alterar um arquivo, confirmar e enviar suas alterações
- Criar uma solicitação pull e revisar suas alterações de código com outros membros do projeto
- Visualizar o fluxo de trabalho do seu projeto, compilar e testar automaticamente as alterações na ramificação de origem da solicitação pull
- Mesclar suas alterações da ramificação de origem na ramificação de destino e fechar a solicitação pull
- Ver as alterações mescladas compiladas e implantadas automaticamente

Para aproveitar ao máximo este tutorial, convide outras pessoas para o seu projeto para que vocês possam trabalhar juntos em uma solicitação pull. Você também pode explorar recursos adicionais CodeCatalyst, como criar problemas e associá-los a uma pull request ou configurar notificações e receber alertas quando o fluxo de trabalho associado for executado. Para uma exploração completa de CodeCatalyst, consulte [Tutoriais de conceitos básicos](#).

## Criar um projeto com um esquema

Criar um projeto é o primeiro passo para poder trabalhar em conjunto. Você pode usar um esquema para criar seu projeto, que também criará um repositório de origem com código de amostra e um fluxo de trabalho que compilará e implantará automaticamente seu código quando você o alterar. Neste tutorial, mostraremos um projeto criado com o esquema aplicação de página única, mas você pode seguir os procedimentos de qualquer projeto com um repositório de origem. Selecione ou adicione um perfil do IAM, se você não tiver um como parte da criação do projeto. Recomendamos



que você use o perfil de serviço CodeCatalystWorkflowDevelopmentRole-**spaceName** para esse projeto.

Se você já tem um projeto, siga para [Visualizar os repositórios de um projeto](#).

 Note

Somente usuários com a função de administrador do Space ou usuário avançado podem criar projetos no CodeCatalyst. Se você não tem esse perfil e precisa de um projeto para trabalhar neste tutorial, peça a alguém com um desses perfis que crie um projeto para você e adicione você ao projeto criado. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).

Como criar um projeto com um esquema

1. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.
2. No painel do espaço, escolha Criar projeto.
3. Selecione Começar com um esquema.

 Tip

Você pode optar por adicionar um esquema fornecendo ao Amazon Q os requisitos do projeto para que o Amazon Q sugira um esquema. Para obter mais informações, consulte [Usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar funcionalidade](#) e [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#). Esse recurso está disponível somente na região do Oeste dos EUA (Oregon).

Essa funcionalidade exige que recursos de IA generativa estejam habilitados para o espaço. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

4. Na guia CodeCatalyst Blueprints ou Space blueprints, escolha um blueprint e, em seguida, escolha Avançar.
5. Em Nomear seu projeto, insira o nome que você deseja atribuir ao seu projeto e os nomes dos recursos associados. O nome deve ser exclusivo em seu espaço.
6. (Opcional) Por padrão, o código-fonte criado pelo blueprint é armazenado em um CodeCatalyst repositório. Também é possível armazenar o código-fonte do esquema em um repositório

de terceiros. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você deseja usar:

- GitHub repositórios: Conecte uma GitHub conta.

Escolha o menu suspenso Avançado, escolha GitHub como provedor do repositório e, em seguida, escolha a GitHub conta na qual você deseja armazenar o código-fonte criado pelo blueprint.

 Note

Se você estiver conectando uma GitHub conta, deverá criar uma conexão pessoal para estabelecer um mapeamento de identidade entre sua CodeCatalyst identidade e sua GitHub identidade. Para obter mais informações, consulte [Conexões pessoais e Acessando GitHub recursos com conexões pessoais](#).

- Repositórios do Bitbucket: conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, selecione Bitbucket como provedor do repositório e, depois, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo esquema.

- GitLab repositórios: Conecte um GitLab usuário.

Escolha o menu suspenso Avançado, escolha GitLab como provedor do repositório e, em seguida, escolha o GitLab usuário no qual você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Recursos do projeto, configure os parâmetros do esquema. Dependendo do esquema, você pode ter a opção de nomear o repositório de origem.
8. (Opcional) Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, selecione Visualizar código ou Visualizar fluxo de trabalho em Gerar visualização do projeto.
9. (Opcional) Selecione Visualizar detalhes no cartão do esquema para visualizar detalhes específicos sobre o esquema, como uma visão geral da arquitetura do esquema, conexões e permissões necessárias e o tipo de recursos que o esquema cria.
10. Escolha Criar projeto.

A página de visão geral do projeto é aberta assim que você cria um projeto ou aceita um convite para um projeto e conclui o processo de login. A página de visão geral do projeto de um novo projeto não contém problemas em aberto nem solicitações pull. Você pode optar por criar um problema e atribuí-lo a si mesmo. Você também pode optar por convidar outra pessoa para o seu projeto. Para obter mais informações, consulte [Criando um problema no CodeCatalyst](#) e [Convite de um usuário para um projeto](#).

## Visualizar os repositórios de um projeto

Como membro de um projeto, você pode visualizar os repositórios de origem do projeto. Você também pode criar repositórios adicionais. Se alguém com a função de administrador do Space tiver instalado e configurado os GitHub repositórios, os repositórios do Bitbucket ou a extensão dos GitLab repositórios, você também poderá adicionar links para repositórios de terceiros nas GitHub contas, nos espaços de trabalho do Bitbucket ou nos usuários configurados para a extensão. GitLab Para obter mais informações, consulte [Criar um repositório de origem](#) e [Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst](#).

### Note

Para projetos criados com o blueprint de aplicativo de página única, o nome padrão do repositório de origem que contém o código de amostra é. ***spa-app***

## Como navegar até os repositórios de origem de um projeto

### 1. Navegue até o projeto e faça o seguinte:

- Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório.
- No painel de navegação, selecione Código e, depois, selecione Repositórios de origem. Em Repositórios de origem, selecione o nome do repositório na lista. Você pode filtrar a lista de repositórios digitando parte do nome do repositório na barra de filtro.

### 2. Na página inicial do repositório, visualize o conteúdo do repositório e as informações sobre os recursos associados, como o número de solicitações pull e fluxos de trabalho. Por padrão, o conteúdo da ramificação padrão é exibido. Você pode alterar a visualização selecionando uma ramificação diferente na lista suspensa.

A página de visão geral do repositório inclui informações sobre os fluxos de trabalho e solicitações pull configuradas para as ramificações desse repositório e seus arquivos. Se você acabou de criar o projeto, os fluxos de trabalho iniciais para compilar, testar e implantar o código ainda estarão em execução, pois demoram alguns minutos para serem concluídos. Você pode visualizar os fluxos de trabalho relacionados e seus status selecionando o número abaixo de Fluxos de trabalho relacionados, mas isso abre a página Fluxos de trabalho em CI/CD. Para este tutorial, permaneça na página de visão geral e explore o código no repositório. O conteúdo do arquivo `README.md` é renderizado nesta página abaixo dos arquivos do repositório. Em Arquivos, o conteúdo da ramificação padrão é exibido. Você pode alterar a visualização do arquivo para mostrar o conteúdo de outra ramificação, se você tiver uma. A pasta `.codecatalyst` contém o código usado para outras partes do projeto, como arquivos YAML do fluxo de trabalho.

Para visualizar o conteúdo das pastas, selecione a seta próxima ao nome da pasta para expandi-la. Por exemplo, selecione a seta ao lado de `src` para visualizar os arquivos da aplicação web de página única contida nessa pasta. Como visualizar o conteúdo de um arquivo, escolha-o na lista. A opção Visualizar arquivos será aberta, onde você pode navegar pelo conteúdo de vários arquivos. Você também pode editar arquivos únicos no console, mas, para editar vários arquivos, crie um Ambiente de Desenvolvimento.

## Criar um Ambiente de Desenvolvimento

Você pode adicionar e alterar arquivos em um repositório de origem no CodeCatalyst console da Amazon. No entanto, para trabalhar de forma eficaz com vários arquivos e ramificações, recomendamos usar um Ambiente de Desenvolvimento ou clonar o repositório em seu computador

local. Neste tutorial, criaremos um ambiente de AWS Cloud9 desenvolvimento com uma ramificação chamada **develop**. Você pode escolher um nome de ramificação diferente, mas ao nomear a ramificação **develop**, um fluxo de trabalho será executado automaticamente para compilar e testar seu código quando você criar uma solicitação pull posteriormente neste tutorial.

 Tip

Se você decidir clonar um repositório localmente em vez de ou além de usar um Ambiente de Desenvolvimento, assegure-se de ter o Git no computador local ou que seu IDE inclua o Git. Para obter mais informações, consulte [Configuração para trabalhar com repositórios de origem](#).

Para criar um Ambiente de Desenvolvimento com uma nova ramificação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja criar um Ambiente de Desenvolvimento.
3. Selecione o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, Repositórios de origem e selecione o repositório para o qual você deseja criar um Ambiente de Desenvolvimento.
4. Na página inicial do repositório, selecione Criar um ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Escolha o repositório a ser clonado, selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.
7. Opcionalmente, adicione um alias para o Ambiente de Desenvolvimento.
8. Opcionalmente, escolha o botão de edição da configuração do Ambiente de Desenvolvimento para editar a configuração de computação, armazenamento ou tempo limite do Ambiente de Desenvolvimento.
9. Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento. Uma nova guia será aberta com seu ambiente de desenvolvimento no IDE de sua escolha. Você pode editar o código, confirmar e enviar suas alterações.

Depois de criar o Ambiente de Desenvolvimento, você pode editar arquivos, confirmar e enviar suas alterações para a ramificação **test**. Para este tutorial, edite o conteúdo entre as tags `<p>` no arquivo `App.tsx` na pasta `src` para alterar o texto exibido na página da Web. Comprometa-se e promova sua alteração e, em seguida, retorne à CodeCatalyst guia.

Para fazer e promover uma mudança a partir de um ambiente de AWS Cloud9 desenvolvimento

1. Em AWS Cloud9, expanda o menu de navegação lateral para navegar pelos arquivos. Expanda `src` e abra `App.tsx`.
2. Faça uma alteração no texto dentro das tags `<p>`.
3. Salve o arquivo e, depois, confirme e envie suas alterações usando o menu Git. Como alternativa, na janela do terminal, confirme e envie suas alterações com os comandos `git commit` e `git push`.

```
git commit -am "Making an example change"
git push
```

#### Tip

Talvez seja necessário alterar os diretórios no terminal para o diretório do repositório Git antes de executar com êxito os comandos do Git.

## Criar uma solicitação pull

Você pode usar solicitações pull para revisar alterações de código de maneira colaborativa em caso de alterações ou correções relativamente pequenas, adições significativas de recursos ou novas versões do seu software lançado. Neste tutorial, você criará uma pull request para revisar as alterações feitas na **test** ramificação em comparação com a ramificação principal. Criar uma solicitação pull em um projeto criado com um modelo também iniciará a execução dos fluxos de trabalho associados, se houver.

Para criar uma solicitação pull

1. Navegue até o projeto.
2. Execute um destes procedimentos:

- No painel de navegação, escolha Código, Solicitações pull e, depois, selecione Criar solicitação pull.
  - Na página inicial do repositório, escolha Mais e, depois, selecione Criar solicitação pull.
  - Na página do projeto, selecione Criar solicitação pull.
3. No Repositório de origem, certifique-se de que o repositório de origem especificado seja aquele que contém o código confirmado. Essa opção só aparece se você não criou a solicitação pull na página principal do repositório.
  4. Em Ramificação de destino, selecione a ramificação na qual mesclar o código depois que ele for revisado.
  5. Em Ramificação de origem, selecione a ramificação que contém o código confirmado.
  6. Em Título da solicitação pull, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
  7. (Opcional) Na Descrição da solicitação pull, forneça informações, como um link para problemas ou uma descrição das alterações.

 Tip

Você pode escolher Escrever descrição para mim para gerar CodeCatalyst automaticamente uma descrição das alterações contidas na pull request. Você pode fazer alterações na descrição gerada automaticamente depois de adicioná-la à solicitação pull.

Essa funcionalidade exige que os recursos de IA generativa estejam habilitados para o espaço e não estejam disponíveis para solicitações pull em repositórios vinculados. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

8. (Opcional) Em Problemas, selecione Vincular problemas e, depois, selecione um problema na lista ou insira seu ID. Para desvincular um problema, selecione o ícone de desvinculação.
9. (Opcional) Em Revisores obrigatórios, selecione Adicionar revisores obrigatórios. Selecione na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que a solicitação pull seja mesclada na ramificação de destino.

 Note

Não é possível adicionar um revisor como revisor obrigatório e revisor opcional. Não é possível se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, selecione Adicionar revisores opcionais. Selecione na lista de membros do projeto para adicioná-los. Revisores opcionais não precisam aprovar as alterações como um requisito antes que a solicitação pull seja mesclada na ramificação de destino.
11. Revise as diferenças entre as ramificações. A diferença exibida em uma solicitação pull são as alterações entre a revisão na ramificação de origem e a base de mesclagem, que é a confirmação principal da ramificação de destino no momento em que a solicitação pull foi criada. Se nenhuma alteração for exibida, as ramificações podem ser idênticas ou você pode ter escolhido a mesma ramificação para a origem e o destino.
12. Quando a solicitação pull tiver o código e as alterações que você deseja revisar, selecione Criar.

 Note

Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos, bem como à solicitação pull geral. Você pode adicionar links para recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

Você pode visualizar informações sobre fluxos de trabalho associados iniciados pela criação dessa solicitação pull selecionando Visão geral e, depois, revisando as informações na área Detalhes da solicitação pull em Execuções do fluxo de trabalho. Para visualizar a execução do fluxo de trabalho, selecione a execução.

 Tip

Se você nomeou sua ramificação de forma diferente de **develop**, um fluxo de trabalho não será executado automaticamente para compilar e testar suas alterações. Se você quiser configurar isso, edite o arquivo YAML para o onPullRequestBuildAndTestfluxo de trabalho. Para obter mais informações, consulte [Criação de um fluxo de trabalho](#).

Você pode comentar sobre essa solicitação pull e pedir que outros membros do projeto comentem sobre ela. Você também pode optar por adicionar ou alterar revisores opcionais ou obrigatórios. Você pode optar por fazer mais alterações na ramificação de origem do repositório e ver como essas alterações confirmadas criam revisões para a solicitação pull. Para ter mais informações, consulte



[Revisão de uma solicitação pull](#), [Atualizar uma solicitação pull](#), [Análise de código com pull requests na Amazon CodeCatalyst](#) e [Visualização do status e detalhes de execução do fluxo de trabalho](#).

## Mesclar uma solicitação pull

Depois que uma pull request for revisada e receber as aprovações dos revisores necessários, você poderá mesclar sua ramificação de origem com a ramificação de destino no console. CodeCatalyst A mesclagem de uma solicitação pull também iniciará a execução das alterações por meio de qualquer fluxo de trabalho associado à ramificação de destino. Neste tutorial, você mesclará a ramificação de teste com a principal, o que iniciará a execução do onPushToMainDeployPipelinefluxo de trabalho.

Como mesclar uma solicitação pull (console)

1. Em Solicitações pull, escolha a solicitação pull que você criou na etapa anterior. Na solicitação pull, selecione Mesclar.
2. Selecione entre as estratégias de mesclagem disponíveis para a solicitação pull. Também é possível selecionar ou desmarcar a opção de excluir a ramificação de origem depois de mesclar a solicitação pull e, depois, selecionar Mesclar. Após a conclusão da mesclagem, o status da solicitação pull muda para Mesclado e não aparece mais na visualização padrão das solicitações pull. A visualização padrão mostra solicitações pull com o status Aberto. Você ainda pode visualizar uma solicitação pull mesclada, mas não pode aprová-la ou alterar seu status.

### Note

Se o botão Mesclar não estiver ativo ou você ver o rótulo Não mesclável, um revisor obrigatório ainda não aprovou a pull request ou a pull request não pode ser mesclada no console. CodeCatalyst Um revisor que não aprovou uma solicitação pull é indicado por um ícone de relógio em Visão geral na área Detalhes da solicitação pull. Se todos os revisores obrigatórios aprovaram a solicitação pull, mas o botão Mesclar ainda não estiver ativo, você pode ter um conflito de mesclagem ou ter excedido a cota de armazenamento do espaço. Você pode resolver conflitos de mesclagem para a ramificação de destino em um ambiente de desenvolvimento, enviar as alterações e, em seguida, mesclar a pull request, ou você pode resolver conflitos e mesclar localmente e, em seguida, enviar a confirmação que contém a mesclagem para. CodeCatalyst Para ter mais informações, consulte [Mescla de uma solicitação pull \(Git\)](#) e a documentação do Git.

## Visualizar o código implantado

Agora, veremos o código implantado originalmente que estava na ramificação padrão e suas alterações mescladas quando elas são compiladas, testadas e implantadas automaticamente. Para fazer isso, você pode retornar à página de visão geral do repositório e selecionar o número ao lado do ícone de fluxos de trabalho relacionados ou, no painel de navegação, selecionar CI/CD e, depois, selecionar Fluxos de trabalho.

### Como visualizar o código implantado

1. Em Fluxos de trabalho, em `onPushToMainDeployPipeline`, expanda Execuções recentes.

#### Note

Esse é o nome padrão do fluxo de trabalho para projetos criados com o esquema de Aplicação de página única.

2. A execução mais recente é aquela iniciada pela confirmação da solicitação pull mesclada na ramificação `main` e provavelmente exibirá o status Em andamento. Selecione uma execução concluída com sucesso na lista para abrir os detalhes dessa execução.
3. Selecione Variáveis. Copie o valor para `AppURL`. Esse é o URL da aplicação web de página única implantada. Abra uma nova guia do navegador e cole o valor para visualizar o código compilado e implantado. Deixe a guia aberta.
4. Retorne à lista de execuções do fluxo de trabalho e aguarde a conclusão da execução mais recente. Quando isso acontecer, retorne à guia aberta para visualizar a aplicação web e atualizar seu navegador. Veja as alterações feitas na sua solicitação pull mesclada.

## Limpar os recursos

Depois de explorar o trabalho com um repositório de origem e uma solicitação pull, talvez você queira remover todos os recursos desnecessários. Não é possível excluir solicitações pull, mas é possível fechá-las. Você pode excluir as ramificações criadas.

Se você não precisar mais do repositório de origem ou do projeto, também poderá excluir esses recursos. Para obter mais informações, consulte [Exclusão de um repositório de origem](#) e [Excluir um projeto](#).

# Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst

Em um repositório de origem, você armazena com segurança códigos e arquivos do seu projeto. Ele também armazena o histórico de origem, desde a primeira confirmação até as alterações mais recentes. Se você escolher um esquema que inclua um repositório de origem, esse repositório também conterá os arquivos de configuração e outras informações para fluxos de trabalho e notificações do projeto. Essas informações de configuração são armazenadas em uma pasta chamada `.codecatalyst`.

Você pode criar um repositório de origem criando um projeto com um blueprint que cria um repositório de origem como parte da criação de um projeto ou criando um repositório de origem em um projeto existente. CodeCatalyst Os usuários do projeto verão automaticamente e poderão usar os repositórios criados para um projeto. Você também pode optar por vincular um repositório Git hospedado no GitHub Bibucket ou GitLab ao seu projeto. Ao fazer isso, os usuários do seu projeto podem visualizar e acessar esse repositório vinculado na lista de repositórios do projeto.

## Note

Antes de vincular o repositório, instale a extensão para o serviço que o hospeda. Você não pode vincular um repositório arquivado. Embora você possa vincular um repositório vazio, você não pode usá-lo CodeCatalyst até inicializá-lo com um commit inicial que crie uma ramificação padrão. Para obter mais informações, consulte [Instalação de uma extensão em um espaço](#).

Por padrão, um repositório de origem é compartilhado com outros membros do seu CodeCatalyst projeto da Amazon. Você pode criar repositórios de origem adicionais para um projeto ou vincular repositórios ao projeto. Todos os membros de um projeto podem visualizar, adicionar, editar e excluir arquivos e pastas nos repositórios de origem do projeto.

Para trabalhar rapidamente no código em um repositório de origem, você pode criar um Ambiente de Desenvolvimento que clona um repositório específico e se ramifica nele, onde pode trabalhar no código no ambiente de desenvolvimento integrado (IDE) escolhido para o Ambiente de Desenvolvimento. Você pode clonar um repositório de origem em seu computador local e extrair e enviar alterações entre seu repositório local e o repositório remoto em. CodeCatalyst Você também

pode trabalhar com repositórios de origem configurando o acesso a eles em seu IDE preferido, desde que esse IDE ofereça suporte ao gerenciamento de credenciais.

Os nomes dos repositórios devem ser exclusivos em um CodeCatalyst projeto.

## Tópicos

- [Criar um repositório de origem](#)
- [Clonar um repositório Git existente em um repositório de origem](#)
- [Vincular um repositório de origem](#)
- [Visualizar um repositório de origem](#)
- [Editar as configurações de um repositório de origem](#)
- [Clonar um repositório de origem](#)
- [Exclusão de um repositório de origem](#)

## Criar um repositório de origem

Quando você cria um projeto usando um blueprint na Amazon CodeCatalyst, CodeCatalyst cria um repositório de origem para você. Esse repositório de origem contém código de amostra, além das informações de configuração dos fluxos de trabalho e outros recursos criados para você. Essa é a forma recomendada de começar a usar repositórios em CodeCatalyst. Você pode optar por criar repositórios para um projeto. Esses repositórios conterão um arquivo README.md que você pode editar ou excluir a qualquer momento. Dependendo das suas escolhas ao criar um repositório de origem, os repositórios também podem conter um arquivo .gitignore.

Se você quiser clonar um repositório Git existente em CodeCatalyst um repositório de origem, considere criar um repositório vazio. Esse repositório ficará indisponível para uso CodeCatalyst até que você adicione conteúdo a ele, o que você pode fazer com alguns comandos simples do Git. Como alternativa, você pode adicionar conteúdo ao repositório vazio diretamente do CodeCatalyst console. Como alternativa, você pode vincular um repositório de origem em um provedor de repositório Git compatível. Para obter mais informações, consulte [Vincular um repositório de origem](#).

### Como criar um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu projeto.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.

4. Escolha Adicionar repositório e selecione Criar repositório.
5. Em Nomes de repositórios, forneça um nome para o repositório. Neste guia, usamos *codecatalyst-source-repository*, mas você pode escolher um nome diferente. Os nomes dos repositórios devem ser exclusivos em um projeto. Para ter mais informações sobre os requisitos para nomes de repositórios, consulte [Cotas para repositórios de origem em CodeCatalyst](#).
6. (Opcional) Em Descrição, adicione uma descrição para o repositório que ajudará outros usuários no projeto a entender para que o repositório é usado.
7. Selecione Criar repositório (padrão). Essa opção cria um repositório que inclui uma ramificação padrão e um arquivo README.md. Ao contrário de um repositório vazio, você pode usar esse repositório assim que ele for criado.
8. Na Ramificação padrão, deixe o nome como principal, a menos que você tenha um motivo para escolher um nome diferente. Todos os exemplos neste guia usam o nome principal para a ramificação padrão.
9. (Opcional) Adicione um arquivo `.gitignore` para o tipo de código que você planeja enviar.
10. Escolha Criar.

 Note

CodeCatalyst adiciona um README.md arquivo ao seu repositório quando você o cria. CodeCatalyst também cria uma confirmação inicial para o repositório em uma ramificação padrão chamada main. Você pode editar ou excluir o arquivo README.md, mas não pode excluir a ramificação padrão.

## Como criar um repositório de origem vazio

1. No CodeCatalyst console, navegue até o projeto em que você deseja criar um repositório vazio.
2. Na página de resumo do seu projeto, em Repositórios de origem, selecione Adicionar repositório e, depois, selecione Criar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Escolha Adicionar repositório e selecione Criar repositório.
3. Em Nomes de repositórios, forneça um nome para o repositório. Neste guia, usamos *codecatalyst-source-repository*, mas você pode escolher um nome diferente. Os nomes dos repositórios devem ser exclusivos em um projeto. Para ter mais informações

sobre os requisitos para nomes de repositórios, consulte [Cotas para repositórios de origem em CodeCatalyst](#).

4. (Opcional) Em Descrição, adicione uma descrição para o repositório que ajudará outros usuários no projeto a entender para que o repositório é usado.
5. Selecione Criar repositório vazio e, depois, selecione Criar.

## Clonar um repositório Git existente em um repositório de origem

Você pode clonar um repositório Git existente em um repositório de origem vazio na Amazon. CodeCatalyst Essa é uma maneira rápida de começar a usar o código que estava hospedado anteriormente em outro provedor de repositório Git. CodeCatalyst Você pode clonar o conteúdo do repositório criando um clone espelhado e, em seguida, empurrando o espelho para. CodeCatalyst Como alternativa, se você tiver um repositório local do repositório cujo conteúdo deseja adicionar CodeCatalyst, poderá adicionar o repositório de CodeCatalyst origem como outro remoto ao repositório local e, em seguida, enviar para o repositório de origem vazio. As duas abordagens são igualmente válidas. Usar um clone espelho não apenas mapeia ramificações, como também todas as referências. É uma maneira simples e limpa de criar uma cópia funcional do repositório em CodeCatalyst. Adicionar um controle remoto a um repositório local que aponta para um repositório de CodeCatalyst origem vazio adicionará o conteúdo do repositório CodeCatalyst, mas também permitirá que você faça envios do repositório local para o repositório de CodeCatalyst origem e para o repositório remoto Git original. Isso pode ser útil se você quiser manter o código em diferentes repositórios remotos, mas pode gerar conflitos se outros desenvolvedores estiverem confirmando o código para apenas um dos remotos.

Os procedimentos a seguir usam comandos básicos do Git para realizar essa tarefa. Há muitas formas de realizar tarefas no Git, incluindo clonar. Para ter mais informações, consulte a [documentação do Git](#).

### Important

Você deve criar um repositório vazio CodeCatalyst antes de poder clonar conteúdo nele. Também é necessário ter um token de acesso pessoal. Para obter mais informações, consulte [Como criar um repositório de origem vazio](#) e [Criar um token de acesso pessoal](#).

Para usar **git clone --mirror** para clonar um repositório Git existente no CodeCatalyst

1. No CodeCatalyst console, navegue até o projeto em que você criou um repositório vazio.
2. Na página de resumo do seu projeto, selecione o repositório vazio na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Selecione o nome do repositório vazio na lista de repositórios de origem do projeto.
3. Copie o URL do clone HTTPS do repositório vazio. Você precisará dele para enviar o clone espelho. Por exemplo, se você nomeou o repositório de origem MyExampleRepo no MyExampleProject projeto no ExampleCorp espaço e seu nome de usuário é LiJuan, sua URL de clone pode ter a seguinte aparência:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

4. Em uma linha de comando ou janela de terminal, use o **git clone --mirror** comando para criar um clone espelho do repositório Git no qual você deseja clonar. CodeCatalyst Por exemplo, se você quiser criar um clone espelho do repositório codecatalyst-blueprints em, insira o seguinte comando: GitHub

```
git clone --mirror https://github.com/aws/codecatalyst-blueprints.git
```

5. Altere os diretórios para o diretório onde você criou o clone.

```
cd codecatalyst-blueprints.git
```

6. Execute o **git push** comando, especificando a URL e o nome do repositório de CodeCatalyst origem de destino e a **--all** opção. (Este é o URL que você copiou na Etapa 3.) Por exemplo:

```
git push https://LiJuan@git.us-west-2.codecatalyst.aws/
v1/ExampleCorp/MyExampleProject/MyExampleRepo --all
```

Para adicionar um controle remoto e enviar um repositório local para CodeCatalyst

1. No CodeCatalyst console, navegue até o projeto em que você criou um repositório vazio.
2. Na página de resumo do seu projeto, selecione o repositório vazio na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher

Repositórios de origem. Selecione o nome do repositório vazio na lista de repositórios de origem do projeto.

3. Copie o URL do clone HTTPS do repositório vazio. Você precisará dele para enviar o clone espelho. Por exemplo, se você nomeou o repositório de origem **MyExampleRepo** no **MyExampleProject** projeto no **ExampleCorp** espaço e seu nome de usuário é **LiJuan**, sua URL de clone pode ter a seguinte aparência:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

4. Em uma linha de comando ou janela de terminal, altere os diretórios para o repositório local para o qual você deseja enviar. CodeCatalyst
5. Execute o comando `git remote -v` para ver os repositórios remotos existentes no repositório local. Por exemplo, se você estiver clonando um repositório local de um repositório AWS CodeCommit nomeado **MyDemoRepo** na região Leste dos EUA (Ohio), a saída do comando pode ser parecida com:

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

Copie o URL remoto se quiser continuar usando o repositório.

6. Use o `git remote remove` comando para remover o CodeCommit repositório URLs para busca e envio para origem:

```
git remote remove origin
```

7. Use o comando `git remote add` para adicionar a URL do repositório de CodeCatalyst origem como o controle remoto de busca e push para seu repositório local. Por exemplo:

```
git remote add origin https://LiJuan@git.us-west-2.codecatalyst.aws/
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Isso substitui o URL push do CodeCommit repositório pelo URL do repositório de CodeCatalyst origem, mas não altera o URL de busca. Portanto, se você executar o comando `git remote -v` novamente, verá que agora está retirando (buscando) o código do repositório CodeCommit remoto, mas está configurado para enviar as alterações do seu repositório local para o repositório de origem: CodeCatalyst



```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
origin https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/
MyExampleProject/MyExampleRepo (push)
```

Opcionalmente, você pode adicionar novamente a URL CodeCommit remota se quiser enviar para os dois repositórios com o `git remote set-url` comando:

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

8. Execute o comando `git push` para enviar o repositório local para todos os repositórios remotos de envio configurados. Você também pode executar o comando `git push -u -origin`, especificando a opção `--all` de enviar o repositório local para ambos os repositórios. Por exemplo:

```
git push -u -origin --all
```

#### Tip

Dependendo da sua versão do Git, `--all` pode não funcionar para enviar todas as ramificações do repositório local para o repositório vazio. Talvez seja necessário conferir e enviar cada ramificação separadamente.

## Vincular um repositório de origem

Ao vincular um repositório de origem a um projeto, você pode incluir repositórios que tenham uma CodeCatalyst extensão para o serviço que hospeda o repositório, se essa extensão estiver instalada no seu espaço. Somente usuários com o perfil de administrador do espaço podem instalar extensões. Depois que a extensão for instalada, você poderá vincular a repositórios configurados para acesso por essa extensão. Para ter mais informações, consulte [Instalação de uma extensão em um espaço](#) ou siga [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

#### Important

Depois de instalar uma extensão de repositório, todos os repositórios aos quais você vincular CodeCatalyst terão seu código indexado e armazenado. CodeCatalyst Isso tornará o código

pesquisável em CodeCatalyst. Para entender melhor a proteção de dados do seu código ao usar repositórios vinculados CodeCatalyst, consulte [Proteção de dados](#) no Guia do CodeCatalyst usuário da Amazon.

Você pode vincular um repositório a somente um projeto em um espaço. Você não pode vincular um repositório arquivado. Embora você possa vincular um repositório vazio, você não pode usá-lo CodeCatalyst até inicializá-lo com um commit inicial que crie uma ramificação padrão. Além disso:

- Um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados, GitHub repositórios Bitbucket ou repositórios de projetos com GitLab projetos. CodeCatalyst
- Você não pode vincular um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto que tenha o mesmo nome de um repositório em um projeto. CodeCatalyst
- A extensão de GitHub repositórios não é compatível com os repositórios do GitHub Enterprise Server.
- A extensão de Repositórios do Bitbucket não é compatível com os repositórios do Bitbucket Data Center.
- A extensão GitLab repositories não é compatível com repositórios de projetos GitLab autogerenciados.
- Não é possível usar os recursos Escrever descrição para mim ou Resumir comentários com repositórios vinculados. Esses recursos só estão disponíveis em pull requests em CodeCatalyst.

Embora você possa vincular um GitHub repositório, um repositório do Bitbucket ou um repositório GitLab do projeto como colaborador, você só pode desvincular um repositório de terceiros como administrador do Space ou administrador do projeto. Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

#### Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-

la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#). Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

## Como vincular um repositório de origem

1. Navegue até o projeto onde você deseja vincular um repositório.

### Note

Antes de vincular um repositório, um usuário com o perfil de administrador do espaço deve primeiro instalar a extensão para o provedor que hospeda o repositório. Para obter mais informações, consulte [Instalação de uma extensão em um espaço](#).

2. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
3. Selecione Adicionar repositório e Vincular repositório.
4. No menu suspenso Provedor de repositório, escolha um dos seguintes provedores de repositório terceirizados: ou Bitbucket. GitHub
5. Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você selecionar para vincular:
  - GitHub repositórios: vincule um GitHub repositório.
    1. No menu suspenso da GitHub conta, escolha a GitHub conta que contém o repositório que você deseja vincular.
    2. No menu suspenso do GitHub repositório, escolha a GitHub conta à qual você deseja vincular seu projeto. CodeCatalyst
    3. (Opcional) Se você não vê um GitHub repositório na lista de repositórios, ele pode não ter sido configurado para acesso ao repositório no aplicativo da Amazon CodeCatalyst em. GitHub Você pode configurar quais GitHub repositórios podem ser usados CodeCatalyst na conta conectada.
      - a. Navegue até sua [GitHub](#) conta, escolha Configurações e, em seguida, escolha Aplicativos.

- b. Na guia **GitHub Aplicativos instalados**, escolha **Configurar** para o CodeCatalyst aplicativo Amazon.
- c. Siga um destes procedimentos para configurar o acesso aos GitHub repositórios aos quais você deseja vincular: CodeCatalyst
  - Para fornecer acesso a todos os repositórios atuais e futuros, selecione **Todos os repositórios**.
  - Para fornecer acesso a repositórios específicos, escolha **Somente selecionar repositórios**, escolha a lista suspensa **Selecionar repositórios** e, em seguida, escolha um repositório ao qual você deseja permitir a vinculação. CodeCatalyst
- **Repositórios do Bitbucket**: vincule um repositório do Bitbucket.
  - 1. No menu suspenso **Espaço de trabalho do Bitbucket**, selecione o espaço de trabalho do Bitbucket que contém o repositório que você deseja vincular.
  - 2. No menu suspenso do repositório do Bitbucket, escolha o repositório do Bitbucket ao qual você deseja vincular seu projeto. CodeCatalyst

 **Tip**

Se o nome do repositório estiver em cinza, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto na Amazon. CodeCatalyst

## 6. Escolha **Vincular**.

Se você não quiser mais usar um GitHub repositório, um repositório Bitbucket ou um repositório de GitLab projetos no CodeCatalyst, você pode desvinculá-lo de um projeto. CodeCatalyst Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

## Visualizar um repositório de origem

Você pode visualizar os repositórios de origem associados a um projeto na Amazon CodeCatalyst. Para repositórios de origem em CodeCatalyst, a página de visão geral de um repositório fornece uma visão geral rápida das informações e atividades nesse repositório, incluindo:

- A descrição do repositório, se houver
- O número de ramificações no repositório
- O número de solicitações pull abertas para o repositório
- O número de fluxos de trabalho relacionados para o repositório
- Os arquivos e pastas na ramificação padrão ou na ramificação selecionada
- O título, o autor e a data da última confirmação na ramificação exibida
- O conteúdo do arquivo README.md renderizado no Markdown, se algum arquivo README.md estiver incluído

Essa página também fornece links para confirmações, ramificações e solicitações pull do repositório, bem como uma maneira rápida de abrir, visualizar e editar arquivos individuais.

 Note

Você não pode visualizar essas informações sobre repositórios vinculados no CodeCatalyst console. Para visualizar informações sobre repositórios vinculados, escolha o link na lista de repositórios para abrir esse repositório no serviço que o hospeda.

Como navegar até os repositórios de origem de um projeto

1. Navegue até o projeto e faça o seguinte:
  - Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório.
  - No painel de navegação, selecione Código e, depois, selecione Repositórios de origem. Em Repositórios de origem, selecione o nome do repositório na lista. Você pode filtrar a lista de repositórios digitando parte do nome do repositório na barra de filtro.
2. Na página inicial do repositório, visualize o conteúdo do repositório e as informações sobre os recursos associados, como o número de solicitações pull e fluxos de trabalho. Por padrão, o conteúdo da ramificação padrão é exibido. Você pode alterar a visualização selecionando uma ramificação diferente na lista suspensa.

 Tip

Você também pode navegar rapidamente até os repositórios do seu projeto selecionando Ver código do projeto na página de resumo do projeto.

## Editar as configurações de um repositório de origem

Você pode gerenciar as configurações do seu repositório, incluindo editar a descrição de um repositório, escolher a ramificação padrão, criar e gerenciar regras de ramificação e criar e gerenciar regras de aprovação para pull requests em. CodeCatalyst Isso pode ajudar os membros do projeto a entender para que o repositório é usado e ajudar você a aplicar as melhores práticas e processos usados pela equipe.

 Note

Não é possível editar o nome de um repositório de origem. Você não pode editar o nome, a descrição ou outras informações de um repositório vinculado no CodeCatalyst. Para modificar as informações sobre um repositório vinculado, edite-as no provedor que hospeda o repositório vinculado. Para ter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.

### Como editar as configurações de um repositório

1. No CodeCatalyst console, navegue até o projeto que contém o repositório de origem cujas configurações você deseja editar.
2. Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Selecione o nome do repositório na lista de repositórios de origem do projeto.
3. Na página de visão geral do repositório, selecione Mais e, depois, selecione Gerenciar configurações.
4. Siga um dos procedimentos abaixo:
  - Edite a descrição do repositório e selecione Salvar.

- Para alterar a ramificação padrão do repositório, em Ramificação padrão, selecione Editar. Para obter mais informações, consulte [Gerenciar a ramificação padrão de um repositório](#).
- Para adicionar, remover ou alterar uma regra sobre quais perfis do projeto têm permissão para realizar determinadas ações em uma ramificação, em Regras de ramificação, selecione Editar. Para obter mais informações, consulte [Gerenciar as ações permitidas para uma ramificação com regras de ramificação](#).
- Para adicionar, remover ou alterar uma regra de aprovação para mesclar solicitações pull em uma ramificação, em Regras de aprovação, selecione Editar. Para obter mais informações, consulte [Gerenciamento de requisitos para mesclar uma solicitação pull com regras de aprovação](#).

## Clonar um repositório de origem

Para trabalhar de forma eficaz com vários arquivos, ramificações e confirmações nos repositórios de origem, clonar o repositório de origem em seu computador local e usar um cliente Git ou um ambiente de desenvolvimento integrado (IDE) para fazer alterações. Confirme e envie suas alterações para o repositório de origem para trabalhar com CodeCatalyst recursos como problemas e pull requests. Você também pode optar por criar um Ambiente de Desenvolvimento para trabalhar com o código. A criação de um Ambiente de Desenvolvimento clona automaticamente o repositório e a ramificação especificada no nele.

### Note

Você não pode clonar repositórios vinculados no CodeCatalyst console nem criar ambientes de desenvolvimento para eles. Para clonar um repositório vinculado localmente, escolha o link na lista de repositórios para abrir esse repositório no serviço que o hospeda e, depois, clone-o. Para ter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.

### Como criar um Ambiente de Desenvolvimento em um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
3. Selecione o repositório de origem em que você deseja trabalhar no código.
4. Escolha Criar ambiente de desenvolvimento.

5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Execute um destes procedimentos:
  - Selecione Trabalhar na ramificação existente e, depois, escolha uma ramificação no menu suspenso Ramificação existente.
  - Selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.
7. Opcionalmente, adicione um nome para o Ambiente de Desenvolvimento ou edite sua configuração.
8. Escolha Criar.

### Como clonar um repositório de origem

1. Navegue até o projeto.
2. Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Selecione o nome do repositório na lista de repositórios de origem do projeto. Você pode filtrar a lista de repositórios digitando parte do nome do repositório na barra de filtro.
- 3.
4. Selecione Clonar um repositório. Copie o URL do clone para o repositório.

#### Note

Se você não tiver um token de acesso pessoal (PAT), selecione Criar token. Copie e salve o token em um lugar seguro. Você usará esse PAT quando solicitado a fornecer uma senha pelo seu cliente Git ou ambiente de desenvolvimento integrado (IDE).

5. Execute um destes procedimentos:
  - Para clonar um repositório em seu computador local, abra um terminal ou linha de comando e execute o comando git clone com o URL do clone após o comando. Por exemplo:

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo
```



Quando solicitado a inserir uma senha, cole o PAT salvo anteriormente.

 Note

Se seu sistema operacional fornece gerenciamento de credenciais ou se você instalou um sistema de gerenciamento de credenciais, forneça o PAT apenas uma vez. Caso contrário, talvez seja necessário fornecer o PAT para cada operação do Git. Como prática recomendada, assegure-se de que seu sistema de gerenciamento de credenciais armazene seu PAT de forma segura. Não inclua o PAT como parte da string de URL do clone.

- Para clonar um repositório usando um IDE, siga a documentação do seu IDE. Escolha a opção de clonar um repositório Git e fornecer o URL. Quando for solicitada uma senha, forneça o PAT.

## Exclusão de um repositório de origem

Se um repositório de origem para um CodeCatalyst projeto da Amazon não for mais necessário, você poderá excluí-lo. A exclusão de um repositório de origem também exclui todas as informações do projeto armazenadas no repositório. Se algum fluxo de trabalho depender do repositório de origem, esses fluxos de trabalho serão excluídos da lista de fluxos de trabalho do projeto após a exclusão do repositório. Os problemas que fazem referência ao repositório de origem não serão excluídos nem alterados, mas quaisquer links para o repositório de origem adicionados aos problemas falharão quando o repositório for excluído.

 Important

A exclusão de um repositório de origem não pode ser desfeita. Depois de excluir um repositório de origem, você não poderá mais cloná-lo, extrair nem enviar dados para ele. A exclusão de um repositório de origem não exclui nenhuma cópia local do repositório (repositórios locais). Para excluir um repositório local, use o diretório do seu computador local e as ferramentas de gerenciamento de arquivo.

 Note

Você não pode excluir um repositório vinculado no CodeCatalyst console. Para excluir um repositório vinculado, selecione o link na lista de repositórios para abrir esse repositório no serviço que o hospeda e, depois, exclua-o. Para ter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.

Para remover um repositório vinculado de um projeto, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

### Como excluir um repositório de origem

1. Navegue até o projeto que contém o repositório de origem que deseja excluir.
2. Na página de resumo do seu projeto, selecione o repositório que você deseja na lista e, depois, selecione Visualizar repositório. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem. Selecione o nome do repositório na lista de repositórios de origem do projeto.
3. Na página inicial do repositório, escolha Mais, Gerenciar configurações e Excluir repositório.
4. Revise a ramificação, as solicitações pull e as informações relacionadas ao fluxo de trabalho para ajudar a garantir que você não esteja excluindo um repositório que ainda esteja em uso ou tenha um trabalho inacabado. Se quiser continuar, digite excluir e selecione Excluir.

## Organizando seu trabalho de código-fonte com filiais na Amazon CodeCatalyst

No Git, as ramificações são simplesmente ponteiros ou referências a uma confirmação. Durante o desenvolvimento, elas são uma forma conveniente de organizar seu trabalho. As ramificações podem ser usadas para separar o trabalho em uma versão nova ou diferente de arquivos sem afetar o trabalho em outras ramificações. Elas também podem ser usadas para desenvolver novos recursos, armazenar uma versão específica do seu projeto e muito mais. Você pode configurar regras para ramificações nos repositórios de origem para limitar determinadas ações em uma ramificação a perfis específicos nesse projeto.

Os repositórios de origem na Amazon CodeCatalyst têm conteúdo e uma ramificação padrão, independentemente de como você os cria. Os repositórios vinculados podem não ter uma

ramificação ou conteúdo padrão, mas não podem ser usados CodeCatalyst até que você os inicialize e crie uma ramificação padrão. Ao criar um projeto usando um blueprint, CodeCatalyst cria um repositório de origem para esse projeto que inclui um arquivo README.md, código de amostra, definições de fluxo de trabalho e outros recursos. Quando você cria um repositório de origem sem usar um esquema, um arquivo README.md é adicionado para você como uma primeira confirmação e uma ramificação padrão é criada para você. Essa ramificação padrão é chamada principal. Essa ramificação padrão é aquela usada como base ou ramificação padrão em repositórios locais quando os usuários clonam o repositório.

#### Note

Não é possível excluir a ramificação padrão. A primeira ramificação criada para um repositório de origem é a ramificação padrão desse repositório. Além disso, a pesquisa exibe somente os resultados da ramificação padrão. Você não pode pesquisar código em outras ramificações.

A criação de um repositório CodeCatalyst também cria um primeiro commit, que cria uma ramificação padrão com um arquivo README.md incluído nela. O nome dessa ramificação padrão é principal. Esse é o nome usado para a ramificação padrão em todos os exemplos desse guia.

## Tópicos

- [Criar uma ramificação](#)
- [Gerenciar a ramificação padrão de um repositório](#)
- [Gerenciar as ações permitidas para uma ramificação com regras de ramificação](#)
- [Comandos do Git para ramificações](#)
- [Visualização de ramificações e detalhes](#)
- [Exclusão de uma ramificação](#)

## Criar uma ramificação

Você pode usar o CodeCatalyst console para criar ramificações em um CodeCatalyst repositório. As ramificações que você criar ficarão visíveis para outros usuários na próxima vez que eles extraírem as alterações do repositório.

 Tip

Você também pode criar ramificações como parte da criação de um Ambiente de Desenvolvimento para trabalhar em seu código. Para obter mais informações, consulte [Criar um Ambiente de Desenvolvimento](#).

Você também pode usar o Git para criar ramificações. Para obter mais informações, consulte [Comandos comuns do Git para ramificações](#).

### Como criar uma ramificação (console)

1. No CodeCatalyst console, navegue até o projeto em que seu repositório de origem reside.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.
3. Selecione o repositório onde você deseja criar uma ramificação.
4. Na página de visão geral do repositório, selecione Mais e, depois, escolha Criar ramificação.
5. Insira um nome para a ramificação.
6. Selecione uma ramificação a partir da qual criar a ramificação e, depois, selecione Criar.

## Gerenciar a ramificação padrão de um repositório

Você pode especificar qual ramificação usar como ramificação padrão em um repositório de origem na Amazon CodeCatalyst. Todos os repositórios de origem CodeCatalyst têm conteúdo e uma ramificação padrão, independentemente de como você os cria. Se você usar um esquema para criar um projeto, a ramificação padrão no repositório de origem criado para esse projeto será chamada de principal. O conteúdo da ramificação padrão é exibido automaticamente na página de visão geral desse repositório.

 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

A ramificação padrão é tratada de forma um pouco diferente de todas as outras ramificações em um repositório de origem. Ela tem um rótulo especial ao lado do nome, Padrão. A ramificação padrão é aquela usada como base ou ramificação padrão em repositórios locais quando os usuários clonam o repositório para computadores locais com um cliente Git. Também é o padrão usado na criação de fluxos de trabalho para armazenar arquivos YAML do fluxo de trabalho e para armazenar informações sobre problemas. Ao usar pesquisar em CodeCatalyst, somente a ramificação padrão de um repositório é pesquisada. Como a ramificação padrão é fundamental para muitos aspectos dos projetos, você não pode excluir uma ramificação se ela for especificada como a ramificação padrão. No entanto, você pode optar por usar uma ramificação diferente como ramificação padrão. Se você fizer isso, todas as [regras de ramificação](#) aplicadas à antiga ramificação padrão serão aplicadas automaticamente à ramificação especificada como ramificação padrão.

 Note

Você deve ter a função de administrador do projeto para alterar a ramificação padrão dos repositórios de origem nos CodeCatalyst projetos. Isso não se aplica aos repositórios vinculados.

### Como visualizar e alterar a ramificação padrão de um repositório

1. Acesse o projeto no qual reside seu repositório.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.  
  
Selecione o repositório em que deseja exibir as configurações, incluindo a ramificação padrão.
3. Na página de visão geral do repositório, selecione Mais e, depois, selecione Gerenciar configurações.
4. Na Ramificação padrão, o nome da ramificação especificada como ramificação padrão é exibido junto com um rótulo chamado Padrão ao lado do nome. Esse mesmo rótulo aparece ao lado do nome da ramificação na lista de ramificações em Ramificações.
5. Para alterar a ramificação padrão, selecione Editar.

 Note

Você deve ter o perfil de administrador do projeto para alterar a ramificação padrão.

- Escolha o nome da ramificação que você deseja tornar a ramificação padrão na lista suspensa e selecione Salvar.

## Gerenciar as ações permitidas para uma ramificação com regras de ramificação

Quando você cria uma ramificação, determinadas ações são permitidas para essa ramificação com base nas permissões desse perfil. É possível alterar quais ações são permitidas para uma ramificação específica configurando regras de ramificação. As regras de ramificação são baseadas no perfil que um usuário tem no projeto. Você pode optar por limitar algumas ações predefinidas, como enviar confirmações para uma ramificação, para usuários com um perfil específico em um projeto. Isso pode ajudar você a proteger ramificações específicas em um projeto, limitando quais perfis podem realizar determinadas ações. Por exemplo, se você configurar uma regra de ramificação para permitir que somente usuários com o perfil Administrador do projeto se mesclam ou façam envios a essa ramificação, os usuários com outros perfis no projeto não poderão fazer alterações no código dessa ramificação.

Você deve pensar cuidadosamente em todas as implicações da criação de uma regra para uma ramificação. Por exemplo, se você optar por limitar os envios a uma ramificação aos usuários com o perfil Administrador do projeto, os usuários com o perfil Colaborador não poderão criar nem editar fluxos de trabalho nessa ramificação porque o YAML do fluxo de trabalho está armazenado nessa ramificação e esses usuários não podem confirmar e enviar alterações ao YAML. Como prática recomendada, teste todas as regras de ramificação depois de criá-las para garantir que elas não tenham nenhum impacto não pretendido. Você também pode usar regras de ramificação com regras de aprovação para solicitações pull. Para obter mais informações, consulte [Gerenciamento de requisitos para mesclar uma solicitação pull com regras de aprovação](#).

 Note

Você deve ter a função de administrador do projeto para gerenciar regras de ramificação para repositórios de origem em CodeCatalyst projetos. Não é possível criar regras de ramificação para repositórios vinculados.

Só é possível criar regras de ramificação mais restritivas do que as permissões padrão para o perfil. Não é possível criar regras de ramificação mais permissivas do que o autorizado pelo perfil de um usuário no projeto. Por exemplo, não é possível criar uma regra de ramificação que permita que usuários com o perfil de revisor acessem a ramificação.

As regras de ramificação aplicadas à ramificação padrão do seu repositório de origem se comportarão de maneira um pouco diferente das regras de ramificação aplicadas a outras ramificações. Qualquer regra aplicada à ramificação padrão será aplicada automaticamente a qualquer ramificação que você especificar como ramificação padrão. A ramificação anteriormente definida como a ramificação padrão ainda manterá as regras aplicadas a ela, exceto que ela não terá mais proteção contra exclusão. Essa proteção é aplicada somente à ramificação padrão atual.

As regras de ramificação têm dois estados, Padrão e Personalizado. O padrão indica que as ações permitidas em uma ramificação são aquelas que correspondem às permissões da função que o usuário tem CodeCatalyst para as ações da ramificação. Para saber mais sobre quais perfis têm quais permissões, consulte [Concessão de acesso com perfis de usuário](#). Personalizado indica que uma ou mais ações de ramificação têm ações que apresentam uma lista específica de perfis permitidos para realizar essa ação, diferente das permissões padrão concedidas pelo perfil de um usuário no projeto.

#### Note

Se você criar uma regra de ramificação para restringir uma ou mais ações para uma ramificação, a ação Excluir a ramificação será automaticamente definida para permitir que somente usuários com o perfil de administrador do projeto excluam essa ramificação.

A tabela a seguir lista as ações e as configurações padrão dos perfis autorizados a realizar essas ações em uma ramificação.

#### Ações e perfis da ramificação

| Ação da ramificação                                                                        | Perfis autorizados a realizar essa ação quando nenhuma regra de ramificação é aplicada |
|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Unificação com a ramificação (isso inclui unificar uma solicitação pull com a ramificação) | Administrador do projeto, Colaborador                                                  |

|                                              |                                                                                        |
|----------------------------------------------|----------------------------------------------------------------------------------------|
| Ação da ramificação                          | Perfis autorizados a realizar essa ação quando nenhuma regra de ramificação é aplicada |
| Envio para a ramificação                     | Administrador do projeto, Colaborador                                                  |
| Exclusão da ramificação                      | Administrador do projeto, Colaborador                                                  |
| Exclusão da ramificação (ramificação padrão) | Não permitido                                                                          |

Você não pode excluir regras de ramificação, mas pode atualizá-las para permitir ações de todos os perfis que poderiam realizar essa ação em uma ramificação, o que efetivamente remove a regra.

#### Note

Você deve ter a função de administrador do projeto para configurar regras de ramificação para repositórios de origem em CodeCatalyst projetos. Isso não se aplica aos repositórios vinculados. Os repositórios vinculados não oferecem suporte às regras de ramificação em CodeCatalyst.

### Como visualizar e editar regras de ramificação para um repositório

1. Acesse o projeto no qual reside seu repositório.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja visualizar regras de ramificação.

3. Na página de visão geral do repositório, selecione Ramificações.
4. Na coluna Regras de ramificação, visualize o status das regras para cada ramificação do repositório. Padrão indica que as regras para a ação da ramificação são as padrão para qualquer ramificação criada em um repositório de origem e correspondem às permissões concedidas a esses perfis em um projeto. Personalizado indica que uma ou mais ações de ramificação têm regras que restringem uma ou mais ações permitidas para essa ramificação a um conjunto diferente de perfis.

Para ver as especificações das regras de ramificação de uma ramificação, selecione a palavra Padrão ou Personalizado ao lado da ramificação que você deseja revisar.



5. Para criar ou alterar uma regra de ramificação, selecione Gerenciar configurações. Na página de configurações do repositório de origem, em Regras de ramificação, selecione Editar.
6. Em Ramificação, selecione o nome da ramificação para a qual você deseja configurar um perfil na lista suspensa. Em relação a cada um dos tipos de ação permitidos, selecione os perfis que você deseja permitir que executem essa ação na lista suspensa e escolha Salvar.

## Comandos do Git para ramificações

Você pode usar o Git para criar, gerenciar e excluir branches no clone do repositório de origem que você tem em seu computador (seu repositório local) ou em seus ambientes de desenvolvimento e, em seguida, confirmar e enviar suas alterações para o repositório de CodeCatalyst origem (o repositório remoto). Por exemplo:

### Comandos comuns do Git para ramificações

|                                                                                                                                                                                      |                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| Lista todas as ramificações no repositório local com um asterisco (*) exibido ao lado da sua ramificação atual.                                                                      | <code>git branch</code>                    |
| Extraí informações sobre todas as ramificações existentes no repositório remoto para o repositório local.                                                                            | <code>git fetch</code>                     |
| Lista todas as ramificações no repositório local e ramificações de rastreamento remoto no repositório local.                                                                         | <code>git branch -a</code>                 |
| Lista somente ramificações de rastreamento remoto no repositório local.                                                                                                              | <code>git branch -r</code>                 |
| Cria uma ramificação no repositório local usando o nome de ramificação especificado. Essa ramificação não aparecerá no repositório remoto até que você confirme e envie a alteração. | <code>git branch <i>branch-name</i></code> |

|                                                                                                                                                                                                                                                                                       |                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Cria uma ramificação no repositório local usando o nome de ramificação especificado e alterna para ela.                                                                                                                                                                               | <code>git checkout -b <i>branch-name</i></code>                                                                                      |
| Alterna para outra ramificação no repositório local usando o nome de ramificação especificado.                                                                                                                                                                                        | <code>git checkout <i>other-branch-name</i></code>                                                                                   |
| Envia uma ramificação do repositório local para o repositório remoto usando o apelido especificado do repositório local para o repositório remoto e o nome da ramificação especificado. Também configura informações de rastreamento upstream da ramificação no repositório local.    | <code>git push -u <i>remote-name</i> <i>branch-name</i></code>                                                                       |
| Mescla alterações de outra ramificação no repositório local à ramificação atual no repositório local.                                                                                                                                                                                 | <code>git merge <i>from-other-branch-name</i></code>                                                                                 |
| Exclui uma ramificação no repositório local, a menos que ela contenha trabalho que ainda não foi mesclado.                                                                                                                                                                            | <code>git branch -d <i>branch-name</i></code>                                                                                        |
| Exclui uma ramificação do repositório remoto usando o apelido especificado que o repositório local usa para o repositório remoto e o nome de ramificação especificado. (Observe o uso de dois pontos (:).) Como alternativa, especifique <code>--delete</code> como parte do comando. | <code>git push <i>remote-name</i> :<i>branch-name</i></code><br><code>git push <i>remote-name</i> --delete <i>branch-name</i></code> |

Para ter mais informações, consulte a documentação do Git.

## Visualização de ramificações e detalhes

Você pode visualizar informações sobre filiais remotas na Amazon CodeCatalyst, incluindo detalhes específicos dos arquivos, pastas e a confirmação mais recente de uma filial específica,

no CodeCatalyst console da Amazon. Você também pode usar os comandos do Git e seu sistema operacional local para visualizar essas informações para ramificações remotas e locais.

### Para visualizar ramificações (console)

1. No CodeCatalyst console, navegue até o projeto que contém o repositório de origem no qual você deseja visualizar as ramificações. Selecione Código, selecione Repositórios de origem e, depois, selecione o repositório de origem.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja visualizar uma ramificação.

3. A ramificação padrão do repositório é exibida. Você pode ver uma lista de arquivos e pastas na ramificação, informações sobre a confirmação mais recente e o conteúdo do arquivo README.md, se ele existir na ramificação. Para visualizar as informações de uma ramificação diferente, selecione-a na lista suspensa de ramificações do repositório.
4. Para visualizar todas as ramificações de um repositório, selecione Visualizar tudo. A página Ramificações exibe informações sobre o nome, a confirmação mais recente e as regras de cada ramificação.

Para ter informações sobre como usar o Git e o sistema operacional para visualizar ramificações e detalhes, consulte [Comandos comuns do Git para ramificações](#), a documentação do Git e a documentação do sistema operacional.

## Exclusão de uma ramificação

Se você não precisar mais de uma ramificação, poderá excluí-la. Por exemplo, se você mesclou uma ramificação com uma alteração de recurso na ramificação padrão e esse recurso foi lançado, talvez queira excluir a ramificação original do recurso, pois as alterações já fazem parte da ramificação padrão. Manter o número de ramificações baixo pode ajudar os usuários a descobrir a ramificação que contém as alterações nas quais desejam trabalhar. Quando você exclui uma ramificação, as cópias dessa ramificação permanecem nos clones do repositório nos computadores locais até que os usuários obtenham e sincronizem essas alterações.

### Como excluir uma ramificação (console)

1. Acesse o projeto no qual reside seu repositório.

2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja excluir uma ramificação.

3. Na página de visão geral do repositório, escolha o seletor suspenso ao lado do nome da ramificação e, depois, selecione Exibir tudo.
4. Escolha a ramificação que você deseja excluir e, depois, selecione Excluir ramificação.

 **Note**

Você não pode excluir a ramificação padrão de um repositório.

5. Uma caixa de diálogo de confirmação é exibida. Ela mostra o repositório, o número de pull requests abertos e o número de fluxos de trabalho associados à ramificação.
6. Para confirmar a exclusão da ramificação, digite excluir na caixa de texto e selecione Excluir.

Você também pode usar o Git para excluir ramificações. Para obter mais informações, consulte [Comandos comuns do Git para ramificações](#).

## Gerenciando arquivos de código-fonte na Amazon CodeCatalyst

Na Amazon CodeCatalyst, um arquivo é uma informação independente e controlada por versão disponível para você e outros usuários do repositório de origem e da ramificação em que o arquivo está armazenado. Você pode organizar os arquivos do repositório com uma estrutura de diretórios. CodeCatalyst rastreia automaticamente todas as alterações confirmadas em um arquivo. Você pode armazenar diferentes versões dele em diferentes ramificações do repositório.

Para adicionar ou editar vários arquivos em um repositório de origem, use um cliente Git, um Ambiente de Desenvolvimento ou um ambiente de desenvolvimento integrado (IDE). Para adicionar ou editar um único arquivo, você pode usar o CodeCatalyst console.

### Tópicos

- [Criar ou adicionar um arquivo](#)
- [Visualizar um arquivo](#)
- [Visualizar o histórico de alterações em um arquivo](#)
- [Edição de um arquivo](#)

- [Renomear ou excluir um arquivo](#)

## Criar ou adicionar um arquivo

Para criar e adicionar arquivos a um repositório de origem, você pode usar o CodeCatalyst console da Amazon, um ambiente de desenvolvimento, um ambiente de desenvolvimento integrado (IDE) conectado ou um cliente Git. O CodeCatalyst console inclui um editor de código para criar arquivos. Esse editor é uma forma conveniente de criar ou editar um arquivo simples, como um arquivo README.md, em uma ramificação em um repositório. Ao trabalhar em mais de um arquivo, pense em [criar um Ambiente de Desenvolvimento](#).

Como criar um Ambiente de Desenvolvimento em um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
3. Selecione o repositório de origem em que você deseja trabalhar no código.
4. Escolha Criar ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Execute um destes procedimentos:
  - Selecione Trabalhar na ramificação existente e, depois, escolha uma ramificação no menu suspenso Ramificação existente.
  - Selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.
7. Opcionalmente, adicione um nome para o Ambiente de Desenvolvimento ou edite sua configuração.
8. Escolha Criar.

Para criar um arquivo no CodeCatalyst console

1. Navegue até o projeto onde você deseja criar um arquivo. Para ter mais informações sobre como navegar até um repositório, consulte [Visualizar um repositório de origem](#).

2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório no qual você deseja criar o arquivo.

3. (Opcional) Escolha a ramificação onde você deseja criar o arquivo, se quiser criar o arquivo em uma ramificação diferente da ramificação padrão.
4. Selecione Criar arquivo.
5. Insira o nome do arquivo em Nome do arquivo. Adicione o conteúdo do arquivo no editor.

 Tip

Se você quiser criar o arquivo em uma subpasta ou subdiretório da raiz da ramificação, inclua essa estrutura como parte do nome do arquivo.

Quando estiver satisfeito com suas alterações, selecione Confirmar.

6. Em Nome do arquivo, revise o nome do arquivo e faça as alterações que desejar nele. Também é possível selecionar a ramificação onde você deseja criar o arquivo na lista de ramificações disponíveis em Ramificação. Em Mensagem de confirmação, opcionalmente, insira uma descrição breve, mas informativa, do motivo pelo qual você fez essa alteração. Isso será exibido como as informações básicas da confirmação que adiciona o arquivo ao repositório de origem.
7. Selecione Confirmar para confirmar e enviar o arquivo para o repositório de origem.

Você também pode adicionar arquivos a um repositório de origem clonando-o em seu computador local e usando um cliente Git ou um ambiente de desenvolvimento integrado (IDE) conectado para enviar seus arquivos e alterações.

 Note

Se quiser adicionar um submódulo do Git, você deve usar um cliente Git ou um Ambiente de Desenvolvimento e executar o comando `git submodule add`. Você não pode adicionar ou visualizar submódulos do Git no CodeCatalyst console nem visualizar as diferenças nos submódulos do Git nas pull requests. Para ter mais informações sobre os submódulos do Git, consulte a [documentação do Git](#).

Como adicionar um arquivo usando um cliente Git ou um ambiente de desenvolvimento integrado (IDE) conectado

1. Clone o repositório de origem para o computador local. Para obter mais informações, consulte [Clonar um repositório de origem](#).
2. Crie arquivos ou copie arquivos no seu repositório local.
3. Crie e envie uma confirmação seguindo um destes procedimentos:
  - Se você estiver usando um cliente Git, no terminal ou na linha de comando, execute o comando `git add`, especificando os nomes dos arquivos que você deseja adicionar. Como alternativa, para adicionar todos os arquivos adicionados ou alterados, execute o comando `git add` seguido por um ponto único ou duplo para indicar se você deseja incluir todas as alterações no nível do diretório atual (ponto único) ou todas as alterações no diretório atual e em todos os subdiretórios (ponto duplo). Para confirmar as alterações, execute o comando `git commit -m` e forneça uma mensagem de confirmação. Para enviar suas alterações para o repositório de origem CodeCatalyst, execute `git push`. Para ter mais informações sobre os comandos Git, consulte a documentação e [Comandos do Git para ramificações](#).
  - Se você estiver usando um Ambiente de Desenvolvimento ou um IDE, crie e adicione arquivos no IDE e, depois, confirme e envie suas alterações. Para ter mais informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#) ou a documentação do IDE.

## Visualizar um arquivo

Você pode visualizar os arquivos em seu repositório de origem no CodeCatalyst console da Amazon. Você pode visualizar os arquivos na ramificação padrão e em qualquer outra ramificação. O conteúdo do arquivo pode variar dependendo da ramificação que você escolher visualizar.

Para visualizar arquivos no CodeCatalyst console

1. Navegue até o projeto onde você deseja visualizar arquivos. Para obter mais informações, consulte [Visualizar um repositório de origem](#).
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja visualizar os arquivos.

3. Uma lista de arquivos e pastas é exibida para a ramificação padrão. Os arquivos são indicados por um ícone de papel, enquanto as pastas são indicadas por um ícone de pasta.
4. Faça o seguinte:
  - Para visualizar arquivos e pastas em uma ramificação diferente, selecione-a na lista de ramificações.
  - Para expandir uma pasta, selecione-a na lista.
5. Para visualizar o conteúdo de um arquivo específico, selecione-o na lista. O conteúdo do arquivo será exibido na ramificação. Para visualizar o conteúdo do arquivo em uma ramificação diferente, escolha a ramificação desejada no seletor de ramificações.

 Tip

Ao visualizar o conteúdo de um arquivo, você pode selecionar arquivos adicionais para visualizar em Visualizar arquivos. Para editar um arquivo, selecione Editar.

Você pode visualizar vários arquivos no console. Você também pode visualizar arquivos clonados em seu computador local usando um cliente Git ou um ambiente de desenvolvimento integrado (IDE). Para ter mais informações, consulte a documentação do cliente Git ou IDE.

 Note

Você não pode visualizar os submódulos do Git no console. CodeCatalyst Para ter mais informações sobre os submódulos do Git, consulte a [documentação do Git](#).

## Visualizar o histórico de alterações em um arquivo

Você pode ver o histórico de alterações em um arquivo no seu repositório de origem no CodeCatalyst console da Amazon. Isso pode ajudá-lo a entender as alterações feitas no arquivo por várias confirmações na ramificação em que você optou por visualizar o histórico do arquivo. Por exemplo, se você visualizar o histórico de alterações no arquivo **readme.md** na ramificação **main** do repositório de origem, verá uma lista de confirmações que incluíram alterações nesse arquivo nessa ramificação.



 Note

Você não pode visualizar o histórico de um arquivo em um repositório vinculado no CodeCatalyst console.

Para ver o histórico de um arquivo no CodeCatalyst console

1. Navegue até o projeto onde você deseja visualizar o histórico de um arquivo. Para obter mais informações, consulte [Visualizar um repositório de origem](#).
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.
3. Selecione o repositório onde você deseja visualizar o histórico de um arquivo. Selecione a ramificação onde deseja visualizar o histórico do arquivo e selecione o arquivo na lista. Selecione Visualizar histórico.
4. Revise a lista de confirmações que incluíram alterações nesse arquivo na ramificação especificada. Para visualizar os detalhes das alterações incluídas em uma confirmação específica, escolha a mensagem de confirmação dessa confirmação na lista. As diferenças entre essa confirmação e sua confirmação principal são exibidas.
5. Para revisar o histórico de alterações no arquivo em outra ramificação, use o seletor de ramificação para alterar as visualizações dessa ramificação, selecione o arquivo na lista de arquivos e selecione Visualizar histórico.

 Note

Você não pode ver o histórico de alterações nos submódulos do Git no console. CodeCatalyst Para ter mais informações sobre os submódulos do Git, consulte a [documentação do Git](#).

## Edição de um arquivo

Você pode editar arquivos individuais no CodeCatalyst console da Amazon. Para editar vários arquivos ao mesmo tempo, crie um Ambiente de Desenvolvimento ou clone o repositório e faça suas alterações usando um cliente Git ou um ambiente de desenvolvimento integrado (IDE). Para ter mais

informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#) ou [Clonar um repositório de origem](#).

Para editar um arquivo no CodeCatalyst console

1. Navegue até o projeto onde você deseja editar um arquivo. Para ter mais informações sobre como navegar até um repositório, consulte [Visualizar um repositório de origem](#).
2. Selecione o repositório onde você deseja editar o arquivo. Selecione Visualizar ramificações e, depois, selecione a ramificação em que você deseja trabalhar. Escolha o arquivo na lista de arquivos e pastas dessa ramificação.

O conteúdo do arquivo é exibido.

3. Escolha Editar.
4. No editor, edite o conteúdo do arquivo e selecione Confirmar. Opcionalmente, em Confirmar alterações, adicione mais informações sobre a alteração na mensagem de confirmação. Quando estiver satisfeito com suas alterações, selecione Confirmar.

## Renomear ou excluir um arquivo

Você pode renomear ou excluir arquivos em um Ambiente de Desenvolvimento, localmente em seu computador ou em um ambiente de desenvolvimento integrado (IDE). Depois de renomear ou excluir os arquivos, confirme e envie essas alterações para o repositório de origem. Você não pode renomear ou excluir arquivos no CodeCatalyst console da Amazon.

## Análise de código com pull requests na Amazon CodeCatalyst

Uma solicitação pull é a principal maneira para você e outros membros do projeto revisarem, comentarem e mesclarem alterações de código de uma ramificação para outra. Você pode usar solicitações pull para revisar alterações de código de maneira colaborativa em caso de alterações ou correções relativamente pequenas, adições significativas de recursos ou novas versões do seu software lançado. Se você usar problemas para rastrear o trabalho em seu projeto, poderá vincular problemas específicos a solicitações pull para ajudar a rastrear quais problemas estão sendo resolvidos pelas alterações de código na solicitação pull. Quando você cria, atualiza, comenta, mescla ou fecha uma solicitação pull, um e-mail é enviado automaticamente para o autor da solicitação pull, bem como para todos os revisores obrigatórios ou opcionais da solicitação pull.

 Tip

Você pode configurar quais eventos de solicitação pull receberá e-mails como parte do seu perfil. Para obter mais informações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

As solicitações pull exigem duas ramificações em um repositório de origem: uma ramificação de origem que contém o código que você deseja revisar e uma ramificação de destino, onde você deseja mesclar o código revisado. A ramificação de origem contém a confirmação AFTER, que é a confirmação que contém as alterações que você deseja mesclar na ramificação de destino. A ramificação de destino contém a confirmação BEFORE, que representa o estado do código (antes de a ramificação da solicitação pull ser mesclada à ramificação de destino).

 Note

Enquanto você cria uma solicitação pull, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois de criar a solicitação pull, a diferença exibida será entre a revisão da solicitação pull escolhida e a confirmação que foi a ponta da ramificação de destino quando você criou a solicitação pull. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

Enquanto uma solicitação pull é criada para um repositório de origem e ramificações específicas, você pode criá-las, visualizá-las, revisá-las e fechá-las como parte do trabalho com seu projeto. Você não precisa visualizar o repositório de origem para visualizar e trabalhar com solicitações pull. Um estado de solicitação pull é definido como Aberto ao criá-lo. A pull request permanece aberta até que você a mescle no CodeCatalyst console, o que muda o estado para Mesclado, ou a feche, o que altera o estado para Fechado.

Quando seu código for revisado, você poderá alterar o estado da solicitação pull de várias maneiras:

- Mescle o pull request no CodeCatalyst console. O código na ramificação de origem da solicitação pull será mesclado na ramificação de destino. O status da solicitação pull mudará para Mesclado. Ele não pode ser alterado de volta para Aberto.
- Mescle as ramificações localmente, envie suas alterações e, em seguida, feche a pull request no CodeCatalyst console.

- Use o CodeCatalyst console para fechar a pull request sem mesclar. Isso mudará o status para Fechado e não mesclará o código da ramificação de origem com a ramificação de destino.

Antes de criar uma solicitação pull:

- Confirme e envie as alterações de código que você deseja revisar para uma ramificação (a ramificação de origem).
- Configure notificações para seu projeto para que outros usuários possam ser notificados sobre quaisquer fluxos de trabalho executados quando você cria uma solicitação pull. (Esta etapa é opcional, mas recomendada.)

## Tópicos

- [Criar uma solicitação pull](#)
- [Visualizar solicitações pull](#)
- [Gerenciamento de requisitos para mesclar uma solicitação pull com regras de aprovação](#)
- [Revisão de uma solicitação pull](#)
- [Atualizar uma solicitação pull](#)
- [Mesclar uma solicitação pull](#)
- [Encerrar uma solicitação pull](#)

## Criar uma solicitação pull

A criação de solicitações pull ajuda outros usuários a ver e revisar suas alterações de código antes de mesclá-las em outra ramificação. Primeiro, você cria uma ramificação para suas alterações de código. Ela é chamada de ramificação de origem para uma solicitação pull. Depois de confirmar e enviar as alterações para o repositório, você pode criar uma solicitação pull que compara o conteúdo da ramificação de origem com o conteúdo da ramificação de destino.

Você pode criar uma pull request no CodeCatalyst console da Amazon a partir de uma ramificação específica, da página de pull requests ou da visão geral do projeto. A criação de uma solicitação pull a partir de uma ramificação específica fornece automaticamente o nome do repositório e a ramificação de origem na página de criação da solicitação pull. Ao criar uma solicitação pull, você receberá automaticamente e-mails sobre quaisquer atualizações na solicitação pull, bem como quando a solicitação pull for mesclada ou fechada.

 Note

Enquanto você cria uma solicitação pull, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a solicitação pull for criada, a diferença exibida será entre a revisão da solicitação pull escolhida e a confirmação que foi a ponta da ramificação de destino quando você criou a solicitação pull. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

Você pode usar o recurso Escrever descrição para mim ao criar solicitações pull para que o Amazon Q crie automaticamente uma descrição das alterações contidas em uma solicitação pull. Quando você escolhe essa opção, o Amazon Q analisa as diferenças entre a ramificação de origem que contém as alterações de código e a ramificação de destino na qual você deseja mesclar essas alterações. Em seguida, ele cria um resumo dessas alterações, bem como sua melhor interpretação da intenção e do efeito dessas alterações. Esse recurso está disponível somente na região Oeste dos EUA (Oregon) para CodeCatalyst pull requests. O recurso Escrever descrição para mim não está disponível para solicitações pull em repositórios vinculados.

 Note Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

Para criar uma solicitação pull

1. Navegue até o projeto.

## 2. Execute um destes procedimentos:

- No painel de navegação, escolha Código, Solicitações pull e, depois, selecione Criar solicitação pull.
  - Na página inicial do repositório, escolha Mais e, depois, selecione Criar solicitação pull.
  - Na página do projeto, selecione Criar solicitação pull.
3. No Repositório de origem, certifique-se de que o repositório de origem especificado seja aquele que contém o código confirmado. Essa opção só aparece se você não criou a solicitação pull na página principal do repositório.
  4. Em Ramificação de destino, selecione a ramificação na qual mesclar o código depois que ele for revisado.
  5. Em Ramificação de origem, selecione a ramificação que contém o código confirmado.
  6. Em Título da solicitação pull, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
  7. (Opcional) Na Descrição da solicitação pull, forneça informações, como um link para problemas ou uma descrição das alterações.

### Tip

Você pode escolher Escrever descrição para mim para gerar CodeCatalyst automaticamente uma descrição das alterações contidas na pull request. Você pode fazer alterações na descrição gerada automaticamente depois de adicioná-la à solicitação pull.

Essa funcionalidade exige que os recursos de IA generativa estejam habilitados para o espaço e não estejam disponíveis para solicitações pull em repositórios vinculados. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

8. (Opcional) Em Problemas, selecione Vincular problemas e, depois, selecione um problema na lista ou insira seu ID. Para desvincular um problema, selecione o ícone de desvinculação.
9. (Opcional) Em Revisores obrigatórios, selecione Adicionar revisores obrigatórios. Selecione na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que a solicitação pull seja mesclada na ramificação de destino.

 Note

Não é possível adicionar um revisor como revisor obrigatório e revisor opcional. Não é possível se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, selecione Adicionar revisores opcionais. Selecione na lista de membros do projeto para adicioná-los. Revisores opcionais não precisam aprovar as alterações como um requisito antes que a solicitação pull seja mesclada na ramificação de destino.
11. Revise as diferenças entre as ramificações. A diferença exibida em uma solicitação pull são as alterações entre a revisão na ramificação de origem e a base de mesclagem, que é a confirmação principal da ramificação de destino no momento em que a solicitação pull foi criada. Se nenhuma alteração for exibida, as ramificações podem ser idênticas ou você pode ter escolhido a mesma ramificação para a origem e o destino.
12. Quando a solicitação pull tiver o código e as alterações que você deseja revisar, selecione Criar.

 Note

Depois que você criar a solicitação pull, poderá adicionar comentários. Os comentários podem ser adicionados à solicitação pull ou a linhas individuais em arquivos, bem como à solicitação pull geral. Você pode adicionar links para recursos, como arquivos, usando o sinal @ seguido pelo nome do arquivo.

## Como criar uma solicitação pull a partir de uma ramificação

1. Navegue até o projeto onde você deseja criar uma solicitação pull.
2. No painel de navegação, selecione Repositórios de origem e, depois, escolha o repositório que contém a ramificação em que você tem alterações de código a serem revisadas.
3. Escolha a seta suspensa ao lado do nome da ramificação padrão e, depois, escolha a ramificação desejada na lista. Para visualizar todas as ramificações de um repositório, selecione Visualizar tudo.
4. Selecione Mais e, depois, Criar solicitação pull.
5. O repositório e a ramificação de origem estão pré-selecionados. Em Ramificação de destino, escolha a ramificação na qual você mesclará o código depois que ele for revisado. Em Título da

solicitação pull, insira um título que ajudará outros usuários do projeto a entender o que deve ser revisado e por quê. Opcionalmente, forneça mais informações na descrição do Pull Request, como colar um link para problemas relacionados ou adicionar uma descrição das alterações feitas. CodeCatalyst

 Note

Os fluxos de trabalho configurados para serem executados em eventos de criação de solicitação pull serão executados após a criação da solicitação pull, se a ramificação de destino da solicitação pull corresponder a uma das ramificações especificadas no fluxo de trabalho.

6. Revise as diferenças entre as ramificações. Se nenhuma alteração for exibida, as ramificações podem ser idênticas ou você pode ter escolhido a mesma ramificação para a origem e o destino.
7. (Opcional) Em Problemas, selecione Vincular problemas e, depois, selecione um problema na lista ou insira seu ID. Para desvincular um problema, selecione o ícone de desvinculação.
8. (Opcional) Em Revisores obrigatórios, selecione Adicionar revisores obrigatórios. Selecione na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que a solicitação pull seja mesclada na ramificação de destino.

 Note

Você não pode adicionar um revisor tanto como obrigatório quanto opcional. Não é possível se adicionar como revisor.

9. (Opcional) Em Revisores opcionais, selecione Adicionar revisores opcionais. Selecione na lista de membros do projeto para adicioná-los. Revisores opcionais não precisam aprovar as alterações antes que a solicitação pull seja mesclada na ramificação de destino.
10. Quando a solicitação pull tiver as alterações que você deseja revisar e incluir os revisores obrigatórios, selecione Criar.

Se você tiver algum fluxo de trabalho configurado para execução em que a ramificação corresponda à ramificação de destino na solicitação pull, verá informações sobre essas execuções de fluxo de trabalho em Visão geral na área Detalhes da solicitação pull após a criação da solicitação pull. Para obter mais informações, consulte [Adição de gatilhos aos fluxos de trabalho](#).



## Visualizar solicitações pull

Você pode visualizar pull requests para um projeto no CodeCatalyst console da Amazon. A página de resumo do projeto exibe todas as solicitações pull abertas para um projeto. Para ver todas as solicitações pull, independentemente do estado, navegue até a página de solicitações pull do projeto. Ao visualizar uma solicitação pull, você pode optar por ter um resumo de todos os comentários deixados sobre as alterações na solicitação pull criada para você.

### Note

#### Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

Para visualizar solicitações pull abertas

1. Navegue até o projeto em que deseja visualizar solicitações pull.
2. Na página do projeto, as solicitações pull abertas são exibidas, incluindo informações sobre quem criou a solicitação pull, qual repositório contém as ramificações da solicitação pull e a data em que a solicitação pull foi criada. Você pode filtrar a visualização da solicitação pull aberta por repositório de origem.
3. Para ver todas as solicitações pull, selecione Exibir tudo. Você pode usar os seletores para escolher entre as opções. Por exemplo, para ver todas as solicitações pull, selecione Qualquer status e Qualquer autor.

Como alternativa, no painel de navegação, selecione Código e Solicitações pull e use os seletores para refinar a visualização.

4. Na página Solicitações pull, você pode classificar as solicitações pull por ID, título, status e muito mais. Para personalizar quais informações e quantas informações são exibidas na página de solicitações pull, escolha o ícone de roda dentada.
5. Para ver uma solicitação pull específica, escolha-a na lista.
6. Para visualizar o status das execuções de fluxos de trabalho associadas a essa solicitação pull, se houver, selecione Visão geral e revise as informações na área Detalhes da solicitação pull em Execuções de fluxo de trabalho.

A execução do fluxo de trabalho ocorrerá se o fluxo de trabalho estiver configurado com eventos de criação ou revisão de solicitação pull e se os requisitos da ramificação de destino no fluxo de trabalho corresponderem à ramificação de destino especificada na solicitação pull. Para obter mais informações, consulte [Adição de gatilhos aos fluxos de trabalho](#).

7. Para visualizar problemas vinculados, se houver, selecione Visão geral e revise as informações em Detalhes da solicitação pull em Problemas. Se você quiser ver um problema vinculado, escolha o ID na lista.
8. (Opcional) Para criar um resumo dos comentários deixados nas alterações de código nas revisões da solicitação pull, selecione Criar resumo do conteúdo. O resumo não incluirá nenhum comentário deixado na solicitação pull geral.

 Note

Essa funcionalidade exige que os recursos de IA generativa estejam habilitados para o espaço, não estejam disponíveis para repositórios vinculados e estejam disponíveis somente na região Oeste dos EUA (Oregon). Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

9. Para ver as alterações de código na solicitação pull, selecione Alterações. Em Arquivos alterados, você pode ver rapidamente quantos arquivos têm alterações na solicitação pull e quais arquivos na solicitação pull têm comentários sobre eles. O número de comentários mostrados ao lado de uma pasta indica o número de comentários nos arquivos dessa pasta. Expanda a pasta para ver o número de comentários de cada arquivo na pasta. Você também pode ver os comentários deixados em linhas de código específicas.

 Note

Nem todas as alterações em uma solicitação pull podem ser exibidas no console. Por exemplo, você não pode visualizar os submódulos do Git no console, portanto, não pode ver as diferenças em um submódulo em uma solicitação pull. Algumas diferenças podem ser muito grandes para serem exibidas. Para obter mais informações, consulte [Cotas para repositórios de origem em CodeCatalyst](#) e [Visualizar um arquivo](#).

10. Para visualizar os relatórios de qualidade dessa solicitação pull, selecione Relatórios.

 Note

Um fluxo de trabalho deve ser configurado para gerar relatórios para que eles apareçam nas solicitações pull. Para obter mais informações, consulte [Teste com fluxos de trabalho](#).

## Gerenciamento de requisitos para mesclar uma solicitação pull com regras de aprovação

Ao criar uma solicitação pull, você pode optar por adicionar revisores obrigatórios ou opcionais a essa solicitação pull individual. No entanto, você também pode criar requisitos que todas as solicitações pull devem atender ao mesclar com uma ramificação de destino específica. Esses requisitos são chamados de regras de aprovação. As regras de aprovação são configuradas para ramificações em um repositório. Quando você cria uma solicitação pull cuja ramificação de destino tem uma regra de aprovação configurada para ela, os requisitos dessa regra devem ser atendidos, além das aprovações de todos os revisores necessários, antes que você possa mesclar a solicitação pull com essa ramificação. A criação de regras de aprovação pode ajudá-lo a manter os padrões de qualidade para mesclagens em ramificações, como sua ramificação padrão.

As regras de aprovação aplicadas à ramificação padrão do seu repositório de origem se comportarão de maneira um pouco diferente das regras de aprovação aplicadas a outras ramificações. Qualquer regra aplicada à ramificação padrão será aplicada automaticamente a qualquer ramificação que você especificar como ramificação padrão. A ramificação anteriormente definida como a ramificação padrão ainda manterá as regras aplicadas a ela.

Ao criar regras de aprovação, considere como essa regra será cumprida pelos usuários do seu projeto no presente e no futuro. Por exemplo, se você tem seis usuários em seu projeto e cria uma regra de aprovação que exige cinco aprovações antes que ela seja mesclada na ramificação de destino, você efetivamente criou uma regra que exige que todos, exceto a pessoa que criou as solicitações pull, aprovem essa solicitação pull antes que ela possa ser mesclada.

 Note

Você deve ter a função de administrador do projeto para criar e gerenciar regras de aprovação em CodeCatalyst projetos. Você não pode criar regras de aprovação para repositórios vinculados.

Você não pode excluir as regras de aprovação, mas pode atualizá-las para que não exijam nenhuma aprovação, o que efetivamente remove a regra.

Como visualizar e editar regras de aprovação para ramificações de destino para solicitações pull

1. Acesse o projeto no qual reside seu repositório.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja visualizar regras de aprovação.

3. Na página de visão geral do repositório, selecione Ramificações.
4. Na coluna Regras de aprovação, selecione Visualizar para ver o status de quaisquer regras para cada ramificação do repositório.

Em Número mínimo de aprovações, o número corresponde ao número de aprovações necessárias antes que uma solicitação pull seja mesclada a essa ramificação.

5. Para criar ou alterar uma regra de aprovação, selecione Gerenciar configurações. Na página de configurações do repositório de origem, em Regras de aprovação, selecione Editar.

 Note

Você deve ter o perfil de administrador do projeto para editar as regras de aprovação.

6. Em Ramificação, selecione o nome da ramificação para a qual você deseja configurar uma regra de aprovação na lista suspensa. Em Número mínimo de aprovações, insira um número e selecione Salvar.

## Revisão de uma solicitação pull

Você pode usar o CodeCatalyst console da Amazon para analisar e comentar de forma colaborativa as alterações incluídas em uma pull request. Você pode adicionar comentários a linhas individuais de código na diferença entre as ramificações de origem e de destino ou na diferença entre as revisões da solicitação pull. Você pode optar por criar um resumo dos comentários deixados sobre as alterações de código na solicitação pull para entender rapidamente os comentários deixados por outros usuários. Você também pode optar por criar um Ambiente de Desenvolvimento para trabalhar com o código.

### Note

#### Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

### Tip

Você pode configurar quais eventos de solicitação pull receberá e-mails como parte do seu perfil. Para obter mais informações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

As solicitações pull mostram qual será a diferença entre a revisão da solicitação pull e a confirmação que foi a ponta da ramificação de destino quando você criou a solicitação pull. Isso é chamado de base de mesclagem. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

 Tip

Ao trabalhar no console, especialmente se você tiver uma solicitação pull aberta há algum tempo, considere atualizar seu navegador para garantir que tenha a revisão mais recente disponível para uma solicitação pull antes de começar a revisá-la.

Para revisar uma pull request no CodeCatalyst console

1. Navegue até o projeto.
2. Navegue até as solicitações pull realizando uma das seguintes etapas:
  - Se a solicitação pull estiver listada na página do projeto, selecione-a na lista.
  - Se a solicitação pull não estiver listada na página do projeto, escolha Exibir tudo. Use os filtros e classifique para encontrar a solicitação pull e, depois, escolha-a na lista.
  - No painel de navegação, selecione Código e Solicitações pull.
3. Escolha a solicitação pull que você deseja revisar na lista. Você pode filtrar a lista de solicitações pull digitando parte do nome na barra de filtro.
4. Em Visão geral, você pode revisar o nome e o título da solicitação pull. Você pode criar e visualizar os comentários deixados na própria solicitação pull. Você também pode visualizar os detalhes da solicitação pull, incluindo informações sobre execuções de fluxo de trabalho, problemas vinculados, revisores, o autor da solicitação pull e estratégias de mesclagem viáveis.

 Note

Os comentários deixados em linhas de código específicas aparecem em Alterações.

5. (Opcional) Para adicionar um comentário que se aplique a toda a solicitação pull, expanda Comentários na solicitação pull e selecione Criar comentário.
6. (Opcional) Para ver um resumo de todos os comentários deixados sobre as alterações nas revisões dessa solicitação pull, selecione Criar resumo de comentários.

 Note

Essa funcionalidade exige que os recursos de IA generativa estejam habilitados para o espaço e só estejam disponíveis na região Oeste dos EUA (Oregon). Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

7. Em Alterações, você pode ver as diferenças entre a ramificação de destino e a revisão mais recente da solicitação pull. Se houver mais de uma revisão, você poderá alterar quais revisões são comparadas na diferença entre elas. Para ter mais informações sobre revisões, consulte [Revisões](#).

 Tip

Em Arquivos alterados, você pode ver rapidamente quantos arquivos têm alterações na solicitação pull e quais arquivos na solicitação pull têm comentários sobre eles. O número de comentários mostrados ao lado de uma pasta indica o número de comentários nos arquivos dessa pasta. Expanda a pasta para ver o número de comentários de cada arquivo na pasta.

8. Para alterar a forma como as diferenças são exibidas, escolha entre Unificado e Dividido.
9. Para adicionar um comentário a uma linha na solicitação pull, vá até a linha na qual deseja comentar. Escolha o ícone de comentário que é exibido para essa linha, insira um comentário e escolha Salvar.
10. Para visualizar as alterações entre as revisões em uma solicitação pull ou entre suas ramificações de origem e destino, escolha entre as opções disponíveis em Comparação. Os comentários sobre as linhas nas revisões são preservados nessas revisões.
11. Se você configurou seu fluxo de trabalho para gerar um relatório de cobertura de código sobre gatilhos de solicitação pull, você pode ver as descobertas de cobertura de linha e ramificação na solicitação pull relevante. Para ocultar as descobertas de cobertura de código, selecione Ocultar cobertura de código. Para obter mais informações, consulte [Relatórios de cobertura de código](#).
12. Se quiser fazer alterações no código da solicitação pull, você pode criar um Ambiente de Desenvolvimento a partir da solicitação pull. Escolha Criar ambiente de desenvolvimento. Se desejar, adicione um nome para o Ambiente de Desenvolvimento ou edite a configuração e selecione Criar.

13. Em Relatórios, você pode visualizar os relatórios de qualidade nessa solicitação pull. Se houver mais de uma revisão, você poderá alterar quais revisões são comparadas na diferença entre elas. Você pode filtrar os relatórios por nome, status, fluxo de trabalho, ação e tipo.

 Note

Um fluxo de trabalho deve ser configurado para gerar relatórios para que eles apareçam nas solicitações pull. Para obter mais informações, consulte [Configuração de relatórios de qualidade em uma ação](#).

14. Para ver um relatório específico, escolha-o na lista. Para obter mais informações, consulte [Teste com fluxos de trabalho](#).
15. Se você estiver listado como revisor dessa solicitação pull e quiser aprovar as alterações, verifique se está visualizando a revisão mais recente e selecione Aprovar.

 Note

Todos os revisores obrigatórios devem aprovar uma solicitação pull antes que ela possa ser mesclada.

## Atualizar uma solicitação pull

Você pode facilitar a revisão do código por outros membros do projeto atualizando a solicitação pull. Você pode atualizar uma solicitação pull para alterar os revisores, os links para problemas, o título da solicitação pull ou a descrição. Por exemplo, você talvez queira alterar os revisores obrigatórios de uma solicitação pull para remover alguém que está de férias e adicionar outra pessoa. Você também pode atualizar uma solicitação pull com mais alterações no código enviando confirmações para a ramificação de origem de uma solicitação pull aberta. Cada envio para a ramificação de origem de uma pull request no repositório de CodeCatalyst origem cria uma revisão. Os membros do projeto podem ver as diferenças entre as revisões em uma solicitação pull.

Para atualizar os revisores de uma solicitação pull

1. Navegue até o projeto em que deseja atualizar os revisores de uma solicitação pull.
2. Na página do projeto, em Abrir solicitações pull, escolha a solicitação pull em que você deseja atualizar os revisores. Como alternativa, no painel de navegação, selecione Código, selecione Solicitações pull e, depois, escolha a solicitação pull que você deseja atualizar.



3. (Opcional) Em Visão geral, na área Detalhes da solicitação pull, escolha o sinal de adição para adicionar revisores obrigatórios ou opcionais. Escolha o X ao lado de um revisor para removê-lo como revisor opcional ou obrigatório.
4. (Opcional) Em Visão geral, na área Detalhes da solicitação pull, selecione Vincular problemas para vincular um problema à solicitação pull e, depois, escolha um problema na lista ou insira o ID. Para desvincular um problema, escolha o ícone de desvinculação ao lado do problema que você deseja desvincular.

Para atualizar arquivos e códigos na ramificação de origem de uma solicitação pull

1. Para atualizar vários arquivos, [crie um Ambiente de Desenvolvimento](#) ou clone o repositório e a ramificação de origem e use um cliente Git ou um Ambiente de Desenvolvimento integrado (IDE) para fazer alterações nos arquivos na ramificação de origem. Confirme e envie as alterações para a ramificação de CodeCatalyst origem no repositório de origem para atualizar automaticamente a pull request com as alterações. Para obter mais informações, consulte [Clonar um repositório de origem](#) e [Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst](#).
2. Para atualizar um arquivo individual em uma ramificação de origem, você pode usar um cliente Git ou IDE da mesma forma que faria com vários arquivos. Você também pode editá-lo diretamente no CodeCatalyst console. Para obter mais informações, consulte [Edição de um arquivo](#).

Para atualizar o título e a descrição de uma solicitação pull

1. Navegue até o projeto em que deseja atualizar o título ou a descrição de uma solicitação pull.
2. A página do projeto exibe solicitações pull abertas, incluindo informações sobre quem criou a solicitação pull, qual repositório contém as ramificações da solicitação pull e quando a solicitação pull foi criada. Você pode filtrar a visualização da solicitação pull aberta por repositório de origem. Selecione na lista a solicitação pull que você deseja alterar.
3. Para ver todas as solicitações pull, selecione Exibir tudo. Como alternativa, no painel de navegação, selecione Código e Solicitações pull. Use a caixa de filtro ou as funções de classificação para localizar a solicitação pull que você deseja alterar e escolha essa solicitação.
4. Em Visão geral, selecione Editar.
5. Altere o título ou a descrição e selecione Salvar.

## Mesclar uma solicitação pull

Depois que seu código for revisado e todos os revisores necessários o aprovarem, você poderá mesclar uma pull request no CodeCatalyst console usando uma estratégia de mesclagem compatível, como o fast-forward. Nem todas as estratégias de mesclagem suportadas no CodeCatalyst console estão disponíveis como opções para todas as pull requests. CodeCatalyst avalia a mesclagem e só permite que você escolha entre as estratégias de mesclagem que estão disponíveis no console e são capazes de mesclar a ramificação de origem com a ramificação de destino. Você também pode mesclar uma solicitação pull com sua escolha de estratégias de mesclagem do Git executando o comando `git merge` no computador local ou em um Ambiente de Desenvolvimento para mesclar a ramificação de origem com a ramificação de destino. Em seguida, você pode enviar essas alterações na ramificação de destino para o repositório de origem em CodeCatalyst.

### Note

Mesclar a ramificação e enviar as alterações no Git não fecha automaticamente a solicitação pull.

Se você tiver o perfil de administrador do projeto, também poderá optar por mesclar uma solicitação pull que ainda não satisfaz todos os requisitos de aprovações e regras de aprovação.

### Mescla de uma solicitação pull (console)

Você pode mesclar uma pull request no CodeCatalyst console se não houver conflitos de mesclagem entre as ramificações de origem e de destino e se todos os revisores necessários tiverem aprovado a pull request. Se houver conflitos ou se a mesclagem não puder ser concluída, o botão de mesclagem ficará inativo e um rótulo Não mesclável será exibido. Nesse caso, você deve ter a aprovação de todos os aprovadores necessários, resolver conflitos localmente, se necessário, e enviar essas alterações antes de poder mesclar. Mesclar uma solicitação pull enviará automaticamente um e-mail para o criador da solicitação pull, bem como para quaisquer revisores obrigatórios ou opcionais. Isso não fechará ou alterará automaticamente o status de nenhum problema vinculado à solicitação pull.

 Tip

Você pode configurar quais eventos de solicitação pull receberá e-mails como parte do seu perfil. Para obter mais informações, consulte [Envio de notificações do Slack e por e-mail do CodeCatalyst](#).

## Para mesclar uma solicitação pull

1. Navegue até o projeto onde você deseja mesclar uma solicitação pull.
2. Na página do projeto, em Abrir solicitações pull, escolha a solicitação pull que você deseja mesclar. Se você não vir a solicitação pull, selecione Visualizar todas as solicitações pull e, depois, escolha-a na lista. Como alternativa, no painel de navegação, selecione Código, selecione Solicitações pull e, depois, escolha a solicitação pull que você deseja mesclar. Escolha Merge (Mesclar).
3. Selecione entre as estratégias de mesclagem disponíveis para a solicitação pull. Também é possível selecionar ou desmarcar a opção de excluir a ramificação de origem depois de mesclar a solicitação pull e, depois, selecionar Mesclar.

 Note

Se o botão Mesclar estiver inativo ou você ver o rótulo Não mesclável, os revisores obrigatórios ainda não aprovaram a pull request ou a pull request não pode ser mesclada no console. CodeCatalyst Um revisor que não aprovou uma solicitação pull é indicado por um ícone de relógio na área Detalhes da solicitação pull em Visão geral. Se todos os revisores obrigatórios aprovaram a solicitação pull, mas o botão Mesclar ainda estiver inativo, você pode ter um conflito de mesclagem. Escolha o rótulo sublinhado Não mesclável para ver mais detalhes sobre por que a solicitação pull não pode ser mesclada. Você pode resolver conflitos de mesclagem para a ramificação de destino em um ambiente de desenvolvimento ou no CodeCatalyst console e, em seguida, mesclar a pull request, ou pode resolver conflitos e mesclar localmente e, em seguida, enviar a confirmação que contém a mesclagem para a ramificação de origem em. CodeCatalyst Para ter mais informações, consulte [Mescla de uma solicitação pull \(Git\)](#) e a documentação do Git.

## Ignorar requisitos de mescla

Se você tiver o perfil de Administrador do projeto, poderá optar por mesclar uma solicitação pull que ainda não satisfaz todos os requisitos das aprovações e regras de aprovação necessárias. Isso se chama ignorar os requisitos de uma solicitação pull. Você pode optar por fazer isso se um revisor obrigatório não estiver disponível ou se surgir uma necessidade urgente de mesclar uma solicitação pull específica em uma ramificação que tenha regras de aprovação que não possam ser atendidas rapidamente.

Para mesclar uma solicitação pull

1. Na solicitação pull em que você deseja ignorar os requisitos e mesclar, escolha a seta suspensa ao lado do botão Mesclar. Selecione Ignorar requisitos de aprovação.
2. Em Motivo para ignorar, forneça detalhes do motivo pelo qual você está mesclando essa solicitação pull sem que ela atenda às regras de aprovação e aos requisitos do revisor obrigatório. Embora seja opcional, isso é altamente recomendado.
3. Se desejar, escolha uma estratégia de mesclagem ou aceite o padrão. Você também pode optar por atualizar a mensagem de confirmação gerada automaticamente com mais detalhes.
4. Selecione ou desmarque a opção para excluir a ramificação de origem na mesclagem. Recomendamos que você retenha a ramificação de origem ao ignorar os requisitos para mesclar uma solicitação pull até ter a chance de revisar a decisão com outros membros da equipe.
5. Escolha Merge (Mesclar).

## Mescla de uma solicitação pull (Git)

O Git suporta muitas opções para mesclar e gerenciar ramificações. Os comandos a seguir são algumas das opções que você pode usar. Para ter mais informações, consulte a documentação disponível no [site do Git](#). Depois de mesclar e enviar as alterações, feche manualmente a solicitação pull. Para obter mais informações, consulte [Encerrar uma solicitação pull](#).

Comandos comuns do Git para mesclar ramificações

Mescla alterações da ramificação de origem no repositório local à ramificação de destino no repositório local.

```
git checkout destination-branch-name
```

```
git merge source-branch-name
```

Mescla a ramificação de origem com a ramificação de destino, especificando uma mesclagem fast-forward. Isso mescla as ramificações e move o ponteiro da ramificação de destino até a ponta da ramificação de origem.

```
git checkout destination-branch-name
```

```
git merge --ff-only source-branch-name
```

Mescla a ramificação de origem com a ramificação de destino, especificando uma mesclagem squash. Isso combina todas as confirmações da ramificação de origem em uma única confirmação de mesclagem na ramificação de destino.

```
git checkout destination-branch-name
```

```
git merge --squash source-branch-name
```

Mescla a ramificação de origem com a ramificação de destino, especificando uma mesclagem tripla. Isso cria uma confirmação de mesclagem e adiciona as confirmações individuais da ramificação de origem à ramificação de destino.

```
git checkout destination-branch-name
```

```
git merge --no-ff source-branch-name
```

Exclui a ramificação de origem no repositório local. Isso é útil para limpar o repositório local depois de mesclar com a ramificação de destino e enviar as alterações para o repositório de origem.

```
git branch -d source-branch-name
```

Exclui a ramificação de origem no repositório remoto (o repositório de origem em CodeCatalyst) usando o apelido especificado do repositório local para o repositório remoto. (Observe o uso de dois pontos (:).) Como alternativa, especifique `--delete` como parte do comando.

```
git push remote-name :source-branch-name
```

```
git push remote-name --delete source-branch-name
```

## Encerrar uma solicitação pull

Você pode marcar uma solicitação pull como Fechada. Isso não mescla a solicitação pull, mas pode ajudar você a determinar quais solicitações pull exigem ação e quais não são mais relevantes. Recomendamos fechar uma solicitação pull se você não planeja mais mesclar essas alterações ou se as alterações foram mescladas por outra solicitação pull.

Fechar uma solicitação pull enviará automaticamente um e-mail para o criador da solicitação pull, bem como para quaisquer revisores obrigatórios ou opcionais. Isso não alterará automaticamente o status de nenhum problema vinculado à solicitação pull.

### Note

Você não pode reabrir uma solicitação pull após ela ter sido fechada.

Para fechar uma solicitação pull

1. Navegue até o projeto onde você deseja fechar uma solicitação pull.
2. Na página do projeto, as solicitações pull abertas são exibidas. Escolha a solicitação pull que você deseja fechar.
3. Escolha Fechar.
4. Revise as informações e selecione Fechar solicitação pull.

## Entendendo as mudanças no código-fonte com confirmações na Amazon CodeCatalyst

As confirmações são snapshots dos conteúdos e das alterações feitas nos conteúdos do repositório. Sempre que um usuário confirma e envia uma alteração para uma ramificação, essa informação é salva. As informações de confirmação do Git incluem o autor da confirmação, a pessoa que confirmou a alteração, a data e a hora, e as alterações feitas. Informações semelhantes são incluídas automaticamente quando você cria ou edita um arquivo no CodeCatalyst console da Amazon, mas o nome do autor é seu nome de CodeCatalyst usuário. Você também pode adicionar tags Git a confirmações para ajudar a identificar confirmações específicas.

Na Amazon CodeCatalyst, você pode:

- Visualizar uma lista de confirmações para uma ramificação.
- Visualizar confirmações individuais, incluindo as alterações feitas em uma confirmação em comparação com sua principal ou principais.

Você também pode visualizar arquivos e pastas. Para obter mais informações, consulte [Gerenciando arquivos de código-fonte na Amazon CodeCatalyst](#).

## Tópicos

- [Visualização de confirmações em uma ramificação](#)
- [Alterando a forma como os commits são exibidos \(CodeCatalystconsole\)](#)

## Visualização de confirmações em uma ramificação

Você pode ver o histórico das alterações feitas em uma ramificação revisando os commits da ramificação no CodeCatalyst console. Isso ajuda você a entender quem fez alterações na ramificação e quando. Você também pode revisar as alterações feitas em uma confirmação específica.

### Tip

Você também pode visualizar o histórico de confirmações que fizeram alterações em um arquivo específico. Para ter mais informações, consulte [Visualizar um arquivo](#).

Você também pode visualizar as confirmações usando seu cliente Git. Para ter mais informações, consulte a documentação do Git.

### Como visualizar confirmações (console)

1. Navegue até o tópico que contém o repositório de origem em que deseja exibir confirmações.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja visualizar as confirmações de uma ramificação.

3. A ramificação padrão do repositório é exibida, incluindo informações sobre a confirmação mais recente para a ramificação. Selecione Confirmações. Como alternativa, selecione Mais e, depois, selecione Visualizar confirmações.
4. Para visualizar as confirmações de uma ramificação diferente, escolha o seletor de ramificação e, depois, escolha o nome da ramificação.
5. Para visualizar detalhes sobre uma determinada confirmação, selecione seu título em Título de confirmação. Os detalhes da confirmação são exibidos, incluindo informações sobre a confirmação principal e as alterações feitas no código ao comparar a confirmação principal com a confirmação especificada.

 Tip

Se uma confirmação tiver mais de uma principal, você poderá escolher em qual confirmação principal visualizar as informações e exibir as alterações selecionando o ícone suspenso ao lado do ID da confirmação principal.

## Alterando a forma como os commits são exibidos (CodeCatalystconsole)

Você pode alterar quais informações são exibidas na visualização Confirmações. Você pode optar por ocultar ou exibir colunas como autor e ID de confirmação.

Como alterar a forma como as confirmações são exibidas (console)

1. Navegue até o tópico que contém o repositório de origem em que deseja exibir confirmações.
2. Selecione o nome do repositório na lista de repositórios de origem do projeto. No painel de navegação, também é possível selecionar Código e escolher Repositórios de origem.

Selecione o repositório onde você deseja alterar a forma como visualiza as confirmações.

3. A ramificação padrão do repositório é exibida, incluindo informações sobre a confirmação mais recente para a ramificação. Selecione Confirmações.
4. Escolha o ícone de engrenagem.
5. Em Preferências, selecione o número de confirmações a serem exibidas e escolha se deseja exibir informações sobre o autor da confirmação, a data da confirmação e o ID da confirmação.



**Note**

Você não pode ocultar o título da confirmação na exibição das informações.

- Depois de fazer suas alterações, selecione Salvar para salvá-las ou Cancelar para descartá-las.

## Cotas para repositórios de origem em CodeCatalyst

A tabela a seguir descreve as cotas e os limites dos repositórios de origem na Amazon.

CodeCatalyst Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

| Recurso                                | Informações                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomes de ramificação                   | <p>Qualquer combinação de caracteres permitidos entre 1 e 256 caracteres de comprimento deve ser única dentro de um repositório. Nomes de ramificação não podem:</p> <ul style="list-style-type: none"> <li>começar nem terminar com uma barra (/) ou um ponto (.)</li> <li>consistir em único caractere @</li> <li>conter dois ou mais pontos consecutivos (.), barras (//) ou a seguinte combinação de caracteres: @{</li> <li>conter espaços ou qualquer um dos seguintes caracteres: ? ^ * [ \ ~ :</li> </ul> <p>Os nomes de ramificação são referências. Muitas limitações nos nomes de ramificação são baseadas no padrão de referência do Git. Para obter mais informações, consulte <a href="#">Git Internals e. git-check-ref-format</a></p> |
| Comentários sobre uma solicitação pull | Máximo de 1.000 em uma solicitação pull.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Recurso                                            | Informações                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Confirmar mensagem                                 | Máximo de 1024 caracteres.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Caminhos do arquivo                                | <p>Qualquer combinação dos caracteres permitidos entre 1 e 4,096 caracteres. Os caminhos de arquivo devem ser um nome exclusivo que especifica o arquivo e o local exato dele. Os caminhos de arquivo não podem exceder 20 diretórios de extensão. Além disso, os caminhos de arquivo não podem:</p> <ul style="list-style-type: none"><li>• conter sequências de caracteres vazias</li><li>• ser um caminho de arquivo relativo</li><li>• incluir qualquer uma das seguintes combinações:<br/><br/>    <code>/./</code><br/><br/>    <code>/. ./</code><br/><br/>    <code>//</code></li><li>• terminar com barra no final ou barra invertida</li></ul> <p>Os nomes de arquivo e caminhos devem ser totalmente qualificados. O nome e o caminho de um arquivo no seu computador local devem seguir os padrões do sistema operacional em questão. Ao especificar o caminho de um arquivo em um repositório, use os padrões do Amazon Linux.</p> |
| Tamanho do arquivo                                 | Máximo de 6 MB para qualquer arquivo individual ao usar o CodeCatalyst console.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Tamanho do arquivo visível no console CodeCatalyst | Máximo de 6 MB para qualquer arquivo individual ao usar o CodeCatalyst console.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Recurso                                                                      | Informações                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tamanho do blob do Git                                                       | <p>Máximo de 2 GB.</p> <div><p> <b>Note</b></p><p>Não há limite no número ou tamanho total de todos os arquivos em uma única confirmação, desde que os metadados não ultrapassem 6 MB e um único blob não exceda 2 GB. No entanto, como prática recomendada, pense em fazer várias confirmações menores em vez de uma confirmação grande.</p></div>                                                                                                                      |
| Metadados de uma confirmação                                                 | <p>Máximo de 6 MB para os <a href="#">metadados combinados de uma confirmação</a> (por exemplo, a combinação de informações do autor, de data, de lista de confirmação principal e de mensagens de confirmação).</p> <div><p> <b>Note</b></p><p>Não há limite no número ou tamanho total de todos os arquivos em uma única confirmação, desde que os dados não ultrapassem 20 MB, um arquivo individual não ultrapasse 6 MB e um único blob não exceda 2 GB.</p></div> |
| Número de CodeCatalyst problemas que podem ser vinculados a uma pull request | 50                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Número de problemas do Jira que podem ser vinculados a uma solicitação pull  | 50                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| Recurso                                          | Informações                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Número de solicitações pull abertas em um espaço | Máximo de 1.000 para um CodeCatalyst espaço da Amazon.                                                                                                                                                                                                                                                                                                                                                                    |
| Número de solicitações pull totais em um espaço  | Máximo de 10.000 para um CodeCatalyst espaço da Amazon.                                                                                                                                                                                                                                                                                                                                                                   |
| Número de referências em um único push           | Máximo de 4.000, incluindo criação, exclusão e atualização. Não há limites para o número total de referências no repositório.                                                                                                                                                                                                                                                                                             |
| Número de repositórios em um espaço              | Máximo de 5.000 para um CodeCatalyst espaço da Amazon.                                                                                                                                                                                                                                                                                                                                                                    |
| Descrições de repositório                        | Qualquer combinação de caracteres entre 0 e 1.000 caracteres. As descrições de repositório são opcionais.                                                                                                                                                                                                                                                                                                                 |
| Nomes de repositório                             | Os nomes dos repositórios devem ser exclusivos em um projeto. Eles podem ter qualquer combinação de letras, números, pontos, sublinhados e traços entre 1 e 100 caracteres. Os nomes não diferenciam maiúsculas de minúsculas. Os nomes de repositórios não podem terminar em .git, não podem conter espaços e não podem conter nenhum dos seguintes caracteres: ! ? @ # \$ % ^ & * ( ) + = { } [ ]   \ / > < ~ ` ' " ; : |
| Tamanho do repositório                           | Os tamanhos dos repositórios são afetados pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte <a href="#">Preços</a> e <a href="#">Solucionar problemas com repositórios de origem</a> .                                                                                                                                                                                           |

| Recurso                               | Informações                                                                                                                                                             |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Revisores de uma solicitação pull     | Máximo de 100 revisores no total (opcional ou obrigatório) para uma solicitação pull.                                                                                   |
| Resumos escritos de solicitações pull | O número máximo de resumos escritos para solicitações pull depende do nível de faturamento do seu espaço. Para obter mais informações, consulte <a href="#">Preço</a> . |

# Escreva e modifique o código com Dev Environments em CodeCatalyst

Ambientes de desenvolvimento são ambientes de desenvolvimento baseados em nuvem. Na Amazon CodeCatalyst, você usa Dev Environments para trabalhar no código armazenado nos repositórios de origem do seu projeto. Ao criar um Ambiente de Desenvolvimento, você tem várias opções:

- Crie um ambiente de desenvolvimento específico do projeto CodeCatalyst para trabalhar no código com um ambiente de desenvolvimento integrado (IDE) compatível.
- Crie um Ambiente de Desenvolvimento vazio, clone o código de um repositório de origem e trabalhe nesse código com um IDE compatível.
- Criar um Ambiente de Desenvolvimento em um IDE e clonar um repositório de origem no Ambiente de Desenvolvimento

Um devfile é um arquivo YAML de padrão aberto que padroniza seus ambientes de desenvolvimento. Em outras palavras, esse arquivo codifica as ferramentas de desenvolvimento necessárias para seu Ambiente de Desenvolvimento. Como resultado, você pode configurar rapidamente um Ambiente de Desenvolvimento, alternar entre projetos e replicar a configuração do Ambiente de Desenvolvimento entre os membros da equipe. Os Ambientes de Desenvolvimento minimizam o tempo gasto criando e mantendo um Ambiente de Desenvolvimento local, porque eles usam um devfile que configura todas as ferramentas necessárias para codificar, testar e depurar em um determinado projeto.

## Important

Os ambientes de desenvolvimento podem executar scripts por meio de devfiles que têm acesso às suas CodeCatalyst credenciais. Inspecione o repositório antes de abrir recursos não confiáveis.

As ferramentas do projeto e as bibliotecas de aplicações incluídas no Ambiente de Desenvolvimento são definidas pelo devfile no repositório de origem do projeto. Se você não tiver um devfile no seu repositório de origem, aplica CodeCatalyst automaticamente um devfile padrão. Esse devfile padrão inclui ferramentas para as linguagens e estruturas de programação usadas com mais

frequência. Se seu projeto foi criado usando um blueprint, um devfile será criado automaticamente pelo. CodeCatalyst Para ter mais informações sobre o devfile, consulte <https://devfile.io>.

Depois de criar um Ambiente de Desenvolvimento, somente você pode acessá-lo. Em seu Ambiente de Desenvolvimento, você pode visualizar e trabalhar no código do seu repositório de origem em um IDE compatível.

Por padrão, um Ambiente de Desenvolvimento é configurado para ter um processador de 2 núcleos, 4 GB de RAM e 16 GB de armazenamento persistente. Se você tiver permissões de administrador do espaço, poderá alterar o nível de cobrança do seu espaço para usar diferentes opções de configuração do Ambiente de Desenvolvimento e gerenciar os limites de computação e armazenamento.

## Tópicos

- [Criar um Ambiente de Desenvolvimento](#)
- [Interromper um Ambiente de Desenvolvimento](#)
- [Retomar um Ambiente de Desenvolvimento](#)
- [Edição de um Ambiente de Desenvolvimento](#)
- [Excluir um Ambiente de Desenvolvimento](#)
- [Estabelecer conexão com um Ambiente de Desenvolvimento usando SSH](#)
- [Configuração de um devfile para um Ambiente de Desenvolvimento](#)
- [Associar uma conexão VPC a um Ambiente de Desenvolvimento](#)
- [Cotas para ambientes de desenvolvimento em CodeCatalyst](#)

# Criar um Ambiente de Desenvolvimento

Você pode criar um Ambiente de Desenvolvimento de várias maneiras:

- Crie um ambiente de desenvolvimento CodeCatalyst com um repositório de CodeCatalyst origem ou um repositório de [origem vinculado nas páginas](#) Visão geral, Ambientes de desenvolvimento ou repositórios de origem
- Crie um ambiente de desenvolvimento vazio CodeCatalyst que não esteja conectado a um repositório de origem na página Ambientes de desenvolvimento
- Criar um Ambiente de Desenvolvimento no IDE de sua escolha e clonar qualquer repositório de origem no Ambiente de Desenvolvimento

 Important

Os ambientes de desenvolvimento não estão disponíveis para usuários em espaços em que o Active Directory é usado como provedor de identidade. Para obter mais informações, consulte [Não consigo criar um Ambiente de Desenvolvimento quando estou conectado ao CodeCatalyst usando uma conta de login único](#).

Você pode criar um Ambiente de Desenvolvimento por ramificação de um repositório. Um projeto pode ter vários repositórios. Os ambientes de desenvolvimento que você cria só podem ser gerenciados com sua CodeCatalyst conta, mas você pode abrir o ambiente de desenvolvimento e trabalhar nele com qualquer um dos ambientes compatíveis IDEs. Você deve ter o AWS Toolkit instalado para usar os ambientes de desenvolvimento no IDE. Para obter mais informações, consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#). Por padrão, os Ambientes de Desenvolvimento são criados com um processador de 2 núcleos, 4 GB de RAM e 16 GB de armazenamento persistente.

 Note

Se você criou um Ambiente de Desenvolvimento associado a um repositório de origem, a coluna Recurso sempre mostra a ramificação especificada ao criar esse Ambiente de Desenvolvimento. Isso se aplica mesmo se você criar outra ramificação, alternar para outra ramificação no Ambiente de Desenvolvimento ou clonar um repositório adicional. Se você criou um Ambiente de Desenvolvimento vazio, a coluna Recurso ficará em branco.

## Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento

Você pode usar Dev Environments com os seguintes ambientes de desenvolvimento integrados compatíveis (IDEs):

- [AWS Cloud9](#)
- [JetBrains IDEs](#)
  - [IntelliJ IDEA Ultimate](#)
  - [GoLand](#)
  - [PyCharmProfessional](#)



- [Visual Studio Code](#)

## Criando um ambiente de desenvolvimento em CodeCatalyst

Para começar a trabalhar com o Dev Environment in CodeCatalyst, autentique-se e faça login com seu [ID do AWS Builder](#) ou [SSO](#).

Como criar um Ambiente de Desenvolvimento com uma nova ramificação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja criar um Ambiente de Desenvolvimento.
3. No painel de navegação, execute uma das seguintes opções:
  - Selecione Visão geral e navegue até a seção Meus ambientes de desenvolvimento.
  - Selecione Código e, depois, escolha Ambientes de Desenvolvimento.
  - Selecione Código, Repositórios de origem e selecione o repositório para o qual você deseja criar um Ambiente de Desenvolvimento.
4. Escolha Criar ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Escolha Clonar um repositório.
7. Execute um destes procedimentos:
  - a. Escolha o repositório a ser clonado, selecione Trabalhar na ramificação existente e, depois, escolha uma ramificação no menu suspenso Ramificação existente.

### Note

Se você escolher um repositório de terceiros, deverá trabalhar em uma ramificação existente.

- b. Escolha o repositório a ser clonado, selecione Trabalhar em uma nova ramificação, insira um nome de ramificação no campo Nome da ramificação e escolha uma ramificação a partir da qual criar a nova ramificação no menu Criar ramificação.

 Note

Se você criar um Ambiente de Desenvolvimento na página Repositórios de origem ou em um repositório de origem específico, não precisará escolher um repositório. O Ambiente de Desenvolvimento será criado no repositório de origem que você escolheu na página Repositórios de origem.

8. (Opcional) Em Alias - opcional, insira um alias para o Ambiente de Desenvolvimento.
9. (Opcional) Escolha o botão de edição da configuração do Ambiente de Desenvolvimento para editar a configuração de computação, armazenamento ou tempo limite do Ambiente de Desenvolvimento.
10. (Opcional) Em Amazon Virtual Private Cloud (Amazon VPC) - opcional, selecione uma conexão VPC que você gostaria de associar ao seu Ambiente de Desenvolvimento no menu suspenso.

Se uma VPC padrão for definida para seu espaço, seus ambientes de desenvolvimento serão executados conectados a essa VPC. Você pode substituir isso associando uma conexão VPC diferente. Além disso, observe que os ambientes de desenvolvimento conectados à VPC não são compatíveis com AWS Toolkit.

Se a conexão VPC que você deseja usar não estiver listada, talvez seja porque ela inclui uma Conta da AWS conexão que não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

 Note

Quando você cria um ambiente de desenvolvimento com uma conexão VPC, uma nova interface de rede é criada dentro da VPC. CodeCatalyst interage com essa interface usando a função VPC associada. Além disso, certifique-se de que seu bloco IPv4 CIDR não esteja configurado para o intervalo de endereços 172.16.0.0/12 IP.

11. Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento.

## Para criar um Ambiente de Desenvolvimento vazio

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja criar um Ambiente de Desenvolvimento.
3. No painel de navegação, execute uma das seguintes opções:
  - Selecione Visão geral e navegue até a seção Meus ambientes de desenvolvimento.
  - Selecione Código e, depois, escolha Ambientes de Desenvolvimento.
4. Escolha Criar ambiente de desenvolvimento.
5. Selecione um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
6. Selecione Criar um ambiente de desenvolvimento vazio.
7. (Opcional) Em Alias - opcional, insira um alias para o Ambiente de Desenvolvimento.
8. (Opcional) Escolha o botão de edição da configuração do Ambiente de Desenvolvimento para editar a configuração de computação, armazenamento ou tempo limite do Ambiente de Desenvolvimento.
9. (Opcional) Em Amazon Virtual Private Cloud (Amazon VPC) - opcional, selecione uma conexão VPC que você gostaria de associar ao seu Ambiente de Desenvolvimento no menu suspenso.

Se uma VPC padrão for definida para seu espaço, seus ambientes de desenvolvimento serão executados conectados a essa VPC. Você pode substituir isso associando uma conexão VPC diferente. Além disso, observe que os ambientes de desenvolvimento conectados à VPC não são compatíveis com AWS Toolkit.

Se a conexão VPC que você deseja usar não estiver listada, talvez seja porque ela inclui uma Conta da AWS conexão que não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

### Note

Quando você cria um ambiente de desenvolvimento com uma conexão VPC, uma nova interface de rede é criada dentro da VPC. CodeCatalyst interage com essa interface usando a função VPC associada. Além disso, certifique-se de que seu bloco IPv4 CIDR não esteja configurado para o intervalo de endereços 172.16.0.0/12 IP.

10. Escolha Criar. Enquanto seu Ambiente de Desenvolvimento estiver sendo criado, a coluna de status do Ambiente de Desenvolvimento exibirá Iniciando e a coluna de status exibirá Executando após a criação do Ambiente de Desenvolvimento.

 Note

A criação e a abertura de um Ambiente de Desenvolvimento pela primeira vez podem levar de um a dois minutos.

 Note

Depois que o Ambiente de Desenvolvimento for aberto no IDE, talvez seja necessário alterar o diretório para o repositório de origem antes de confirmar e enviar as alterações em seu código.

## Criação de um Ambiente de Desenvolvimento em um IDE

Você pode usar ambientes de desenvolvimento para trabalhar rapidamente no código armazenado nos repositórios de origem do seu projeto. Ambientes de desenvolvimento aumentam sua velocidade de desenvolvimento porque você pode começar a programar imediatamente em um Ambiente de Desenvolvimento em nuvem totalmente funcional e específico do projeto com um Ambiente de Desenvolvimento integrado (IDE) compatível.

Para obter informações sobre como trabalhar com CodeCatalyst um IDE, consulte a documentação a seguir.

- [Amazon CodeCatalyst para JetBrains IDEs](#)
- [Amazon CodeCatalyst para VS Code](#)
- [Amazon CodeCatalyst para AWS Cloud9](#)

## Interromper um Ambiente de Desenvolvimento

O diretório `/projects` de um Ambiente de Desenvolvimento armazena os arquivos que são extraídos do repositório de origem e do `devfile` que é usado para configurar o Ambiente de Desenvolvimento. O diretório `/home`, que fica vazio na criação do Ambiente de Desenvolvimento,

armazena os arquivos que você cria enquanto usa seu Ambiente de Desenvolvimento. Tudo nos diretórios `/projects` e `/home` de um Ambiente de Desenvolvimento é armazenado de forma persistente, então você pode parar de trabalhar em um Ambiente de Desenvolvimento se precisar mudar para outro Ambiente de Desenvolvimento, repositório ou projeto.

 Warning

Um ambiente de desenvolvimento não atingirá o tempo limite se alguma instância, incluindo navegadores da Web, shells remotos e IDEs, permanecer conectada. Portanto, feche todas as instâncias conectadas para evitar custos adicionais.

Um Ambiente de Desenvolvimento vai parar automaticamente se ficar inativo pelo tempo selecionado nos campos Tempo limite quando ele foi criado. Você pode interromper o Ambiente de Desenvolvimento antes que ele fique ocioso. Se você escolher Sem tempo limite ao criar seu Ambiente de Desenvolvimento, o Ambiente de Desenvolvimento não será interrompido automaticamente. Em vez disso, ele será executado continuamente.

 Warning

Se você interromper um Ambiente de Desenvolvimento associado a uma conexão VPC excluída, ele não poderá ser retomado.

Como interromper um Ambiente de Desenvolvimento na página Ambientes de desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja interromper um Ambiente de Desenvolvimento.
3. No painel de navegação, selecione Código.
4. Selecione Ambientes de desenvolvimento.
5. Marque o botão de seleção do Ambiente de Desenvolvimento que você deseja interromper.
6. No menu Ações, selecione Parar.

 Note

O uso da computação é cobrado somente enquanto o Ambiente de Desenvolvimento está em execução, mas o uso do armazenamento é cobrado por todo o tempo em que o Ambiente

de Desenvolvimento existe. Interrompa seu Ambiente de Desenvolvimento quando ele não estiver em uso para interromper o faturamento de computação.

## Retomar um Ambiente de Desenvolvimento

O diretório `/projects` de um Ambiente de Desenvolvimento armazena os arquivos que são extraídos do repositório de origem e do `devfile` que é usado para configurar o Ambiente de Desenvolvimento. O diretório `/home`, que fica vazio na criação do Ambiente de Desenvolvimento, armazena os arquivos que você cria enquanto usa seu Ambiente de Desenvolvimento. Tudo nos diretórios `/projects` e `/home` de um Ambiente de Desenvolvimento é armazenado de forma persistente, para que você possa parar de trabalhar nele se precisar mudar para outro Ambiente de Desenvolvimento, repositório ou projeto e continuar trabalhando nele posteriormente.

Um Ambiente de Desenvolvimento vai parar automaticamente se ficar inativo pelo tempo selecionado nos campos Tempo limite quando ele foi criado. Você deve fechar a guia do AWS Cloud9 navegador para que o Dev Environment fique ocioso.

### Note

O Ambiente de Desenvolvimento ainda está disponível e em execução mesmo se você excluir a ramificação com a qual o criou. Se você quiser continuar trabalhando em um Ambiente de Desenvolvimento do qual excluiu a ramificação, crie uma nova ramificação e envie suas alterações para ela.

Para retomar um Ambiente de Desenvolvimento na página de visão geral

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja retomar um Ambiente de Desenvolvimento e navegue até a seção Meus ambientes de desenvolvimento.
3. Selecione Retomar em (IDE).
  - Para JetBrains IDEs, escolha JetBrains Gateway-EAP quando solicitado a Escolher um aplicativo para abrir o link -gateway. JetBrains Quando solicitado, selecione Abrir link para confirmar.
  - Para o IDE do VS Code, escolha VS Code quando solicitado a Escolher uma aplicação para abrir o link do VS Code. Selecione Abrir link para confirmar.

Para retomar um Ambiente de Desenvolvimento no repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja retomar um Ambiente de Desenvolvimento.
3. No painel de navegação, selecione Código.
4. Selecione Repositórios de origem.
5. Escolha o repositório de origem que contém o Ambiente de Desenvolvimento que você deseja retomar.
6. Escolha o nome da ramificação para ver um menu suspenso de suas ramificações e, depois, escolha sua ramificação.
7. Selecione Retomar Ambiente de Desenvolvimento.
  - Para JetBrains IDEs, escolha Abrir link para confirmar quando solicitado a opção Permitir que este site abra o link JetBrains -gateway com JetBrains o Gateway? .
  - Para o IDE do VS Code, escolha Abrir link para confirmar quando solicitado a Permitir que este site abra o link do VS Code com o Visual Studio Code?.

Para retomar um Ambiente de Desenvolvimento na página Ambientes de desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja retomar um Ambiente de Desenvolvimento.
3. No painel de navegação, selecione Código.
4. Selecione Ambientes de desenvolvimento.
5. Na coluna IDE, selecione Retomar em (IDE) para o Ambiente de Desenvolvimento.
  - Para JetBrains IDEs, escolha Abrir link para confirmar quando solicitado a opção Permitir que este site abra o link JetBrains -gateway com JetBrains o Gateway? .
  - Para o IDE do VS Code, escolha Abrir link para confirmar quando solicitado a Permitir que este site abra o link do VS Code com o Visual Studio Code?.

 Note

A retomada de um Ambiente de Desenvolvimento pode levar alguns minutos.

# Edição de um Ambiente de Desenvolvimento

Enquanto seu IDE está em execução, você pode editar o Ambiente de Desenvolvimento. Se você editar o tempo limite de computação ou inatividade, seu Ambiente de Desenvolvimento será reiniciado depois que você salvar as alterações.

## Como editar um Ambiente de Desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja editar um Ambiente de Desenvolvimento.
3. No painel de navegação, selecione Código.
4. Selecione Ambientes de desenvolvimento.
5. Escolha o Ambiente de Desenvolvimento que você deseja editar.
6. Escolha Editar.
7. Faça as alterações desejadas no tempo limite de computação ou inatividade.
8. Escolha Salvar.

## Excluir um Ambiente de Desenvolvimento

Quando terminar de trabalhar no conteúdo armazenado em seu Ambiente de Desenvolvimento, você poderá excluí-lo. Crie um Ambiente de Desenvolvimento para trabalhar em novos conteúdos. Se você excluir seu Ambiente de Desenvolvimento, o conteúdo persistente será excluído permanentemente. Antes de excluir o Ambiente de Desenvolvimento, confirme e envie suas alterações de código para o repositório de origem original do Ambiente de Desenvolvimento. Depois de excluir seu Ambiente de Desenvolvimento, o faturamento de computação e armazenamento do Ambiente de Desenvolvimento será interrompido.

Depois de excluir seu Ambiente de Desenvolvimento, pode demorar alguns minutos até que a cota de armazenamento seja atualizada. Se você atingiu a cota de armazenamento, não poderá criar um Ambiente de Desenvolvimento durante esse período.

### Important

A exclusão de um Ambiente de Desenvolvimento não pode ser desfeita. Depois de excluir um Ambiente de Desenvolvimento, não é mais possível recuperá-lo.



## Como excluir um Ambiente de Desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja excluir um Ambiente de Desenvolvimento.
3. No painel de navegação, selecione Código.
4. Selecione Ambientes de desenvolvimento.
5. Escolha o Ambiente de Desenvolvimento que você deseja excluir.
6. Escolha Excluir.
7. Digite **delete** para confirmar a exclusão do Ambiente de Desenvolvimento.
8. Escolha Excluir.

### Note

Antes de excluir uma conexão VPC em seu espaço, remova o Ambiente de Desenvolvimento associado a essa VPC.

Mesmo se você excluir um Ambiente de Desenvolvimento, talvez não exclua a interface de rede na VPC. Limpe seus recursos conforme necessário. Se ocorrer um erro ao excluir um Ambiente de Desenvolvimento conectado à VPC, você deverá [desconectar](#) sua conexão obsoleta e [excluí-la](#) depois de confirmar que ela não está sendo usada.

## Estabelecer conexão com um Ambiente de Desenvolvimento usando SSH

Você pode se conectar ao seu ambiente de desenvolvimento usando SSH para realizar ações sem limitações, como encaminhamento de portas, upload e download de arquivos e uso de outros. IDEs

### Note

Se você quiser continuar usando o SSH por um longo período após fechar a guia ou janela do IDE, defina um tempo limite alto para seu Ambiente de Desenvolvimento para que ele não pare devido à inatividade no IDE.

## Pré-requisitos

- Você precisa de um dos seguintes sistemas operacionais:
  - Windows 10 ou mais recente e OpenSSH habilitado
  - macOS e Bash versão 3 ou superior
  - Linux com yum, dpkg ou gerenciadores de pacotes rpm e Bash versão 3 ou superior
- Você também precisa da AWS CLI versão 2.9.4 ou superior.

Para se conectar a um Ambiente de Desenvolvimento usando SSH

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que deseja se conectar a um Ambiente de Desenvolvimento usando SSH.
3. No painel de navegação, selecione Código.
4. Selecione Ambientes de desenvolvimento.
5. Escolha um Ambiente de Desenvolvimento em execução ao qual você deseja se conectar usando SSH.
6. Selecione Conectar via SSH, escolha o sistema operacional desejado e faça o seguinte:
  - Caso ainda não tenha feito isso, cole e execute o primeiro comando no terminal especificado. O comando baixa um script e executa as seguintes modificações em seu ambiente local para que você possa se conectar ao seu Ambiente de Desenvolvimento usando SSH:
    - Instala o [plug-in do Gerenciador de Sessões para o AWS CLI](#)
    - Modifica seu local AWS Config e adiciona um CodeCatalyst perfil para que você possa realizar o login do SSO. Para obter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).
    - Modifica sua configuração SSH local e adiciona a configuração necessária para se conectar ao seu Ambiente de Desenvolvimento usando SSH.
    - Adiciona um script no diretório `~/.aws/codecatalyst-dev-env` usado pelo cliente SSH para se conectar ao seu Ambiente de Desenvolvimento. Esse script chama a [CodeCatalyst StartDevEnvironmentSession API](#) e usa o AWS Systems Manager Session Manager plug-in para estabelecer uma AWS Systems Manager sessão com seu ambiente de desenvolvimento, que é usada pelo cliente SSH local para se conectar com segurança ao ambiente de desenvolvimento remoto.

- Faça login na Amazon CodeCatalyst usando o AWS SSO usando o segundo comando. [Esse comando solicita e recupera credenciais para que o script no ~/.aws/codecatalyst-dev-env diretório possa chamar CodeCatalyst StartDevEnvironmentSession a API](#). Esse comando deve ser executado sempre que suas credenciais expirarem. Ao executar o último comando no modal(ssh<destination>), você receberá um erro se suas credenciais expirarem ou se você não tiver realizado o login do SSO conforme as instruções nesta etapa.
- Conecte-se ao Ambiente de Desenvolvimento especificado usando SSH usando o terceiro comando. Este comando tem a seguinte estrutura:

```
ssh codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

Você também pode usar esse comando para realizar outras ações permitidas pelo cliente SSH, como encaminhamento de porta ou upload e download de arquivos:

- Encaminhamento de portas:

```
ssh -L <local-port>:127.0.0.1:<remote-port> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

- Upload de um arquivo para o diretório inicial em seu Ambiente de Desenvolvimento:

```
scp -0 </path-to-local-file> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>:</path-to-remote-file-or-directory>
```

## Configuração de um devfile para um Ambiente de Desenvolvimento

Um devfile é um padrão aberto que ajuda você a personalizar seus ambientes de desenvolvimento em toda a sua equipe. Um devfile é um arquivo YAML que codifica as ferramentas de desenvolvimento necessárias. Ao configurar um devfile, você pode pré-determinar as ferramentas de projeto e as bibliotecas de aplicativos de que precisa e a Amazon as CodeCatalyst instala em seu ambiente de desenvolvimento para você. O devfile é específico do repositório para o qual foi criado, e você pode criar um devfile separado para cada repositório. Seu Ambiente de Desenvolvimento oferece suporte a comandos e eventos e fornece uma imagem padrão de devfile universal.

Se você criar um projeto usando o esquema vazio, poderá criar um devfile manualmente. Se você criar um projeto usando um blueprint diferente, CodeCatalyst cria um devfile automaticamente. O diretório /projects de um Ambiente de Desenvolvimento armazena os arquivos que são

extraídos do repositório de origem e do devfile. O diretório `/home`, que está vazio quando você cria um Ambiente de Desenvolvimento pela primeira vez, armazena os arquivos criados enquanto usa o Ambiente de Desenvolvimento. Tudo nos diretórios `/projects` e `/home` de um Ambiente de Desenvolvimento é armazenado de forma persistente.

 **Note**

A pasta `/home` só será alterada se você alterar o nome do devfile ou o nome do componente devfile. Se você alterar o nome do devfile ou do componente devfile, o conteúdo do diretório `/home` será substituído e os dados do diretório `/home` anterior não poderão ser recuperados.

Se você criar um Ambiente de Desenvolvimento com um repositório de origem que não contenha um devfile na raiz ou criar um Ambiente de Desenvolvimento sem um repositório de origem, um devfile universal padrão será aplicado automaticamente ao repositório de origem. A mesma imagem devfile universal padrão é usada para todos IDEs. CodeCatalyst atualmente suporta a versão 2.0.0 do devfile. Para ter mais informações sobre o devfile, consulte [Esquema Devfile - Versão 2.0.0](#).

 **Note**

Você só pode incluir imagens de contêiner público em seu devfile.

Os ambientes de desenvolvimento conectados à VPC oferecem suporte apenas às seguintes imagens de devfile:

- Imagem universal
- Imagens privadas do Amazon ECR, se o repositório estiver na mesma região da VPC

## Tópicos

- [Edição do repositório devfile para um Ambiente de Desenvolvimento](#)
- [Recursos do Devfile suportados pelo CodeCatalyst](#)
- [Exemplo de um devfile para um Ambiente de Desenvolvimento](#)
- [Solução de problemas de um repositório devfile usando o modo de recuperação](#)
- [Especificar imagens de devfile universais para um Ambiente de Desenvolvimento](#)
- [Comandos Devfile](#)

- [Eventos do Devfile](#)
- [Componentes do Devfile](#)

## Edição do repositório devfile para um Ambiente de Desenvolvimento

Use o procedimento a seguir para editar o repositório devfile para um Ambiente de Desenvolvimento.

### Editando um devfile de repositório para um ambiente de desenvolvimento no CodeCatalyst

Para editar o repositório devfile

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto que contém o repositório de origem para o qual você deseja editar o devfile.
3. No painel de navegação, selecione Código.
4. Selecione Repositórios de origem.
5. Escolha o repositório de origem que contém o devfile que você deseja editar.
6. Na lista de arquivos, escolha o arquivo `devfile.yaml`.
7. Escolha Editar.
8. Edite o devfile.
9. Selecione Confirmar ou crie uma solicitação pull para que um membro da equipe possa revisar e aprovar as alterações.

#### Note

Se editar o devfile, você precisará reiniciar o devfile para que as alterações sejam implementadas. Isso pode ser feito executando `/aws/mde/mde start --location devfile.yaml`. Se houver algum problema ao iniciar o devfile, ele entrará no modo de recuperação. No entanto, se você editar um devfile associado a um Ambiente de Desenvolvimento conectado à VPC, precisará reiniciar o Ambiente de Desenvolvimento para que as alterações entrem em vigor.

Você pode revisar qual devfile está sendo usado executando `/aws/mde/mde status`. O campo de localização tem o caminho do devfile em relação à pasta `/projects` do ambiente.

```
{
 "status": "STABLE",
 "location": "devfile.yaml"
}
```

Você também pode mover o devfile padrão em `/projects/devfile.yaml` para o seu repositório de código-fonte. Para atualizar a localização do devfile, use o seguinte comando: `/aws/mde/mde start --location repository-name/devfile.yaml`.

## Edição do repositório devfile para um Ambiente de Desenvolvimento em um IDE

Para alterar a configuração de um Ambiente de Desenvolvimento, você deve editar o devfile. Recomendamos que você edite o devfile em um IDE compatível e, em seguida, atualize seu ambiente de desenvolvimento, mas você também pode editar o devfile a partir da raiz do repositório de origem em CodeCatalyst. Se você editar o devfile em um IDE compatível, confirme e envie suas alterações para o repositório de origem ou crie uma solicitação pull para que um membro da equipe possa revisar e aprovar as edições no devfile.

- [Editando o arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento no AWS Cloud9](#)
- [Edição do repositório devfile para um Ambiente de Desenvolvimento no VS Code](#)
- [Editando o arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento no JetBrains](#)

## Recursos do Devfile suportados pelo CodeCatalyst

CodeCatalyst suporta os seguintes recursos do devfile na versão 2.0.0. Para ter mais informações sobre o devfile, consulte [Esquema Devfile - Versão 2.0.0](#).

| Recurso                | Tipo    |
|------------------------|---------|
| <code>exec</code>      | Command |
| <code>postStart</code> | Event   |

| Recurso      | Tipo                       |
|--------------|----------------------------|
| container    | Componente                 |
| args         | Propriedades do componente |
| env          | Propriedades do componente |
| mountSources | Propriedades do componente |
| volumeMounts | Propriedades do componente |

## Exemplo de um devfile para um Ambiente de Desenvolvimento

Veja a seguir um exemplo de um devfile simples.

```
schemaVersion: 2.0.0
metadata:
 name: al2
components:
 - name: test
 container:
 image: public.ecr.aws/amazonlinux/amazonlinux:2
 mountSources: true
 command: ['sleep', 'infinity']
 - name: dockerstore
commands:
 - id: setupscript
 exec:
 component: test
 commandLine: "chmod +x script.sh"
 workingDir: /projects/devfiles
 - id: executescript
 exec:
 component: test
 commandLine: "/projects/devfiles/script.sh"
 - id: yumupdate
 exec:
 component: test
 commandLine: "yum -y update --security"
events:
```

```
postStart:
- setupscript
- executescript
- yumupdate
```

Os logs de inicialização, comandos e eventos do devfile são capturados e armazenados em `/aws/mde/logs`. Para depurar o comportamento do devfile, inicie seu Ambiente de Desenvolvimento usando um devfile ativo e acesse os logs.

## Solução de problemas de um repositório devfile usando o modo de recuperação

Se houver algum problema ao iniciar seu devfile, ele entrará no modo de recuperação para que você ainda possa se conectar ao ambiente e corrigir o devfile. No modo de recuperação, a execução de `/aws/mde/mde status` não conterá a localização do devfile.

```
{
 "status": "STABLE"
}
```

Você pode verificar o erro nos logs em `/aws/mde/logs`, corrigir o devfile e tentar executar `/aws/mde/mde start` novamente.

## Especificar imagens de devfile universais para um Ambiente de Desenvolvimento

A imagem universal padrão inclui as linguagens de programação mais usadas e as ferramentas relacionadas que podem ser usadas em seu IDE. Se nenhuma imagem for especificada, o CodeCatalyst fornecerá essa imagem e conterá ferramentas que são mantidas pelo CodeCatalyst. Para permanecer notificado sobre o lançamento de novas imagens, consulte [Assinatura de notificações de imagem universal com o SNS](#).

O Amazon CodeCatalyst oferece suporte ativo às seguintes imagens de devfile:

| Versão da imagem    | Identificador da imagem                    |
|---------------------|--------------------------------------------|
| Universal image 4.0 | public.ecr.aws/aws-mde/universal-image:4.0 |



| Versão da imagem    | Identificador da imagem                                 |
|---------------------|---------------------------------------------------------|
| Universal image 5.0 | <code>public.ecr.aws/aws-mde/universal-image:5.0</code> |

**Note**

Você também pode usar `public.ecr.aws/aws-mde/universal-image:latest` para ter a imagem mais recente, que atualmente é `public.ecr.aws/aws-mde/universal-image:5.0`.

O CodeCatalyst tornou obsoletas as imagens a seguir. Ainda é possível usar essas imagens, mas elas não serão armazenadas em cache no host de criação e resultarão em tempos de inicialização do Ambiente de Desenvolvimento mais altos.

| Versão da imagem    | Identificador da imagem                                 | Data da substituição |
|---------------------|---------------------------------------------------------|----------------------|
| Universal image 1.0 | <code>public.ecr.aws/aws-mde/universal-image:1.0</code> | 16 de agosto de 2024 |
| Universal image 2.0 | <code>public.ecr.aws/aws-mde/universal-image:2.0</code> | 16 de agosto de 2024 |
| Universal image 3.0 | <code>public.ecr.aws/aws-mde/universal-image:3.0</code> | 30 de julho de 2025  |

**Note**

Se você estiver usando AWS Cloud9, o preenchimento automático não funcionará para PHP, Ruby e CSS após a atualização para `universal-image:3.0`.

## Tópicos

- [Assinatura de notificações de imagem universal com o SNS](#)
- [Versões de runtime da imagem universal 4.0](#)
- [Versões de runtime da imagem universal 5.0](#)

## Assinatura de notificações de imagem universal com o SNS

O CodeCatalyst fornece um serviço de notificação de imagens universais. Você pode usá-lo para se inscrever em um tópico do Amazon Simple Notification Service (SNS) que envia notificações quando as atualizações de imagens universais do CodeCatalyst são lançadas. Para ter mais informações sobre tópicos do SNS, consulte [O que é o Amazon Simple Notification Service?](#).

Sempre que novas imagens universais são lançadas, enviamos notificações aos assinantes. Esta seção descreve como assinar as atualizações de imagens universais do CodeCatalyst.

### Exemplo de mensagem

```
{
 "Type": "Notification",
 "MessageId": "123456789",
 "TopicArn": "arn:aws:sns:us-east-1:1234657890:universal-image-updates",
 "Subject": "New Universal Image Release",
 "Message": {
 "v1": {
 "Message": "A new version of the Universal Image has been released. You are now able to launch new DevEnvironments using this image.",
 "image ": {
 "release_type": "MAJOR VERSION",
 "image_name": "universal-image",
 "image_version": "2.0",
 "image_uri": "public.ecr.aws/amazonlinux/universal-image:2.0"
 }
 }
 },
 "Timestamp": "2021-09-03T19:05:57.882Z",
 "UnsubscribeURL": "example url"
}
```

Para se inscrever nas atualizações de imagens universais do CodeCatalyst usando o console do Amazon SNS

1. Abra o console do Amazon SNS no [Painel](#).
2. No painel de navegação, selecione sua Região da AWS.
3. No painel de navegação, escolha Subscriptions (Assinaturas) e, depois, selecione Create subscription (Criar assinatura).
4. Em ARN do tópico, insira `arn:aws:sns:us-east-1:089793673375:universal-image-updates`.
5. Em Protocol (Protocolo), selecione Email.
6. No Endpoint, forneça um endereço de e-mail. O endereço de e-mail será usado para receber notificações.
7. Selecione Criar assinatura.
8. Você receberá um e-mail de confirmação com o assunto “Notificação da AWS – Confirmação de assinatura”. Abra o e-mail e selecione Confirmar assinatura.

Para cancelar a inscrição nas atualizações de imagens universais do CodeCatalyst usando o console do Amazon SNS

1. Abra o console do Amazon SNS no [Painel](#).
2. No painel de navegação, selecione sua Região da AWS.
3. No painel de navegação, selecione Assinaturas e selecione a assinatura da qual você deseja cancelar a assinatura.
4. Escolha Ações e escolha Excluir assinaturas.
5. Escolha Excluir.

## Versões de runtime da imagem universal 4.0

A tabela a seguir lista os runtimes para `universal-image:4.0`.

**universal-image:4.0** Versões do runtime de

| Nome do runtime | Versão     | Versão secundária mais recente e principal específica |
|-----------------|------------|-------------------------------------------------------|
| aws cli         | 2.11       | aws-cli: 2.x                                          |
| docker compose  | 2.17       | docker-compose: 2.x                                   |
| dotnet          | 8.0        | dotnet: 8.x                                           |
| golang          | 1.22       | golang: 1.x                                           |
| java            | corretto21 | java: corretto21.x                                    |
| nodejs          | 20.6       | nodejs: 20.x                                          |
| php             | 8.2        | php: 8.x                                              |
| python          | 3.9        | python: 3.x                                           |
|                 | 3.12       |                                                       |
| ruby            | 3.3        | ruby: 3.x                                             |
| terraform       | 1.5        | terraform: 1.x                                        |

**Versões de runtime da imagem universal 5.0**

A tabela a seguir lista os runtimes disponíveis para `universal-image:5.0`.

**universal-image:5.0** Versões do runtime de

| Nome do runtime | Versão | Versão secundária mais recente e principal específica |
|-----------------|--------|-------------------------------------------------------|
| aws cli         | 2.25   | aws-cli: 2.x                                          |
| docker compose  | 2.34   | docker-compose: 2.x                                   |
| dotnet          | 8.0    | dotnet: 8.x                                           |

| Nome do runtime | Versão     | Versão secundária mais recente e principal específica |
|-----------------|------------|-------------------------------------------------------|
| golang          | 1.24       | golang: 1.x                                           |
| java            | corretto21 | java: corretto21.x                                    |
| nodejs          | 22.0       | nodejs: 22.x                                          |
| php             | 8.3.16     | php: 8.x                                              |
| python          | 3.12       | python: 3.x                                           |
| ruby            | 3.4.2      | ruby: 3.x                                             |
| terraform       | 1.10.5     | terraform: 1.x                                        |

## Comandos Devfile

Atualmente, CodeCatalyst só suporta `exec` comandos em seu devfile. Para ter mais informações, consulte [Adicionar comandos](#) na documentação do Devfile.io.

O exemplo a seguir mostra como especificar comandos `exec` no devfile.

```
commands:
 - id: setupscript
 exec:
 component: test
 commandLine: "chmod +x script.sh"
 workingDir: /projects/devfiles
 - id: executescript
 exec:
 component: test
 commandLine: "./projects/devfiles/script.sh"
 - id: updateyum
 exec:
 component: test
 commandLine: "yum -y update --security"
```

Depois de se conectar ao Ambiente de Desenvolvimento, você pode executar comandos definidos por meio do terminal.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescript
```

Para comandos de longa execução, você pode usar o sinalizador de streaming `-s` para gerar a execução do comando em tempo real.

```
/aws/mde/mde -s command <command-id>
```

#### Note

`command-id` deve estar em letras minúsculas.

## Parâmetros de execução suportados por CodeCatalyst

CodeCatalyst suporta os seguintes exec parâmetros na versão 2.0.0 do devfile.

- `commandLine`
- `component`
- `id`
- `workingDir`

## Eventos do Devfile

Atualmente, CodeCatalyst só oferece suporte a `postStart` eventos em seu devfile. Para obter mais informações, consulte a [postStartObject](#) documentação do Devfile.io.

O exemplo a seguir mostra como adicionar vinculações de eventos `postStart` ao devfile.

```
commands:
 - id: executescript
 exec:
 component: test
 commandLine: "./projects/devfiles/script.sh"
```

```
- id: updateyum
 exec:
 component: test
 commandLine: "yum -y update --security"
events:
 postStart:
 - updateyum
 - executescript
```

Após a inicialização, seu Ambiente de Desenvolvimento executará os comandos `postStart` especificados na ordem em que foram definidos. Se um comando falhar, o Ambiente de Desenvolvimento continuará em execução e a saída da execução será armazenada nos logs em `/aws/mde/logs`.

## Componentes do Devfile

Atualmente, CodeCatalyst só oferece suporte a `container` componentes em seu devfile. Para ter mais informações, consulte [Adicionar componentes](#) na documentação do Devfile.io.

O exemplo a seguir mostra como adicionar um comando de inicialização ao contêiner no devfile.

```
components:
 - name: test
 container:
 image: public.ecr.aws/amazonlinux/amazonlinux:2
 command: ['sleep', 'infinity']
```

### Note

Quando o contêiner tem um comando de entrada de curta duração, você deve incluir `command: ['sleep', 'infinity']` para manter o contêiner em execução.

CodeCatalyst também oferece suporte às seguintes propriedades em seu componente de contêiner: `args`, `envmountSources`, `volumeMounts` e.

## Associar uma conexão VPC a um Ambiente de Desenvolvimento

Uma conexão VPC é um CodeCatalyst recurso que contém todas as configurações necessárias para que seu fluxo de trabalho acesse uma VPC. Os administradores do espaço podem adicionar suas

próprias conexões de VPC no console da CodeCatalyst Amazon em nome dos membros do espaço. Ao adicionar uma conexão VPC, os membros do espaço podem executar ações de fluxo de trabalho e criar ambientes de desenvolvimento que sigam as regras de rede e possam acessar recursos na VPC associada.

 Important

Ambientes de desenvolvimento com uma conexão VPC não oferecem suporte a [repositórios de origem de terceiros](#) vinculados a. CodeCatalyst

Você só pode associar um Ambiente de Desenvolvimento a uma conexão VPC após a criação do Ambiente de Desenvolvimento. Não é possível alterar a conexão VPC associada ao Ambiente de Desenvolvimento depois de criá-la. Se quiser usar uma conexão VPC diferente, você precisa excluir o Ambiente de Desenvolvimento atual e criar outro.

 Note

Os ambientes de desenvolvimento só podem ser associados a uma conexão VPC com uma AWS conta que tenha acesso ao seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

Observe que os ambientes de desenvolvimento utilizam vários AWS recursos e serviços na criação. Isso significa que os Dev Environments se conectam aos seguintes AWS serviços:

- Amazon CodeCatalyst
- AWS SSM
- AWS KMS
- Amazon ECR
- Amazon CloudWatch
- Amazon ECS



**Note**

AWS Toolkit não oferece suporte à criação de ambientes de desenvolvimento com uma conexão VPC associada. Observe também que, se você usar um IDE diferente AWS Cloud9, poderá ter tempos de carregamento de cerca de cinco minutos.

Você deve ter o perfil de Administrador do espaço ou Usuário avançado para gerenciar conexões VPC no nível do espaço. Para obter mais informações sobre VPCs, consulte [Gerenciando a Amazon VPCs CodeCatalyst no](#) Guia do CodeCatalyst Administrador.

## Cotas para ambientes de desenvolvimento em CodeCatalyst

A tabela a seguir descreve cotas e limites para ambientes de desenvolvimento na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

|                                                                       |                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Número de horas de Ambiente de Desenvolvimento por mês                | As horas do Ambiente de Desenvolvimento são afetadas pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte <a href="#">Preços</a> e <a href="#">Solucionar problemas com Ambientes de Desenvolvimento</a> .     |
| Quantidade de armazenamento do Ambiente de Desenvolvimento por espaço | O armazenamento do Ambiente de Desenvolvimento é afetado pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte <a href="#">Preços</a> e <a href="#">Solucionar problemas com Ambientes de Desenvolvimento</a> . |
| Quantidade de computação do Ambiente de Desenvolvimento               | A computação do Ambiente de Desenvolvimento é afetada pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte <a href="#">Preços</a> e <a href="#">Solucionar problemas com Ambientes de Desenvolvimento</a> .    |

# Publique e compartilhe pacotes de software no CodeCatalyst

A Amazon CodeCatalyst contém um serviço de repositório de pacotes totalmente gerenciado que facilita para sua equipe de desenvolvimento armazenar e compartilhar com segurança pacotes de software usados para desenvolvimento de aplicativos. Esses pacotes são armazenados em repositórios de pacotes, que são criados e organizados em projetos em CodeCatalyst.

Um único repositório de pacotes pode armazenar pacotes de todos os tipos de pacotes compatíveis. CodeCatalyst suporta os seguintes formatos de pacote:

- npm
- Maven
- NuGet
- Python

Os pacotes em um repositório de pacotes podem ser descobertos e compartilhados entre os membros do projeto que contém o repositório.

Para publicar pacotes e consumir pacotes de um repositório para outro, configure um gerenciador de pacotes para usar o endpoint do repositório (URL). Em seguida, você pode usar o gerenciador de pacotes para publicar pacotes no repositório. Você pode usar gerenciadores de pacotes, como Maven, Gradle, npm, yarn, nuget, dotnet, pip e twine.

Você também pode configurar CodeCatalyst fluxos de trabalho para usar repositórios de CodeCatalyst pacotes. Para ter mais informações sobre o uso de pacotes em fluxos de trabalho, consulte [Conectar repositórios de pacotes a fluxos de trabalho](#).

Você pode disponibilizar pacotes de um repositório para outro no mesmo projeto. Para isso, adicione-os como um repositório upstream. Todas as versões do pacote disponíveis para o repositório upstream também estão disponíveis para o repositório downstream. Para obter mais informações, consulte [Configuração e uso de repositórios upstream](#).

Você pode disponibilizar pacotes de código aberto para seu CodeCatalyst repositório criando um tipo especial de repositório chamado gateway. O upstreaming para um repositório de gateway permite que você consuma pacotes de repositórios públicos populares, como npmjs.com e pypi.org, e os

armazene automaticamente em seu repositório. CodeCatalyst Para obter mais informações, consulte [Conexão a repositórios externos públicos](#).

## Tópicos

- [Conceitos dos pacotes](#)
- [Configuração e uso de repositórios de pacotes](#)
- [Configuração e uso de repositórios upstream](#)
- [Conexão a repositórios externos públicos](#)
- [Publicação e modificação de pacotes](#)
- [Uso de npm](#)
- [Usando o Maven](#)
- [Usando NuGet](#)
- [Usar o Python](#)
- [Cotas para pacotes](#)

## Conceitos dos pacotes

Aqui estão alguns conceitos e termos que você deve conhecer ao gerenciar, publicar ou consumir pacotes CodeCatalyst.

### Pacotes

Um pacote é um pacote que inclui o software e os metadados necessários para instalar o software e resolver quaisquer dependências. CodeCatalyst suporta o formato de pacote npm.

Um pacote consiste em:

- Um nome (por exemplo, webpack é o nome de um pacote npm conhecido)
- Um [namespace](#) opcional (por exemplo, @types em @types/node)
- Um conjunto de [versões](#) (por exemplo, 1.0.0, 1.0.1, 1.0.2)
- Metadados em nível de pacote (por exemplo, tags npm dist)

## Namespaces de pacote

Alguns formatos de pacotes permitem a utilização de nomes de pacotes hierárquicos para organizar os pacotes em grupos lógicos e ajudar a evitar colisões de nomes. Pacotes com o mesmo nome podem ser armazenados em namespaces diferentes. Por exemplo, o npm suporta escopos, e o pacote `npm @types/node` tem um escopo `@types` e um nome `node`. Há vários outros nomes de pacotes no escopo `@types`. Em CodeCatalyst, o escopo (“tipos”) é chamado de namespace do pacote e o nome (“nó”) é chamado de nome do pacote. Para pacotes do Maven, o namespace do pacote corresponde ao `groupId` do Maven. O pacote `org.apache.logging.log4j:log4j` do Maven tem o `groupId` (namespace do pacote) `org.apache.logging.log4j` e o `artifactId` (nome do pacote) `log4j`. Alguns formatos de pacotes, como do Python, não permitem a utilização de nomes hierárquicos com um conceito semelhante ao escopo do npm ou ao `groupId` do Maven. Se não houver nenhuma maneira de agrupar nomes de pacotes, evitar colisões de nomes pode ser mais difícil.

## Versões do pacote

A versão do pacote identifica a versão específica de um pacote, como `@types/node@12.6.9`. O formato e a semântica do número da versão variam conforme os diferentes formatos de pacote. Por exemplo, as versões dos pacotes do npm devem estar em conformidade com a [especificação de Versionamento semântico](#). Em CodeCatalyst, uma versão do pacote consiste no identificador da versão, nos `package-version-level` metadados e em um conjunto de ativos.

## Ativos

Um ativo é um arquivo individual armazenado CodeCatalyst que está associado a uma versão do pacote, como um arquivo npm ou um `.tgz` arquivo Maven POM ou JAR.

## Repositórios de pacotes

Um repositório de CodeCatalyst pacotes contém um conjunto de [pacotes](#), que contêm [versões de pacotes](#), cada uma delas mapeada para um conjunto de [ativos](#). Os repositórios de pacotes são políglotas, o que significa que um único repositório pode conter pacotes de qualquer tipo compatível. Cada repositório de pacotes expõe endpoints para buscar e publicar pacotes usando ferramentas como NuGet CLIs (`nuget`), a CLIdotnet, a npm CLI do Maven (`mvn`) e o Python (`emvn`). CLIs `pip` `twine` Para obter informações sobre cotas de pacotes em CodeCatalyst, incluindo quantos repositórios de pacotes podem ser criados em cada espaço, consulte [Cotas para pacotes](#)

Você pode vincular um repositório de pacotes a outro definindo-o como upstream. Quando um repositório é definido como upstream, você pode usar qualquer pacote do upstream, bem como quaisquer repositórios upstream adicionais na cadeia. Para obter mais informações, consulte [Repositórios upstream](#).

Os repositórios de gateway são um tipo especial de repositório de pacotes que extrai e armazena pacotes de autoridades externas oficiais de pacotes. Para obter mais informações, consulte [Repositórios de gateway](#).

## Repositórios upstream

Você pode usar CodeCatalyst para criar um relacionamento upstream entre dois repositórios de pacotes. Um repositório de pacotes é upstream quando as versões do pacote contidas nele podem ser acessadas a partir do endpoint do repositório de pacotes downstream. Com um relacionamento inicial, o conteúdo dos dois repositórios de pacotes é mesclado com eficácia do ponto de vista de um cliente.

Por exemplo, se um gerenciador de pacotes solicitar uma versão de pacote que não existe em um repositório, CodeCatalyst pesquisará a versão do pacote nos repositórios upstream configurados. Os repositórios upstream são pesquisados na ordem em que estão configurados e, quando um pacote é encontrado, CodeCatalyst a pesquisa é interrompida.

## Repositórios de gateway

Um repositório de gateway é um tipo especial de repositório de pacotes conectado a uma autoridade de pacotes oficial e externa compatível. Ao adicionar um repositório de gateway como [repositório upstream](#), você pode consumir pacotes da autoridade oficial de pacotes correspondente. Seu repositório downstream não se comunica com o repositório público; em vez disso, tudo é intermediado pelo repositório de gateway. Os pacotes consumidos dessa maneira são armazenados no repositório de gateway e no repositório downstream que recebeu a solicitação original.

Os repositórios de gateway são predefinidos, mas devem ser criados em cada projeto para serem usados. A lista a seguir contém todos os repositórios de gateway que podem ser criados CodeCatalyst e a autoridade do pacote à qual eles estão conectados.

- npm-public-registry-gatewayfornece pacotes npm de npmjs.com.
- maven-central-gatewayfornece pacotes Maven do repositório Maven Central.
- google-android-gatewayfornece pacotes Maven do Google Android.

- commonsware-gateway fornece pacotes Maven de CommonsWare
- gradle-plugins-gateway fornece pacotes Maven dos plug-ins do Gradle.
- nuget-gallery-gateway fornece NuGet pacotes da NuGet Galeria.
- pypi-gateway fornece pacotes Python do Python Package Index.

## Configuração e uso de repositórios de pacotes

Em CodeCatalyst, os pacotes são armazenados e gerenciados dentro dos repositórios de pacotes. Para publicar pacotes CodeCatalyst ou consumir pacotes de um CodeCatalyst (ou de qualquer repositório de pacotes público compatível), você deve criar um repositório de pacotes e conectar seu gerenciador de pacotes a ele.

### Tópicos

- [Criação de um repositório de pacotes](#)
- [Conexão a um repositório de pacotes](#)
- [Exclusão de um repositório de pacotes](#)

## Criação de um repositório de pacotes

Execute as etapas a seguir para criar um repositório de pacotes no CodeCatalyst.

Para criar um repositório de pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto onde você deseja criar um repositório de pacotes.
3. No painel de navegação, selecione Pacotes.
4. Na página Repositórios de pacotes, selecione Criar repositório de pacotes.
5. Na seção Detalhes do repositório de pacotes, adicione o seguinte:
  - a. Nome do repositório. Considere usar um nome descritivo com detalhes como o nome do projeto ou da equipe, ou como o repositório será usado.
  - b. (Opcional) Descrição. Uma descrição do repositório é especialmente útil quando você tem vários repositórios em várias equipes em um projeto.
6. Na seção Repositórios upstream, escolha Selecionar repositórios upstream para adicionar qualquer repositório de pacotes que você queira acessar por meio do seu repositório de

pacotes. CodeCatalyst Você pode adicionar repositórios do Gateway para se conectar a repositórios de pacotes externos ou outros CodeCatalyst repositórios.

- Quando um pacote é solicitado em um repositório de pacotes, os repositórios upstream são pesquisados na ordem em que aparecem nessa lista. Quando um pacote for encontrado, CodeCatalyst ele deixará de pesquisar. Para alterar a ordem dos repositórios upstream, você pode arrastar e soltar os repositórios na lista.

7. Selecione Criar para criar seu repositório de pacotes.

## Conexão a um repositório de pacotes

Para publicar ou consumir pacotes de CodeCatalyst, você deve configurar seu gerenciador de pacotes com as informações e CodeCatalyst credenciais do endpoint do repositório de pacotes. Se você não criou um repositório, pode fazer isso seguindo as instruções em [Criação de um repositório de pacotes](#).

Para obter instruções sobre como conectar um gerenciador de pacotes a um repositório de CodeCatalyst pacotes, consulte a documentação a seguir.

- [Configuração e uso do Gradle Groovy](#)
- [Configuração e uso de mvn](#)
- [Configuração e uso da CLI nuget ou dotnet](#)
- [Configuração e uso de npm](#)
- [Configurar pip e instalação de pacotes Python](#)
- [Configuração do Twine e publicação de pacotes Python](#)

## Exclusão de um repositório de pacotes

Execute as etapas a seguir para excluir um repositório de pacotes no CodeCatalyst.

Para excluir um repositório de pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto que contém o repositório de pacotes que deseja excluir.
3. No painel de navegação, selecione Pacotes.
4. Na página Repositórios de pacotes, escolha o repositório que deseja excluir.

5. Escolha Excluir.
6. Analise as informações fornecidas sobre os efeitos da exclusão de um repositório de pacotes.
7. Insira `delete` no campo de entrada e selecione Excluir.

## Configuração e uso de repositórios upstream

Você pode conectar os repositórios de gateway e outros repositórios de CodeCatalyst pacotes como upstreams aos seus repositórios de pacotes. Isso permite que um cliente gerenciador de pacotes acesse os pacotes contidos em mais de um repositório de pacotes usando um único endpoint de repositório de pacotes. Confira a seguir os principais benefícios do uso de repositórios upstream:

- Você só precisa configurar o gerenciador de pacotes com um único endpoint de repositório para extrair de várias fontes.
- Os pacotes consumidos de um repositório upstream são armazenados em seu repositório downstream, o que garante que seus pacotes estejam disponíveis mesmo que o repositório upstream sofra interrupções inesperadas ou que os pacotes no repositório upstream sejam excluídos.

Você pode adicionar repositórios upstream ao criar um repositório de pacotes. Você também pode adicionar ou remover repositórios upstream dos repositórios de pacotes existentes no console.

### CodeCatalyst

Quando você adiciona um repositório de gateway como repositório upstream, o repositório de pacotes é conectado ao repositório de pacotes público correspondente do repositório de gateway. Para ver uma lista dos repositórios de pacotes públicos compatíveis, consulte [Repositórios de pacotes externos compatíveis e repositórios de gateway](#).

Você pode vincular vários repositórios como repositórios upstream. Por exemplo, suponha que sua equipe crie um repositório chamado `project-repo` e já esteja usando outro repositório chamado `team-repo` que foi `npm-public-registry-gateway` adicionado como um repositório upstream, que está conectado ao repositório npm público, `npmjs.com`. Você pode adicionar `team-repo` como um repositório upstream a `project-repo`. Nesse caso, você só precisa configurar o gerenciador de pacotes para usar `project-repo` para extrair pacotes de `project-repo`, `team-repo`, `npm-public-registry-gateway` e `npmjs.com`.

### Tópicos

- [Adição de um repositório upstream](#)



- [Edição da ordem de pesquisa dos repositórios upstream](#)
- [Solicitar uma versão do pacote com repositórios upstream](#)
- [Remoção de um repositório upstream](#)

## Adição de um repositório upstream

Adicionar um repositório de pacotes público ou outro repositório de CodeCatalyst pacotes como repositório upstream ao seu repositório downstream disponibiliza todos os pacotes no repositório upstream para gerenciadores de pacotes conectados ao repositório downstream.

Para adicionar um repositório upstream

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, escolha o repositório de pacotes ao qual deseja adicionar um repositório upstream.
3. No nome do repositório de pacotes, selecione Upstreams e Selecionar repositórios upstream.
4. Em Selecionar tipo upstream, escolha uma das seguintes opções:
  - Repositórios de gateway

É possível escolher em uma lista de repositórios de gateway disponíveis.

### Note

Para se conectar a autoridades públicas externas de pacotes, como Maven Central, npmjs.com ou Nuget Gallery, CodeCatalyst use repositórios de gateway como repositórios intermediários que pesquisam e armazenam pacotes retirados de repositórios externos. Isso economiza tempo e transferência de dados, pois todos os repositórios de pacotes em um projeto usarão pacotes do repositório intermediário do gateway. Para obter mais informações, consulte [Conexão a repositórios externos públicos](#).

- CodeCatalyst repositórios

Você pode escolher entre uma lista de repositórios de CodeCatalyst pacotes disponíveis em seu projeto.

5. Depois de selecionar todos os repositórios que você deseja adicionar como repositórios upstream, selecione Selecionar e Salvar.

Para ter mais informações sobre como alterar a ordem de pesquisa de repositórios upstream, consulte [Edição da ordem de pesquisa dos repositórios upstream](#).

Depois de adicionar um repositório upstream, você pode usar um gerenciador de pacotes conectado ao seu repositório local para buscar pacotes do repositório upstream. Você não precisa atualizar a configuração do gerenciador de pacotes. Para ter mais informações sobre como solicitar versões de pacotes de um repositório upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Edição da ordem de pesquisa dos repositórios upstream

CodeCatalyst pesquisa repositórios upstream em sua ordem de pesquisa configurada. Quando um pacote é encontrado, CodeCatalyst interrompe a pesquisa. Você pode alterar a ordem na qual os repositórios upstream são pesquisados por pacotes.

Para editar a ordem de pesquisa dos repositórios upstream

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, escolha o repositório de pacotes cuja ordem de pesquisa do repositório upstream você deseja editar.
3. Abaixo do nome do repositório de pacotes, selecione Upstreams.
4. Na seção Repositórios upstream, você pode ver os repositórios upstream e a ordem de pesquisa. Para alterar a ordem de pesquisa, arraste e solte os repositórios na lista.
5. Quando terminar de editar a ordem de pesquisa dos repositórios upstream, escolha Salvar.

## Solicitar uma versão do pacote com repositórios upstream

O exemplo a seguir mostra os cenários possíveis quando um gerenciador de pacotes solicita um pacote de um repositório de CodeCatalyst pacotes que tenha repositórios upstream.

Neste exemplo, um gerenciador de pacotes, como npm, solicita uma versão de pacote de um repositório de pacotes chamado `downstream` que tem vários repositórios upstream. Quando o pacote é solicitado, o seguinte pode ocorrer:

- Se `downstream` contiver a versão do pacote solicitada, ela será devolvida ao cliente.

- Se `downstream` não contiver a versão do pacote solicitada, CodeCatalyst procurará por ela nos `downstream` repositórios `upstream`, na ordem de pesquisa configurada. Se a versão do pacote for encontrada, uma referência a ela será copiada para `downstream` e a versão do pacote será devolvida ao cliente.
- Se `downstream` nem seus repositórios `upstream` contiverem a versão do pacote, uma resposta HTTP 404 de `Not Found` será exibida ao cliente.

A quantidade máxima de repositórios `upstream` diretos permitidos para um repositório é dez. O número máximo de CodeCatalyst pesquisas nos repositórios quando uma versão do pacote é solicitada é 25.

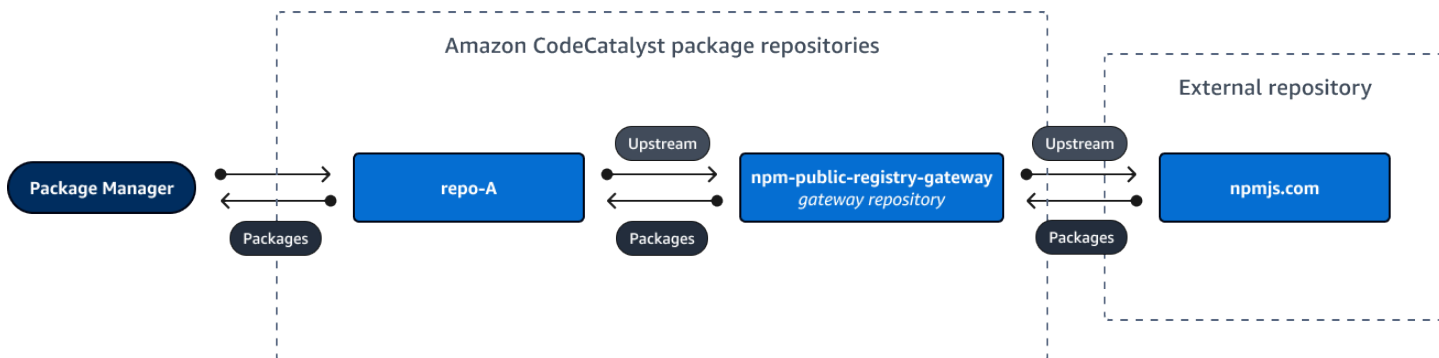
## Retenção de pacotes de repositórios upstream

Se uma versão de pacote solicitada for encontrada em um repositório `upstream`, uma referência a ela será retida e estará sempre disponível no repositório que a solicitou. Isso garante que você tenha acesso aos seus pacotes se houver uma interrupção inesperada do repositório `upstream`. A versão retida do pacote não é afetada por nenhum das seguintes ações:

- Excluir o repositório `upstream`.
- Desconectar o repositório `upstream` do repositório `downstream`.
- Excluir a versão do pacote do repositório `upstream`.
- Editar a versão do pacote no repositório `upstream` (por exemplo, adicionar um novo ativo a ele).

## Busca de pacotes por meio de uma relação upstream

CodeCatalyst pode buscar pacotes por meio de vários repositórios vinculados chamados repositórios `upstream`. Se um repositório de CodeCatalyst pacotes tiver uma conexão `upstream` com outro repositório de CodeCatalyst pacotes que tenha uma conexão `upstream` com um repositório `gateway`, as solicitações de pacotes que não estão no repositório `upstream` serão copiadas do repositório externo. Por exemplo, considere a seguinte configuração: um repositório chamado `repo-A` tem uma conexão `upstream` com o repositório do `gateway`, `npm-public-registry-gateway`. `npm-public-registry-gateway` tem uma conexão `upstream` com o repositório público de pacotes, <https://npmjs.com>.



Se npm estiver configurado para usar o repo-A repositório, a execução `npm install` inicia a cópia dos pacotes de <https://npmjs.com> dentro. `npm-public-registry-gateway` As versões instaladas também são incorporadas em repo-A. O exemplo a seguir instala o `lodash`.

```
$ npm config get registry
https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/
$ npm install lodash
+ lodash@4.17.20
added 1 package from 2 contributors in 6.933s
```

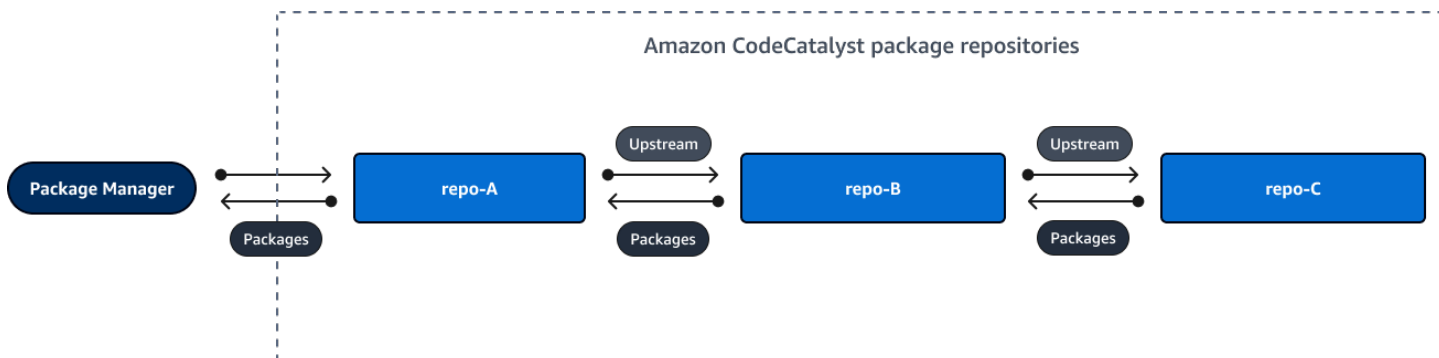
Depois de executar `npm install`, repo-A contém apenas a versão mais recente (`lodash 4.17.20`) porque essa é a versão que foi obtida por npm de repo-A.

Como `npm-public-registry-gateway` tem uma conexão upstream externa com <https://npmjs.com>, todas as versões do pacote importadas <https://npmjs.com> são armazenadas em `npm-public-registry-gateway`. Essas versões do pacote poderiam ter sido obtidas por qualquer repositório downstream com uma conexão upstream que leva a `npm-public-registry-gateway`.

O conteúdo de `npm-public-registry-gateway` fornece uma maneira de você ver todos os pacotes e versões de pacotes importados <https://npmjs.com> ao longo do tempo.

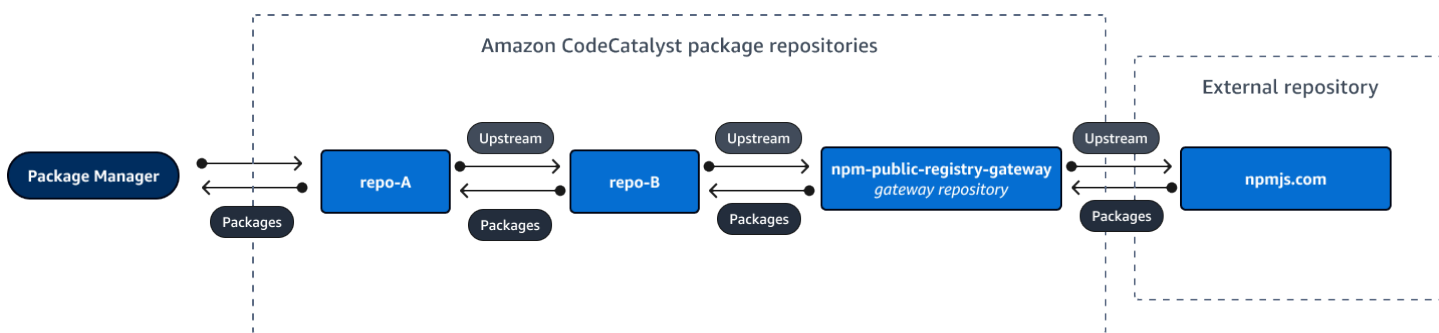
## Retenção de pacotes em repositórios intermediários

CodeCatalyst permite encadear repositórios upstream. Por exemplo, repo-A pode ter repo-B como repositório upstream e repo-B pode ter repo-C como repositório upstream. Essa configuração faz com que as versões do pacote sejam inseridas no repo-B e repo-C disponíveis do repo-A.



Quando um gerenciador de pacotes se conecta ao repositório **repo-A** e busca uma versão do pacote no repositório **repo-C**, a versão do pacote não será retida no repositório **repo-B**. A versão do pacote só será retida no repositório mais downstream, que, neste exemplo, é **repo-A**. Ela não será retida em nenhum repositório intermediário. Isso também vale para cadeias mais longas. Por exemplo, se houvesse quatro repositórios: **repo-A**, **repo-B**, **repo-C** e **repo-D** e um gerenciador de pacotes conectado ao **repo-A** buscasse uma versão do pacote no **repo-D**, a versão do pacote seria retida em **repo-A**, mas não em **repo-B** ou **repo-C**.

O comportamento de retenção do pacote é semelhante ao extrair uma versão do pacote de um repositório de pacotes público, exceto que a versão do pacote é sempre retida no repositório de gateway que tem a conexão upstream direta com o repositório público. Por exemplo, **repo-A** tem **repo-B** como repositório upstream. **repo-B** tem **npm-public-registry-gateway** como um repositório upstream, que tem uma conexão upstream com o repositório público, **npmjs.com**; veja o diagrama abaixo.



Se um gerenciador de pacotes conectado ao **repo-A** solicitar uma versão de pacote específica, por exemplo, **lodash 4.17.20**, e a versão do pacote não estiver presente em nenhum dos três repositórios, ela será obtida em **npmjs.com**. Quando o **lodash 4.17.20** for obtido, ele será retido no **repo-A**, pois é o repositório mais downstream, e no **npm-public-registry-gateway**, pois tem a conexão upstream com o repositório externo público **npmjs.com**. O **lodash 4.17.20** não será retido no **repo-B**, pois é um repositório intermediário.

## Remoção de um repositório upstream

Se você não quiser mais acessar os pacotes em um repositório upstream, poderá remover o repositório upstream de um repositório de pacotes.

### Warning

Ao remover um repositório upstream, você pode quebrar cadeias de relacionamento upstream, o que pode quebrar seus projetos ou compilações.

Para remover um repositório upstream

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, escolha o repositório de pacotes do qual você deseja remover um repositório upstream.
3. Abaixo do nome do repositório de pacotes, selecione Upstreams.
4. Em Editar repositórios upstream, localize o repositório upstream que você deseja remover e escolha .
5. Ao concluir a remoção dos repositórios upstream, selecione Salvar.

## Conexão a repositórios externos públicos

Você pode conectar repositórios de CodeCatalyst pacotes a repositórios externos públicos compatíveis adicionando o repositório de gateway correspondente como um repositório upstream. Os repositórios de gateway atuam como repositórios intermediários que pesquisam e armazenam pacotes extraídos de repositórios externos. Isso economiza tempo e transferência de dados porque todos os repositórios de pacotes em um projeto podem usar pacotes armazenados no repositório de gateway.

Para se conectar a um repositório público usando repositórios de gateway

1. No painel de navegação, selecione Packages (pacotes).
2. Em Pacotes, escolha a página Repositórios do gateway. Você pode ver uma lista dos repositórios de gateway compatíveis e das respectivas descrições.

3. Para usar um repositório de gateway, primeiro você deve criá-lo. Se o repositório de gateway tiver sido criado, a data e a hora em que ele foi criado serão mostradas. Caso contrário, selecione Criar para criá-lo.
4. Para usar pacotes do repositório do gateway, você deve definir uma conexão upstream com ele a partir de um CodeCatalyst repositório. Selecione Repositórios de pacotes e escolha o repositório de pacotes ao qual você deseja se conectar.
5. Para se conectar ao repositório público, selecione Upstreams e Selecionar repositórios upstream.
6. Selecione Repositórios de gateway e selecione o repositório de gateway que corresponde ao repositório público ao qual você deseja se conectar como um repositório upstream.
7. Depois de selecionar todos os repositórios de gateway que você deseja adicionar como repositórios upstream, escolha Selecionar.
8. Ao concluir o pedido dos repositórios upstream, selecione Salvar.

Para obter mais informações sobre como repositórios upstream, consulte [Configuração e uso de repositórios upstream](#).

Depois de adicionar um repositório de gateway como repositório upstream, você pode usar um gerenciador de pacotes conectado ao repositório local para buscar pacotes do repositório de pacotes externo público que corresponde a ele. Você não precisa atualizar a configuração do gerenciador de pacotes. Os pacotes consumidos dessa maneira são armazenados no repositório de gateway e no repositório de pacotes local. Para ter mais informações sobre como solicitar versões de pacotes de um repositório upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Repositórios de pacotes externos compatíveis e repositórios de gateway

CodeCatalyst suporta a adição de uma conexão upstream às seguintes autoridades oficiais de pacotes com repositórios de gateway.

| Tipo de pacote do repositório | Description          | Nome do repositório de gateway |
|-------------------------------|----------------------|--------------------------------|
| npm                           | Registro npm público | npm-public-registry-gateway    |
| Python                        | Python Package Index | pypi-gateway                   |

| Tipo de pacote do repositório | Description                       | Nome do repositório de gateway |
|-------------------------------|-----------------------------------|--------------------------------|
| Maven                         | Maven Central                     | maven-central-gateway          |
| Maven                         | Repositório do Google Android     | google-android-gateway         |
| Maven                         | CommonsWare                       | commonsware-gateway            |
| Maven                         | Repositório de plug-ins do Gradle | gradle-plugins-gateway         |
| NuGet                         | NuGet Galeria                     | nuget-gallery-gateway          |

## Publicação e modificação de pacotes

Um pacote CodeCatalyst é um pacote de software e os metadados necessários para resolver dependências e instalar o software. Para obter uma lista dos formatos de pacote compatíveis em CodeCatalyst, consulte [Publique e compartilhe pacotes de software no CodeCatalyst](#). Esta seção fornece informações sobre publicação, visualização, exclusão de pacotes e atualização do status da versão de um pacote.

### Tópicos

- [Publicando pacotes em um repositório de CodeCatalyst pacotes](#)
- [Visualização de detalhes da versão de pacote](#)
- [Exclusão de uma versão de pacote](#)
- [Atualização do status da versão do pacote](#)
- [Editar controles de origem do pacote](#)



## Publicando pacotes em um repositório de CodeCatalyst pacotes

Você pode publicar versões de qualquer tipo de pacote compatível em um repositório de CodeCatalyst pacotes usando as ferramentas do gerenciador de pacotes. As etapas para publicar uma versão do pacote são as seguintes:

Para publicar uma versão de pacote em um repositório de CodeCatalyst pacotes

1. Caso contrário, [crie um repositório de pacotes](#).
2. Conecte o gerenciador de pacotes ao repositório de pacotes. Para obter instruções sobre como conectar o gerenciador de pacotes npm a um repositório de CodeCatalyst pacotes, consulte [Configuração e uso de npm](#).
3. Use o gerenciador de pacotes conectado para publicar as versões de pacotes.

### Sumário

- [Repositórios de publicação e upstream](#)
- [Pacotes privados e repositórios públicos](#)
- [Substituindo ativos do pacote](#)

## Repositórios de publicação e upstream

Em CodeCatalyst, você não pode publicar versões de pacotes que estejam presentes em repositórios upstream acessíveis ou em repositórios públicos. Por exemplo, suponha que você queira publicar um pacote npm, `lodash@1.0`, em um repositório de pacotes, `myrepo`, e `myrepo` esteja conectado a `npmjs.com` por meio de um repositório de gateway configurado como um repositório upstream. Se `lodash@1.0` estiver presente no repositório upstream ou em `npmjs.com`, CodeCatalyst rejeita qualquer tentativa de publicar nele emitindo um erro de conflito 409. `myrepo` Isso ajuda a evitar que você publique acidentalmente um pacote com o mesmo nome e a mesma versão de um pacote em um repositório upstream, o que pode resultar em um comportamento inesperado.

Você ainda pode publicar versões diferentes de um nome de pacote que existam em um repositório upstream. Por exemplo, se `lodash@1.0` estiver presente em um repositório upstream, mas `lodash@1.1` não estiver, você poderá publicar `lodash@1.1` no repositório downstream.

## Pacotes privados e repositórios públicos

CodeCatalyst não publica pacotes armazenados em CodeCatalyst repositórios em repositórios públicos, como npmjs.com ou Maven Central. CodeCatalyst importa pacotes de repositórios públicos para um CodeCatalyst repositório, mas não move pacotes na direção oposta. Os pacotes que você publica CodeCatalyst nos repositórios permanecem privados e só estão disponíveis para o CodeCatalyst projeto ao qual o repositório pertence.

## Substituindo ativos do pacote

Você não pode republicar um ativo de pacote que já existe com conteúdo diferente. Por exemplo, suponha que você já tenha publicado um pacote Maven com um ativo `mypackage-1.0.jar` JAR. Você só poderá publicar esse ativo outra vez se a soma de verificação dos ativos antigos e novos for idêntica. Para republicar o mesmo ativo com novo conteúdo, primeiro exclua a versão do pacote. Tentar republicar o mesmo nome de ativo com conteúdo diferente resultará em um erro de conflito HTTP 409.

Para formatos de pacote compatíveis com vários ativos (Python e Maven), você pode adicionar novos ativos com nomes diferentes a uma versão de pacote existente a qualquer momento, supondo que você tenha as permissões necessárias. Como o npm suporta NuGet apenas um único ativo por versão do pacote, para modificar uma versão publicada do pacote, você deve primeiro excluí-la.

Se você tentar republicar um ativo que já existe (por exemplo, `mypackage-1.0.jar`) e o conteúdo do ativo publicado e o do novo ativo forem idênticos, a operação será bem-sucedida porque é idempotente.

## Visualização de detalhes da versão de pacote

Você pode usar o CodeCatalyst console para ver detalhes sobre uma versão específica do pacote.

Para visualizar detalhes da versão de pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, selecione o repositório que contém a versão do pacote cujos detalhes você deseja visualizar.
3. Pesquise a versão do pacote na tabela Pacotes. Você pode usar a barra de pesquisa para filtrar pacotes por nome e formato de pacote. Escolha o pacote na lista.
4. Na página Detalhes do pacote, selecione Versões e então selecione a versão que você deseja visualizar.

## Exclusão de uma versão de pacote

Você pode excluir uma versão do pacote na página de detalhes da versão do pacote no CodeCatalyst console.

Para excluir uma versão de pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, selecione o repositório que contém a versão do pacote que você deseja excluir.
3. Pesquise e selecione o pacote na tabela.
4. Na página Detalhes do pacote, selecione Versões e selecione a versão que deseja excluir.
5. Na página Detalhes da versão do pacote, selecione Ações da versão e Excluir.
6. Insira excluir no campo de texto e selecione Excluir.

## Atualização do status da versão do pacote

Cada versão do pacote CodeCatalyst tem um status que descreve o estado atual e a disponibilidade da versão do pacote. Você pode alterar o status da versão do pacote no CodeCatalyst console. Para ter mais informações sobre os possíveis valores de status das versões do pacote e seus significados, consulte [Satus da versão do pacote](#).

Como atualizar o status da versão do pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Repositórios de pacotes, selecione o repositório que contém a versão do pacote cujo status você deseja atualizar.
3. Pesquise e selecione o pacote na tabela.
4. Na página Detalhes do pacote, selecione Versões e então selecione a versão que você deseja visualizar.
5. Na página Detalhes da versão do pacote, selecione Ações e selecione Remover da lista, Arquivar ou Descartar. Para ter mais informações sobre o status da versão do pacote, consulte [Satus da versão do pacote](#).
6. Insira o texto de confirmação no campo de texto e selecione Remover da lista, Arquivar ou Descartar, dependendo do status para o qual você está atualizando.

## Satus da versão do pacote

Os seguintes valores de status da versão do pacote são possíveis. Você pode alterar o status da versão do pacote no console. Para obter mais informações, consulte [Atualização do status da versão do pacote](#).

- **Publicado:** a versão do pacote foi publicada com sucesso e pode ser solicitada por um gerenciador de pacotes. A versão do pacote será incluída nas listas de versões de pacotes retornadas aos gerentes de pacotes, por exemplo, na saída de `npm view <package-name> versions`. Todos os ativos da versão do pacote estão disponíveis no repositório.
- **Não concluído:** a última tentativa de publicação não foi concluída. No momento, apenas as versões do pacote Maven podem ter o status de Não concluído. Isso pode ocorrer quando o cliente carrega um ou mais ativos para uma versão do pacote, mas não publica um arquivo `maven-metadata.xml` para o pacote que inclui essa versão.
- **Não listado:** os ativos da versão do pacote estão disponíveis para download no repositório, mas a versão do pacote não está incluída na lista de versões retornadas aos gerenciadores de pacotes. Por exemplo, para um pacote npm, a saída de `npm view <package-name> versions` não inclui a versão do pacote. Isso significa que a lógica de resolução de dependências do npm não seleciona a versão do pacote porque a versão não aparece na lista de versões disponíveis. No entanto, se a versão do pacote Não listado já estiver referenciada em um arquivo `npm package-lock.json`, ela ainda poderá ser baixada e instalada, por exemplo, durante a execução de `npm ci`.
- **Arquivado:** os ativos da versão do pacote não podem ser baixados. A versão do pacote não será incluída na lista de versões retornada aos gerenciadores de pacotes. Como os ativos não estão disponíveis, o consumo da versão do pacote pelos clientes é bloqueado. Se a compilação da aplicação depender de uma versão atualizada para Arquivado, a compilação falhará, a menos que a versão do pacote tenha sido armazenada em cache localmente. Você não pode usar um gerenciador de pacotes ou uma ferramenta de criação para republicar uma versão de pacote Arquivado porque ela ainda está presente no repositório. No entanto, você pode alterar o status da versão do pacote de volta para Não listado ou Publicado no console.
- **Descartada:** a versão do pacote não aparece nas listagens e os ativos não podem ser baixados do repositório. A principal diferença entre descartado e arquivado é que, com o status de descartado, os ativos da versão do pacote são excluídos permanentemente pelo CodeCatalyst. Por esse motivo, você não pode mover uma versão de pacote de Descartada para Arquivada, Não Listada ou Publicada. A versão do pacote não pode ser usada porque os ativos foram excluídos.

Quando uma versão do pacote for marcada como Descartada, você não será cobrado pelo armazenamento dos ativos do pacote.

Além dos status na lista anterior, uma versão do pacote também pode ser excluída. Depois de ser excluída, uma versão do pacote não está no repositório e você pode republicá-la livremente usando um gerenciador de pacotes ou uma ferramenta de compilação.

## Normalização do nome e da versão do pacote e do nome do ativo

CodeCatalyst normaliza nomes de pacotes, versões de pacotes e nomes de ativos antes de armazená-los, o que significa que os nomes ou versões CodeCatalyst podem ser diferentes do nome ou versão fornecidos quando o pacote foi publicado. Para obter mais informações sobre como os nomes e as versões são normalizados CodeCatalyst para cada tipo de pacote, consulte a documentação a seguir.

- [Normalização do nome do pacote Python](#)
- [NuGet normalização do nome do pacote, versão e nome do ativo](#)

CodeCatalyst não executa a normalização em outros formatos de pacote.

## Editar controles de origem do pacote

Na Amazon CodeCatalyst, as versões de pacotes podem ser adicionadas a um repositório de pacotes publicando-as diretamente, retirando-as de um repositório upstream ou ingerindo-as de um repositório público externo por meio de um gateway. Se você permitir que versões de um pacote sejam adicionadas por publicação direta e ingestão de repositórios públicos, ficará vulnerável a um ataque de substituição de dependências. Para obter mais informações, consulte [Ataques de substituição de dependências](#). Para se proteger contra um ataque de substituição de dependência, configure os controles de origem do pacote em um pacote em um repositório para limitar como as versões desse pacote podem ser adicionadas ao repositório.

Considere a possibilidade de configurar os controles de origem do pacote para permitir que novas versões de pacotes diferentes venham tanto de fontes internas, como publicação direta, quanto de fontes externas, como repositórios públicos. Por padrão, os controles de origem do pacote são configurados com base em como a primeira versão de um pacote é adicionada ao repositório.

## Configurações de controle de origem do pacote

Com os controles de origem do pacote, você pode configurar como as versões do pacote podem ser adicionadas a um repositório. As listas a seguir incluem as configurações e os valores disponíveis do controle de origem do pacote.

### Publicar

Essa configuração define se as versões do pacote podem ser publicadas diretamente no repositório usando gerenciadores de pacotes ou ferramentas similares.

- PERMITIR: as versões do pacote podem ser publicadas diretamente.
- BLOCK: as versões do pacote não podem ser publicadas diretamente.

### Upstream

Essa configuração define se as versões do pacote podem ser ingeridas de repositórios externos públicos ou retidas de repositórios upstream quando solicitadas por um gerenciador de pacotes.

- PERMITIR: Qualquer versão do pacote pode ser retida de outros CodeCatalyst repositórios configurados como repositórios upstream ou ingerida de uma fonte pública com uma conexão externa.
- BLOCO: As versões do pacote não podem ser retidas de outros CodeCatalyst repositórios configurados como repositórios upstream ou ingeridas de uma fonte pública com uma conexão externa.

### Configurações de controle de origem do pacote padrão

Os controles de origem do pacote padrão de um pacote serão baseados em como a primeira versão desse pacote é adicionada ao repositório de pacotes.

- Se a primeira versão do pacote for publicada diretamente por um gerenciador de pacotes, as configurações serão Publicação: PERMITIR e Upstream: BLOQUEAR.
- Se a primeira versão do pacote for ingerida de uma fonte pública, as configurações serão Publicação: BLOQUEAR e Upstream: PERMITIR.

## Cenários comuns de controle de acesso a pacotes

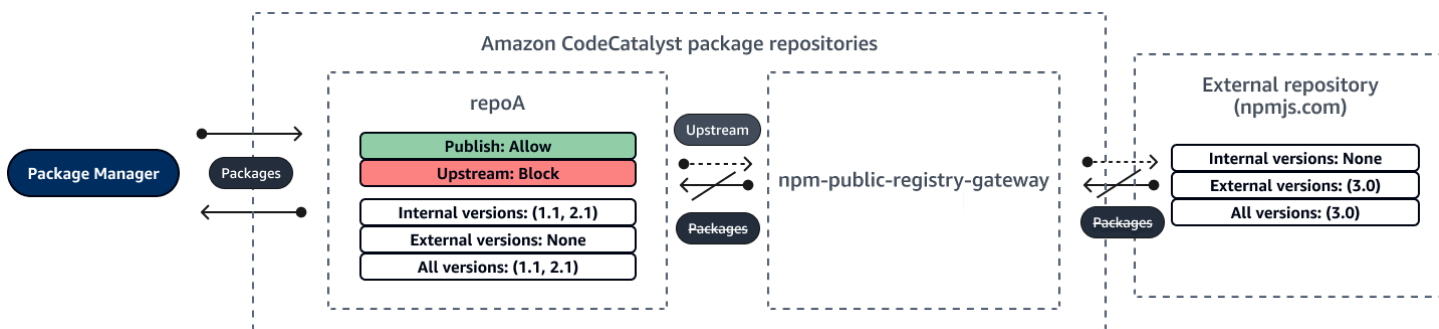
Esta seção descreve alguns cenários comuns de quando uma versão de pacote é adicionada a um repositório de CodeCatalyst pacotes. As configurações de controle de origem do pacote são definidas para novos pacotes, dependendo de como a primeira versão do pacote for adicionada.

Nos cenários a seguir, um pacote interno é publicado de um gerenciador de pacotes no seu repositório, como um pacote que você mantém. Um pacote externo é um pacote que existe em um repositório público que pode ser ingerido em seu repositório por meio de um repositório de gateway upstream.

Uma versão de pacote externo é publicada para um pacote interno existente

Nesse cenário, considere um pacote interno, packageA. Sua equipe publica a primeira versão do pacote do PackageA em um repositório de pacotes. CodeCatalyst Como essa é a primeira versão desse pacote, as configurações de controle de origem do pacote são automaticamente definidas como **Publicar: Permitir** e **Upstream: Bloquear**. Depois que o pacote é publicado no seu repositório, um pacote com o mesmo nome é publicado em um repositório público conectado ao seu repositório de CodeCatalyst pacotes. Isso pode ser uma tentativa de ataque de substituição de dependência contra o pacote interno ou pode ser uma coincidência. Independentemente disso, os controles de origem do pacote são configurados para bloquear a ingestão da nova versão externa para se protegerem contra um possível ataque.

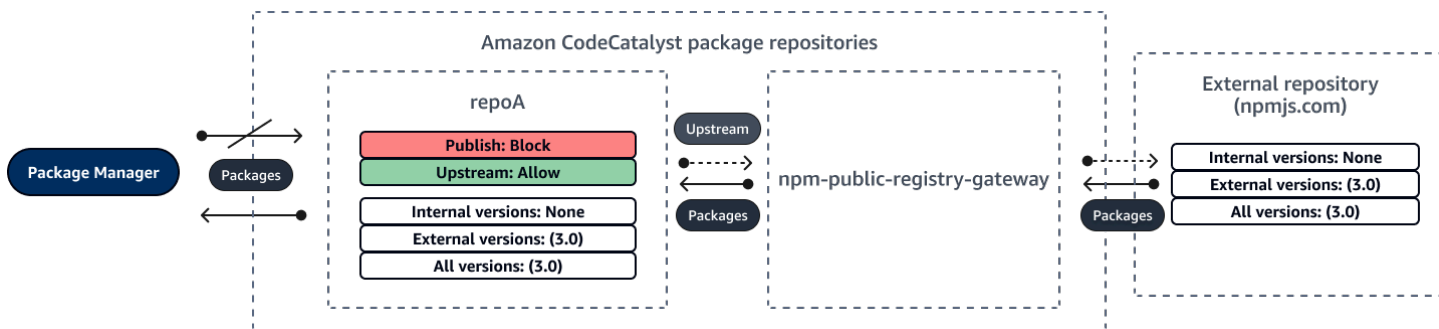
Na imagem a seguir, o RepoA é seu repositório de CodeCatalyst pacotes com uma conexão upstream com o repositório. npm-public-registry-gateway Seu repositório contém as versões 1.1 e 2.1 de packageA, mas a versão 3.0 é publicada no repositório público. Normalmente, repoA poderia ingerir a versão 3.0 depois que o pacote fosse solicitado por um gerenciador de pacotes. Como a ingestão de pacotes está definida como **Bloquear**, a versão 3.0 não é ingerida em seu repositório de CodeCatalyst pacotes e não está disponível para gerenciadores de pacotes conectados a ele.



Uma versão de pacote interno é publicada para um pacote externo existente

Nesse cenário, um pacote, packageB, existe externamente em um repositório público que você conectou ao seu repositório. Quando um gerenciador de pacotes conectado ao seu repositório solicita packageB, a versão do pacote é ingerida no seu repositório a partir do repositório público. Como esta é a primeira versão do pacote de packageB adicionada ao seu repositório, as configurações de origem do pacote são definidas como Publicação: BLOQUEAR e Upstream: PERMITIR. Mais tarde, você tenta publicar uma versão com o mesmo nome de pacote no repositório. Você pode não estar ciente do pacote público e estar tentando publicar um pacote não relacionado com o mesmo nome, ou pode estar tentando publicar uma versão corrigida ou pode estar tentando publicar diretamente a versão exata do pacote que já existe externamente. CodeCatalyst rejeita a versão que você está tentando publicar, mas você pode explicitamente substituir a rejeição e publicar a versão, se necessário.

Na imagem a seguir, o RepoA é seu repositório de CodeCatalyst pacotes com uma conexão upstream com o repositório. npm-public-registry-gateway Seu repositório de pacotes contém a versão 3.0 que foi ingerida do repositório público. Você deseja publicar a versão 1.2 no seu repositório de pacotes. Normalmente, você poderia publicar a versão 1.2 em repoA, mas como a publicação está definida como Bloquear, a versão 1.2 não pode ser publicada.



## Publicar uma versão corrigida de um pacote externo existente

Nesse cenário, um pacote, packageB, existe externamente em um repositório público que você conectou ao seu repositório de pacotes. Quando um gerenciador de pacotes conectado ao seu repositório solicita packageB, a versão do pacote é ingerida no seu repositório a partir do repositório público. Como esta é a primeira versão do pacote de packageB adicionada ao seu repositório, as configurações de origem do pacote são definidas como Publicação: BLOQUEAR e Upstream: PERMITIR. Sua equipe decide publicar versões corrigidas desse pacote no repositório. Para poder publicar versões de pacotes diretamente, sua equipe altera as configurações de controle de origem do pacote para Publicação: PERMITIR e Upstream: BLOQUEAR. Agora, as versões desse pacote podem ser publicadas diretamente no seu repositório e ingeridas de repositórios públicos. Depois que sua equipe publica as versões corrigidas do pacote, ela reverte as configurações de origem do pacote para Publicação: BLOQUEAR e Upstream: PERMITIR.



## Editar controles de origem do pacote

Os controles de origem do pacote são configurados automaticamente com base em como a primeira versão de um pacote é adicionada ao repositório de pacotes. Para obter mais informações, consulte [Configurações de controle de origem do pacote padrão](#). Para adicionar ou editar controles de origem de pacote para um pacote em um repositório de CodeCatalyst pacotes, execute as etapas no procedimento a seguir.

Como adicionar ou editar controles de origem do pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Escolha o repositório de pacotes que contém o pacote que você deseja editar.
3. Na tabela Pacotes, procure e selecione o pacote que você deseja editar.
4. Na página de resumo do pacote, selecione Controles de origem.
5. Em Controles de origem, escolha os controles de origem do pacote que deseja definir para esse pacote. As duas configurações de controle de origem do pacote, Publicar e Upstream, devem ser definidas ao mesmo tempo.
  - Para permitir a publicação direta de versões do pacote, em Publicação, escolha Permitir. Para bloquear a publicação de versões do pacote, escolha Bloquear.
  - Para permitir a ingestão de pacotes de repositórios externos e a extração de pacotes de repositórios upstream, em Fontes upstream, escolha Permitir. Para bloquear toda a ingestão e extração de versões de pacotes de repositórios externos e upstream, escolha Bloquear.
6. Escolha Salvar.

## Repositórios de publicação e upstream

Em CodeCatalyst, você não pode publicar versões de pacotes que estejam presentes em repositórios upstream acessíveis ou em repositórios públicos. Por exemplo, suponha que você queira publicar um pacote npm `lodash@1.0` em um repositório, `myrepo`, e `myrepo` tenha um repositório upstream com uma conexão externa com `npmjs.com`. Considere os seguintes cenários:

1. As configurações de controle de origem do pacote em `lodash` são Publicação: PERMITIR e Upstream: PERMITIR. Se `lodash@1.0` estiver presente no repositório upstream ou em `npmjs.com`, CodeCatalyst rejeita qualquer tentativa de publicar nele emitindo um erro de conflito 409. `myrepo` Você ainda pode publicar uma versão diferente, como `lodash@1.1`.

2. As configurações de controle de origem do pacote em `lodash` são Publicação: PERMITIR e Upstream: BLOQUEAR. Você pode publicar qualquer versão do `lodash` no seu repositório que ainda não exista porque as versões do pacote não estão acessíveis.
3. As configurações de controle de origem do pacote em `lodash` são Publicação: BLOQUEAR e Upstream: PERMITIR. Você não pode publicar nenhuma versão do pacote diretamente no seu repositório.

## Ataques de substituição de dependências

Os gerenciadores de pacotes simplificam o processo de criação de pacotes e compartilhamento do código reutilizável. Esses pacotes podem ser pacotes privados desenvolvidos por uma organização para uso em seus aplicativos ou públicos, geralmente pacotes de código aberto, desenvolvidos fora de uma organização e distribuídos por repositórios públicos de pacotes. Quando solicitam pacotes, os desenvolvedores dependem do gerenciador de pacotes para buscar novas versões de suas dependências. Os ataques de substituição de dependência, também conhecidos como ataques de confusão de dependência, exploram o fato de que um gerenciador de pacotes normalmente não tem como distinguir as versões legítimas de um pacote de versões maliciosas.

Os ataques de substituição de dependência pertencem a um subconjunto de ataques conhecido como ataques à cadeia de suprimentos de software. Um ataque à cadeia de suprimentos de software é um ataque que tira proveito de vulnerabilidades em qualquer ponto da cadeia de suprimentos de software.

Um ataque de substituição de dependência pode ter como alvo qualquer pessoa que use tanto pacotes desenvolvidos internamente quanto aqueles obtidos de repositórios públicos. Os invasores identificam nomes de pacotes internos e, depois, colocam estrategicamente códigos maliciosos com o mesmo nome em repositórios públicos de pacotes. Normalmente, o código malicioso é publicado em um pacote com um número de versão alto. Os gerenciadores de pacotes buscam o código malicioso desses feeds públicos porque acreditam que os pacotes maliciosos são as versões mais recentes do pacote. Isso causa uma “substituição” ou “confusão” entre o pacote desejado e o pacote malicioso, fazendo com que o código seja comprometido.

Para evitar ataques de substituição de dependências, a Amazon CodeCatalyst fornece controles de origem de pacotes. Os controles de origem de pacote são configurações que controlam como os pacotes podem ser adicionados aos seus repositórios. Os controles são configurados automaticamente quando a primeira versão do pacote de um novo pacote é adicionada a um CodeCatalyst repositório. Os controles podem garantir que as versões do pacote não possam ser

publicadas diretamente no seu repositório e ingeridas de fontes públicas, protegendo você contra ataques de substituição de dependência. Para obter mais informações sobre os controles de origem de pacotes e como alterá-los, consulte [Editar controles de origem do pacote](#).

## Uso de npm

Esses tópicos descrevem como você pode usar npm o gerenciador de pacotes Node.js com CodeCatalyst.

### Note

CodeCatalyst suportes node v4.9.1 e mais tarde npm v5.0.0 e mais tarde.

### Tópicos

- [Configuração e uso de npm](#)
- [Tratamento de tags npm](#)

## Configuração e uso de npm

Para usar npm com CodeCatalyst, você deve se conectar npm ao seu repositório de pacotes e fornecer um token de acesso pessoal (PAT) para autenticação. Você pode ver as instruções para se conectar npm ao seu repositório de pacotes no CodeCatalyst console.

### Sumário

- [Configurando o npm com CodeCatalyst](#)
- [Instalando pacotes npm a partir de um repositório de CodeCatalyst pacotes](#)
- [Instalando pacotes npm do npmjs por meio de CodeCatalyst](#)
- [Publicando pacotes npm em seu repositório de CodeCatalyst pacotes](#)
- [Suporte para comandos npm](#)
  - [Comandos compatíveis que interagem com um repositório de pacotes](#)
  - [Comandos do lado do cliente compatíveis](#)
  - [Comandos incompatíveis](#)

## Configurando o npm com CodeCatalyst

As instruções a seguir explicam como se autenticar e se conectar npm ao seu repositório de CodeCatalyst pacotes. Para ter mais informações sobre o npm, consulte a [documentação oficial do npm](#).

Para se conectar **npm** ao seu repositório de CodeCatalyst pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto.
3. No painel de navegação, selecione Packages (pacotes).
4. Escolha o repositório de pacotes na lista.
5. Selecione Conectar ao repositório.
6. Em Detalhes da configuração, no Cliente do Package Manager, selecione Cliente do npm.
7. Escolha o sistema operacional para ver as etapas de configuração correspondentes.
8. É necessário um token de acesso pessoal (PAT) para autenticar o npm com. CodeCatalyst Se você já possui um token, poderá usá-lo. Caso contrário, você pode criar um usando as etapas a seguir.
  - a. (Opcional): atualize o Nome do PAT e a Data de expiração.
  - b. Selecione Criar token.
  - c. Copie e guarde o PAT em um local seguro.

### Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo. As credenciais devem ser de curta duração para minimizar o tempo que um invasor pode usar as credenciais após usá-las indevidamente.

9. Execute os comandos a seguir no diretório raiz do projeto para configurar o npm com seu repositório de pacotes. Os comandos farão o seguinte:
  - Crie um arquivo `.npmrc` no nível do projeto se o seu projeto não tiver um.
  - Adicione as informações do endpoint do repositório de pacotes ao arquivo `.npmrc` no nível do projeto.
  - Adicione suas credenciais (PAT) ao arquivo `.npmrc` no nível do usuário.

Substitua os valores a seguir.

 Note

Se você estiver copiando as instruções do console, os valores nos comandos a seguir serão atualizados e não precisarão ser alterados.

- *username* Substitua CodeCatalyst pelo seu nome de usuário.
- *PAT* Substitua por seu CodeCatalyst PAT.
- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
npm set registry=https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/ --location project
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:_authToken=username:PAT
```

Para npm 6 ou inferior: para fazer com que o npm sempre passe o token de autenticação para CodeCatalyst, mesmo para GET solicitações, defina a variável de configuração always-auth da seguinte forma. `npm config set`

```
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:always-auth=true --location project
```

## Instalando pacotes npm a partir de um repositório de CodeCatalyst pacotes

Depois de conectar o npm ao seu repositório seguindo as etapas em [Configurando o npm com CodeCatalyst](#), você pode executar comandos npm no seu repositório.

Você pode instalar um pacote npm que esteja no seu repositório de CodeCatalyst pacotes ou em um de seus repositórios upstream com o comando. `npm install`

```
npm install Lodash
```

## Instalando pacotes npm do npmjs por meio de CodeCatalyst

Você pode instalar pacotes npm do [npmjs.com](https://www.npmjs.com) por meio de um repositório configurando o CodeCatalyst repositório com uma conexão upstream com o repositório do gateway conectado ao npmjs.com,. npm-public-registry-gateway Os pacotes instalados em npmjs são ingeridos e armazenados no repositório de gateway e no repositório de pacotes downstream mais distante.

Para instalar pacotes do npmjs

1. Se você ainda não tiver feito isso, configure npm com seu repositório de CodeCatalyst pacotes seguindo as etapas em [Configurando o npm com CodeCatalyst](#).
2. Verifique se seu repositório adicionou o repositório do gateway, npm-public-registry-gateway, como uma conexão upstream. Você pode verificar quais fontes upstream foram adicionadas ou adicionadas npm-public-registry-gateway como fonte upstream seguindo as instruções [Adição de um repositório upstream](#) e escolhendo o npm-public-registry-gateway repositório.
3. Instale pacotes com o comando `npm install`.

```
npm install package_name
```

Para ter mais informações sobre como solicitar pacotes de repositórios upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Publicando pacotes npm em seu repositório de CodeCatalyst pacotes

Depois de concluir [Configurando o npm com CodeCatalyst](#), você pode executar comandos npm.

Você pode publicar um pacote npm em um repositório de CodeCatalyst pacotes com o `npm publish` comando.

```
npm publish
```

Para ter informações sobre como criar pacotes do npm, consulte [Criação de módulos Node.js](#) no Documento do npm.

## Suporte para comandos npm

As seções a seguir resumem os npm comandos que são suportados pelos repositórios de CodeCatalyst pacotes, além de listar comandos específicos que não são suportados.

### Tópicos

- [Comandos compatíveis que interagem com um repositório de pacotes](#)
- [Comandos do lado do cliente compatíveis](#)
- [Comandos incompatíveis](#)

### Comandos compatíveis que interagem com um repositório de pacotes

Esta seção lista os comandos npm em que o cliente npm faz uma ou mais solicitações ao registro no qual foi configurado (por exemplo, `npm config set registry`). Verificou-se que esses comandos funcionam corretamente quando invocados em um repositório de CodeCatalyst pacotes.

| Command                   | Description                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">bugs</a>      | Adivinha a localização do URL do rastreador de bugs de um pacote e, depois, tenta abri-lo.                                                      |
| <a href="#">ci</a>        | Instala um projeto do zero.                                                                                                                     |
| <a href="#">deprecate</a> | Deprecia uma versão de um pacote.                                                                                                               |
| <a href="#">dist-tag</a>  | Modifica as tags de distribuição do pacote.                                                                                                     |
| <a href="#">docs</a>      | Adivinha a localização do URL de documentação de um pacote e, depois, tenta abri-lo usando o parâmetro de configuração <code>--browser</code> . |
| <a href="#">doctor</a>    | Executa um conjunto de verificações para validar se sua instalação do npm pode gerenciar seus JavaScript pacotes.                               |
| <a href="#">install</a>   | Instala um pacote.                                                                                                                              |

| Command                         | Description                                                                                                                                                                  |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">install-ci-test</a> | Instala um projeto do zero e executa testes. Alias: <code>npm ci</code> . Esse comando executa um <code>npm ci</code> , seguido imediatamente por um <code>npm test</code> . |
| <a href="#">install-test</a>    | Instala o pacote e executa testes. Executa um <code>npm install</code> , seguido imediatamente por um <code>npm test</code> .                                                |
| <a href="#">outdated</a>        | Verifica o registro configurado para determinar se algum pacote instalado está desatualizado.                                                                                |
| <a href="#">ping</a>            | Faz ping no registro npm configurado ou fornecido e verifica a autenticação.                                                                                                 |
| <a href="#">publish</a>         | Publica uma versão do pacote no registro.                                                                                                                                    |
| <a href="#">update</a>          | Adivinha a localização do URL do repositório de um pacote e, depois, tenta abri-lo usando o parâmetro de configuração <code>--browser</code> .                               |
| <a href="#">view</a>            | Exibe os metadados do pacote. Também pode ser usado para imprimir propriedades de metadados.                                                                                 |

## Comandos do lado do cliente compatíveis

Esses comandos não exigem nenhuma interação direta com um repositório de pacotes, portanto, CodeCatalyst não exige nada para suportá-los.

| Command                      | Description                              |
|------------------------------|------------------------------------------|
| <a href="#">bin (legado)</a> | Exibe o diretório npm <code>bin</code> . |
| <a href="#">build</a>        | Cria um pacote.                          |



| Command                     | Description                                                                                                                                                                                                 |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">cache</a>       | Manipula o cache de pacotes.                                                                                                                                                                                |
| <a href="#">completion</a>  | Ativa o preenchimento de guias em todos os comandos npm.                                                                                                                                                    |
| <a href="#">config</a>      | Atualiza o conteúdo do usuário e dos arquivos npmrc globais.                                                                                                                                                |
| <a href="#">dedupe</a>      | Pesquisa a árvore de pacotes local e tenta simplificar a estrutura movendo as dependências mais para cima na árvore, onde elas podem ser compartilhadas com mais eficiência por vários pacotes dependentes. |
| <a href="#">edit</a>        | Edita um pacote instalado. Seleciona uma dependência no diretório de trabalho atual e abre o diretório do pacote no editor padrão.                                                                          |
| <a href="#">explore</a>     | Navega por um pacote instalado. Gera um subshell no diretório do pacote instalado especificado. Se um comando for especificado, ele será executado no subshell, que será encerrado imediatamente.           |
| <a href="#">help</a>        | Recebe ajuda no npm.                                                                                                                                                                                        |
| <a href="#">help-search</a> | Pesquisa a documentação de ajuda do npm.                                                                                                                                                                    |
| <a href="#">init</a>        | Cria um arquivo package.json.                                                                                                                                                                               |
| <a href="#">link</a>        | Vincula um diretório de pacotes.                                                                                                                                                                            |
| <a href="#">ls</a>          | Lista pacotes instalados.                                                                                                                                                                                   |
| <a href="#">pack</a>        | Cria um tarball a partir de um pacote.                                                                                                                                                                      |

| Command                    | Description                                                                                                                                                   |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">prefix</a>     | Exibe um prefixo. Esse é o diretório pai mais próximo que contém um arquivo <code>package.json</code> , a menos que <code>-g</code> também seja especificado. |
| <a href="#">prune</a>      | Remove pacotes que não estão listados na lista de dependências do pacote pai.                                                                                 |
| <a href="#">rebuild</a>    | Executa o comando <code>npm build</code> nas pastas correspondentes.                                                                                          |
| <a href="#">restart</a>    | Executa os scripts de parada, reinicialização e inicialização de um pacote e os scripts anteriores e posteriores associados.                                  |
| <a href="#">raiz</a>       | Imprime o diretório <code>node_modules</code> efetivo na saída padrão.                                                                                        |
| <a href="#">run-script</a> | Executa scripts de pacotes arbitrários.                                                                                                                       |
| <a href="#">shrinkwrap</a> | Bloqueia as versões de dependência para publicação.                                                                                                           |
| <a href="#">uninstall</a>  | Desinstala um pacote.                                                                                                                                         |

## Comandos incompatíveis

Esses npm comandos não são compatíveis com repositórios de CodeCatalyst pacotes.

| Command                | Description                                      | Observações                                                                     |
|------------------------|--------------------------------------------------|---------------------------------------------------------------------------------|
| <a href="#">access</a> | Define o nível de acesso nos pacotes publicados. | CodeCatalyst usa um modelo de permissão diferente do repositório público npmjs. |

| Command                        | Description                                                               | Observações                                                                                                                                                                                                                                                |
|--------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>adduser</u></a> | Adiciona uma conta de usuário de registro                                 | CodeCatalyst usa um modelo de usuário diferente do repositório público npmjs.                                                                                                                                                                              |
| <a href="#"><u>audit</u></a>   | Executa uma auditoria de segurança.                                       | CodeCatalyst atualmente não vende dados de vulnerabilidade de segurança.                                                                                                                                                                                   |
| <a href="#"><u>hook</u></a>    | Gerencia hooks do npm, incluindo adição, remoção, listagem e atualização. | CodeCatalyst atualmente não oferece suporte a nenhum mecanismo de notificação de alterações.                                                                                                                                                               |
| <a href="#"><u>login</u></a>   | Autentica um usuário. Este é um alias para npm adduser.                   | CodeCatalyst usa um modelo de autenticação diferente do repositório público npmjs. Para mais informações, consulte <a href="#">Configurando o npm com CodeCatalyst</a> .                                                                                   |
| <a href="#"><u>logout</u></a>  | Sai do registro.                                                          | CodeCatalyst usa um modelo de autenticação diferente do repositório público npmjs. Não há como sair de um CodeCatalyst repositório, mas os tokens de autenticação expiram após o tempo de expiração configurável. A duração padrão do token é de 12 horas. |
| <a href="#"><u>owner</u></a>   | Gerencia proprietários de pacotes.                                        | CodeCatalyst usa um modelo de permissões diferente do repositório público npmjs.                                                                                                                                                                           |

| Command                   | Description                                                               | Observações                                                                                                                                                   |
|---------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">profile</a>   | Altera as configurações no seu perfil de registro.                        | CodeCatalyst usa um modelo de usuário diferente do repositório público npmjs.                                                                                 |
| <a href="#">pesquisa</a>  | Pesquisa no registro por pacotes que correspondam aos termos de pesquisa. | CodeCatalyst não suporta o <code>search</code> comando.                                                                                                       |
| <a href="#">star</a>      | Marca seus pacotes favoritos.                                             | CodeCatalyst atualmente não suporta nenhum mecanismo de favoritos.                                                                                            |
| <a href="#">stars</a>     | Exibe pacotes marcados como favoritos.                                    | CodeCatalyst atualmente não suporta nenhum mecanismo de favoritos.                                                                                            |
| <a href="#">team</a>      | Gerencia equipes e associações de equipes.                                | CodeCatalyst usa um modelo de associação de usuário e grupo diferente do repositório público npmjs.                                                           |
| <a href="#">token</a>     | Gerencia seus tokens de autenticação.                                     | CodeCatalyst usa um modelo diferente para obter tokens de autenticação. Para mais informações, consulte <a href="#">Configurando o npm com CodeCatalyst</a> . |
| <a href="#">unpublish</a> | Remove um pacote do registro.                                             | CodeCatalyst não suporta a remoção de uma versão de pacote de um repositório usando o cliente npm. É possível excluir um pacote no console.                   |

| Command                | Description                  | Observações                                                                   |
|------------------------|------------------------------|-------------------------------------------------------------------------------|
| <a href="#">whoami</a> | Exibe o nome do usuário npm. | CodeCatalyst usa um modelo de usuário diferente do repositório público npmjs. |

## Tratamento de tags npm

Os registros npm são compatíveis com tags, que são aliases de string para versões de pacotes. Você pode usar tags para proporcionar um alias em vez de usar números de versão. Por exemplo, você tem um projeto com vários fluxos de desenvolvimento e usa uma tag diferente (por exemplo, `stable`, `beta`, `dev`, `canary`) para cada fluxo. Para ver mais informações, consulte [dist-tag](#) no Documento do npm.

Por padrão, o npm usa a tag `latest` para identificar a versão atual de um pacote. `npm install pkg` (sem especificador de `@version` ou `@tag`) instala a tag mais recente. Normalmente, os projetos usam a tag mais recente para versões de lançamento estáveis. Outras tags são usadas para versões instáveis ou de pré-lançamento.

### Edição de tags com o cliente npm

Os três `npm dist-tag` comandos (`add`, `rm`, `ls`) funcionam da mesma forma nos repositórios de CodeCatalyst pacotes e no registro [npm padrão](#).

### Tags npm e repositórios upstream

Quando npm solicita as tags de um pacote e as versões desse pacote também estão presentes em um repositório upstream, CodeCatalyst mescla as tags antes de devolvê-las ao cliente. Por exemplo, um repositório chamado R tem um repositório upstream chamado U. A tabela a seguir mostra as tags de um pacote chamado `web-helper` presente nos dois repositórios.

| Repositório | Nome do pacote          | Tags do pacote                                  |
|-------------|-------------------------|-------------------------------------------------|
| R           | <code>web-helper</code> | <code>latest</code> (alias para a versão 1.0.0) |
| U           | <code>web-helper</code> | <code>alpha</code> (alias para a versão 1.0.1)  |

Nesse caso, quando o cliente npm busca as tags do pacote `web-helper` no repositório R, ele recebe as tags `latest` e `alpha`. As versões para as quais as tags apontam não mudarão.

Quando a mesma tag está presente no mesmo pacote no repositório upstream e local, CodeCatalyst usa a tag que foi atualizada pela última vez. Por exemplo, suponha que as tags no webhelper tenham sido modificadas para se parecerem com as seguintes.

| Repositório | Nome do pacote | Tags do pacote                     | Última atualização    |
|-------------|----------------|------------------------------------|-----------------------|
| R           | web-helper     | latest (alias para a versão 1.0.0) | 1º de janeiro de 2023 |
| U           | web-helper     | latest (alias para a versão 1.0.1) | 1º de junho de 2023   |

Nesse caso, quando o cliente npm buscar as tags para o pacote `web-helper` do repositório R, a tag `latest` terá um alias para a versão 1.0.1, pois ela foi atualizada na última vez. Isso facilita o consumo de novas versões de pacotes em um repositório upstream que ainda não estão presentes em um repositório local por meio da execução de `npm update`.

## Usando o Maven

O formato do repositório Maven é usado por várias linguagens diferentes, incluindo Java, Kotlin, Scala e Clojure. Ele é compatível com várias ferramentas de construção diferentes, incluindo Maven, Gradle, Scala SBT, Apache Ivy e Leiningen.

Testamos e confirmamos a compatibilidade com CodeCatalyst as seguintes versões:

- Versão mais recente do Maven: 3.6.3.
- Versão mais recente do Gradle: 6.4.1. A versão 5.5.1 também foi testada.

### Tópicos

- [Configuração e uso do Gradle Groovy](#)
- [Configuração e uso de mvn](#)
- [Publicação de pacotes com curl](#)
- [Usar somas de verificação e instantâneos do Maven](#)

## Configuração e uso do Gradle Groovy

Para usar o Gradle Groovy com CodeCatalyst, você deve conectar o Gradle Groovy ao seu repositório de pacotes e fornecer um token de acesso pessoal (PAT) para autenticação. Você pode ver as instruções para conectar o Gradle Groovy ao seu repositório de pacotes no console. CodeCatalyst

### Sumário

- [Buscando dependências de CodeCatalyst](#)
- [Buscando plug-ins de CodeCatalyst](#)
- [Buscando pacotes de repositórios de pacotes externos por meio de CodeCatalyst](#)
- [Publicando pacotes em CodeCatalyst](#)
- [Execução de uma compilação do Gradle no IntelliJ IDEA](#)
  - [Método 1: colocar o PAT em gradle.properties](#)
  - [Método 2: colocar o PAT em um arquivo separado](#)

### Buscando dependências de CodeCatalyst

As instruções a seguir explicam como configurar o Gradle Groovy para buscar dependências no seu repositório de pacotes. CodeCatalyst

Para usar o Gradle Groovy para buscar dependências do seu repositório de pacotes CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto.
3. No painel de navegação, selecione Packages (pacotes).
4. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
5. Selecione Conectar ao repositório.
6. Na caixa de diálogo Conectar ao repositório, selecione Gradle Groovy na lista de clientes do gerenciador de pacotes.
7. Você precisará de um token de acesso pessoal (PAT) para autenticar o Gradle Groovy. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.

 Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

8. Atualize o arquivo de propriedades do gradle com suas credenciais de acesso.

*username* Substitua pelo seu CodeCatalyst nome de usuário e *PAT* substitua pelo seu token de acesso CodeCatalyst pessoal. Você pode usar qualquer valor durante *spaceUsername* e *spacePassword* desde que use os mesmos valores nas etapas a seguir.

```
spaceUsername=username
spacePassword=PAT
```

9. Para buscar dependências CodeCatalyst em uma versão do Gradle, copie o trecho de maven código e adicione-o à *repositories* seção no arquivo do seu projeto. *build.gradle* Substitua os valores a seguir. Você pode usar qualquer valor *spaceName*, desde que use os mesmos valores nas etapas a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
maven {
 name = 'spaceName'
 url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
 credentials(PasswordCredentials)
}
```

10. (Opcional) Para usar o repositório de CodeCatalyst pacotes como a única fonte para as dependências do seu projeto, remova todas as outras seções nos repositórios do arquivo.



`build.gradle` Se você tiver mais de um repositório, o Gradle pesquisará dependências em cada repositório, na ordem em que estão listadas.

## Buscando plug-ins de CodeCatalyst

Por padrão, o Gradle resolverá plug-ins do [Portal de plug-ins do Gradle](#) público. As etapas a seguir configuram seu projeto Gradle para resolver plug-ins do seu repositório de CodeCatalyst pacotes.

Para usar o Gradle para buscar plug-ins do seu repositório de pacotes CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto.
3. No painel de navegação, selecione Packages (pacotes).
4. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
5. Selecione Conectar ao repositório.
6. Na caixa de diálogo Conectar ao repositório, selecione Gradle na lista de clientes do gerenciador de pacotes.
7. Você precisará de um token de acesso pessoal (PAT) para autenticar o Gradle. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.

### Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

8. Atualize o arquivo de propriedades do gradle com suas credenciais de acesso.  
*username* Substitua pelo seu CodeCatalyst nome de usuário e *PAT* substitua pelo seu token de acesso CodeCatalyst pessoal. Você pode usar qualquer valor durante *spaceUsername* e *spacePassword* desde que use os mesmos valores nas etapas a seguir.

```
spaceUsername=username
spacePassword=PAT
```

9. Adicione um bloco `pluginManagement` ao seu arquivo `settings.gradle`. O bloco `pluginManagement` deve aparecer antes de qualquer outra declaração em `settings.gradle`. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- `spaceName` Substitua pelo valor do nome usado na etapa anterior.
- `space_name` Substitua pelo nome do seu CodeCatalyst espaço.
- `proj_name` Substitua pelo nome CodeCatalyst do seu projeto.
- `repo_name` Substitua pelo nome CodeCatalyst do repositório do pacote.

```
pluginManagement {
 repositories {
 maven {
 name = 'spaceName'
 url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
 credentials(PasswordCredentials)
 }
 }
}
```

Isso garantirá que o Gradle resolva plug-ins do repositório especificado. O repositório deve ter uma conexão upstream configurada com o Portal de plug-ins do Gradle (por exemplo, `gradle-plugins-store`) para que os plug-ins do Gradle normalmente exigidos estejam disponíveis para a compilação. Para obter mais informações, consulte a [documentação do Gradle](#).

## Buscando pacotes de repositórios de pacotes externos por meio de CodeCatalyst

Você pode instalar pacotes Maven de repositórios públicos por meio de um CodeCatalyst repositório configurando-o com uma conexão upstream com o gateway que representa o repositório do gateway. Os pacotes instalados do repositório do gateway são ingeridos e armazenados no seu CodeCatalyst repositório.

CodeCatalyst suporta os seguintes repositórios públicos de pacotes Maven.

- maven-central-gateway
- google-android-gateway
- gradle-plugins-gateway
- commonsware-gateway

Para instalar pacotes de repositórios de pacotes públicos do Maven

1. Se você ainda não o fez, configure o Gradle com seu repositório de CodeCatalyst pacotes seguindo as etapas em [Buscando dependências de CodeCatalyst](#) ou. [Buscando plug-ins de CodeCatalyst](#)
2. Certifique-se de que seu repositório tenha adicionado o repositório de gateway a partir do qual você deseja instalar como uma conexão upstream. Você pode fazer isso seguindo as instruções em [Adição de um repositório upstream](#) e escolhendo o repositório público de pacotes que deseja adicionar como upstream.

Para ter mais informações sobre como solicitar pacotes de repositórios upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Publicando pacotes em CodeCatalyst

Esta seção descreve como publicar uma biblioteca Java criada com o Gradle Groovy em um repositório. CodeCatalyst

Para usar o Gradle Groovy para publicar pacotes em um repositório de pacotes CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.
3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.
5. Na caixa de diálogo Conectar ao repositório, selecione Gradle Groovy na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para autenticar o Gradle. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.

- a. Selecione Criar token.
- b. Para copiar o PAT, selecione Copiar.

 Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Atualize o arquivo de propriedades do gradle com suas credenciais de acesso.  
*username* Substitua pelo seu CodeCatalyst nome de usuário e *PAT* substitua pelo seu token de acesso CodeCatalyst pessoal. Você pode usar qualquer valor durante *spaceUsername* e *spacePassword* desde que use os mesmos valores nas etapas a seguir.

```
spaceUsername=username
spacePassword=PAT
```

8. Adicione o plug-in maven-publish à seção plugins do arquivo build.gradle do projeto.

```
plugins {
 id 'java-library'
 id 'maven-publish'
}
```

9. Em seguida, adicione uma seção publishing ao arquivo build.gradle do projeto. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
publishing {
```

```
publications {
 mavenJava(MavenPublication) {
 groupId = 'group-id'
 artifactId = 'artifact-id'
 version = 'version'
 from components.java
 }
}
repositories {
 maven {
 name = 'spaceName'
 url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
 credentials(PasswordCredentials)
 }
}
}
```

O plug-in maven-publish gera um arquivo POM com base no groupId, artifactId e version especificados na seção publishing.

- Depois que essas alterações em build.gradle forem concluídas, execute o comando a seguir para criar o projeto e carregá-lo no repositório.

```
./gradlew publish
```

- Navegue até o repositório de pacotes no CodeCatalyst console para verificar se o pacote foi publicado com sucesso. Você deve ver o pacote na lista Pacotes do seu repositório de pacotes.

Para obter mais informações, consulte os tópicos a seguir no site do Gradle:

- [Criar bibliotecas Java](#)
- [Publicar um projeto como um módulo](#)

## Execução de uma compilação do Gradle no IntelliJ IDEA

Você pode executar uma compilação do Gradle no IntelliJ IDEA que extrai dependências de CodeCatalyst. Para autenticar o Gradle com CodeCatalyst, você deve usar um token de acesso pessoal (PAT). Você pode armazenar seu CodeCatalyst PAT em gradle.properties um arquivo separado de sua escolha.

## Método 1: colocar o PAT em **gradle.properties**

Use esse método se você não estiver usando o arquivo `gradle.properties` e puder substituir o conteúdo pelo PAT. Se você estiver usando `gradle.properties`, poderá modificar esse método para adicionar o PAT em vez de substituir o conteúdo do arquivo.

### Note

O exemplo mostra o arquivo `gradle.properties` localizado em `GRADLE_USER_HOME`.

Primeiro, crie um PAT se você não tiver um.

Para criar um token de acesso pessoal (PAT)

1. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações.

### Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

2. Em Nome do PAT, insira um nome descritivo para o seu PAT.
3. Em Data de expiração, mantenha a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de vencimento é padronizada para um ano a contar da data atual.
4. Escolha Criar.

Você também pode criar esse token ao escolher Clonar repositório para um repositório de origem.

5. Salve o segredo do PAT em um local seguro.

### Important

O segredo do PAT só é exibido uma vez. Não é possível recuperá-lo depois de fechar a janela.

Depois, atualize o arquivo `build.gradle` com o seguinte trecho:

```
repositories {
 maven {
 name = 'spaceName'
 url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
 credentials(PasswordCredentials)
 }
}
```

## Método 2: colocar o PAT em um arquivo separado

Use esse método se você não quiser modificar o arquivo `gradle.properties`.

Primeiro, crie um PAT se você não tiver um.

Para criar um token de acesso pessoal (PAT)

1. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações.

### Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

2. Em Nome do PAT, insira um nome descritivo para o seu PAT.
3. Em Data de expiração, mantenha a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de vencimento é padronizada para um ano a contar da data atual.
4. Escolha Criar.

Você também pode criar esse token ao escolher Clonar repositório para um repositório de origem.

5. Salve o segredo do PAT em um local seguro.

### Important

O segredo do PAT só é exibido uma vez. Não é possível recuperá-lo depois de fechar a janela.

## Para colocar seu PAT em um arquivo separado

1. Atualize o arquivo `build.gradle` com o seguinte trecho. Substitua `space_nameproj_name`, e `repo_name` por seu nome de CodeCatalyst usuário, nome do espaço, nome do projeto e nome do repositório do pacote.

```
def props = new Properties()
file("fileName").withInputStream { props.load(it) }

repositories {
 maven {
 name = 'spaceName'
 url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
 credentials(PasswordCredentials)
 }
}
```

2. Grave o PAT no arquivo que foi especificado em seu arquivo `build.gradle`:

```
echo "codecatalystArtifactsToken=PAT" > fileName
```

## Configuração e uso de mvn

É possível usar o comando `mvn` para executar compilações do Maven. Você deve configurar `mvn` para usar seu repositório de pacotes e fornecer um token de acesso pessoal (PAT) para autenticação.

### Sumário

- [Buscando dependências de CodeCatalyst](#)
- [Buscando pacotes de repositórios de pacotes externos por meio de CodeCatalyst](#)
- [Publicando pacotes em CodeCatalyst](#)
- [Publicação de pacotes de terceiros](#)



## Buscando dependências de CodeCatalyst

Para configurar `mvn` para buscar dependências de um CodeCatalyst repositório, você deve editar o arquivo de configuração do Maven `settings.xml` e, opcionalmente, o arquivo Project Model Object (POM) do seu projeto. O arquivo POM contém informações sobre o projeto e informações de configuração do Maven para criar o projeto, como dependências, diretório de criação, diretório de origem, diretório de origem de teste, plug-in e metas.

Para usar `mvn` para buscar dependências do seu CodeCatalyst repositório de pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.
3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.
5. Na caixa de diálogo Conectar ao repositório, selecione `mvn` na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para se autenticar. `mvn` CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.



### Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Adicione um perfil contendo seu repositório ao arquivo `settings.xml`. Substitua os valores a seguir.



### Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- `space_name` Substitua pelo nome do seu CodeCatalyst espaço.

- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
<profiles>
 <profile>
 <id>repo_name</id>
 <activation>
 <activeByDefault>true</activeByDefault>
 </activation>
 <repositories>
 <repository>
 <id>repo_name</id>
 <url>https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name</url>
 </repository>
 </repositories>
 </profile>
</profiles>
```

8. Adicione seu servidor à lista de servidores no arquivo `settings.xml`. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.
- *username* Substitua CodeCatalyst pelo seu nome de usuário.
- *PAT* Substitua por seu CodeCatalyst PAT.

```
<servers>
 <server>
 <id>repo_name</id>
 <username>username</username>
 <password>PAT</password>
 </server>
```

```
</servers>
```

9. (Opcional) Defina um espelho no arquivo `settings.xml` que capture todas as conexões e as encaminhe para seu repositório em vez de um repositório de gateway.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
<mirrors>
 <mirror>
 <id>repo_name</id>
 <name>repo_name</name>
 <url>https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/</url>
 <mirrorOf>*</mirrorOf>
 </mirror>
</mirrors>
```

 Important

Você pode usar qualquer valor no elemento `<id>`, mas ele deve ser o mesmo nos elementos `<server>` e `<repository>`. Isso permite que as credenciais especificadas sejam incluídas nas solicitações para CodeCatalyst.

Depois de fazer essas alterações na configuração, você pode criar o projeto.

```
mvn compile
```

## Buscando pacotes de repositórios de pacotes externos por meio de CodeCatalyst

Você pode instalar pacotes Maven de repositórios públicos por meio de um CodeCatalyst repositório configurando-o com uma conexão upstream com o gateway que representa o repositório do gateway. Os pacotes instalados do repositório do gateway são ingeridos e armazenados no seu CodeCatalyst repositório.

Atualmente, CodeCatalyst oferece suporte aos seguintes repositórios públicos de pacotes Maven.

- maven-central-gateway
- google-android-gateway
- gradle-plugins-gateway
- commonsware-gateway

Para instalar pacotes de repositórios de pacotes públicos do Maven

1. Se você ainda não o fez, configure mvn com seu repositório de CodeCatalyst pacotes seguindo as etapas em [Buscando dependências de CodeCatalyst](#).
2. Certifique-se de que seu repositório tenha adicionado o repositório de gateway a partir do qual você deseja instalar como uma conexão upstream. Para verificar quais fontes upstream foram adicionadas ou para adicionar um repositório de gateway como fonte upstream, siga as instruções em [Adição de um repositório upstream](#).

Para ter mais informações sobre como solicitar pacotes de repositórios upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Publicando pacotes em CodeCatalyst

Para publicar um pacote Maven em um CodeCatalyst repositório, você também deve editar `~/ .m2/ settings.xml` o POM do projeto. mvn

Para usar **mvn** para publicar pacotes em seu repositório de CodeCatalyst pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.
3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.

5. Na caixa de diálogo Conectar ao repositório, selecione mvn na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para se autenticar. mvn CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.

 Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Configure uma variável de ambiente na máquina local com seu PAT. Você usará essa variável de ambiente no arquivo `setting.xml`.

```
export CODECATALYST_ARTIFACTS_TOKEN=your_PAT
```

8. Adicione uma seção `<servers>` a `settings.xml` com uma referência à variável de ambiente `CodeCatalyst_ARTIFACTS_TOKEN` para que o Maven passe o token nas solicitações HTTP.

```
<settings>
...
 <servers>
 <server>
 <id>repo-name</id>
 <username>username</username>
 <password>${env.CodeCatalyst_ARTIFACTS_TOKEN}</password>
 </server>
 </servers>
...
</settings>
```

9. Adicione uma seção `<distributionManagement>` ao `pom.xml` do seu projeto.

```
<project>
...
 <distributionManagement>
 <repository>
 <id>repo_name</id>
```

```
<name>repo_name</name>
<url>https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name</url>
</repository>
</distributionManagement>
...
</project>
```

Depois de fazer essas alterações na configuração, você pode criar o projeto e publicá-lo no repositório específico.

```
mvn deploy
```

Você pode navegar até o repositório de pacotes no CodeCatalyst console para verificar se o pacote foi publicado com sucesso.

## Publicação de pacotes de terceiros

Você pode publicar pacotes Maven de terceiros em um CodeCatalyst repositório com `mvn deploy:deploy-file`. Isso pode ser útil para usuários que desejam publicar pacotes e têm somente arquivos JAR e não têm acesso ao código-fonte do pacote ou aos arquivos POM.

O comando `mvn deploy:deploy-file` gerará um arquivo POM com base nas informações passadas na linha de comando.

Primeiro, crie um PAT se você não tiver um.

Para criar um token de acesso pessoal (PAT)

1. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações.

### Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

2. Em Nome do PAT, insira um nome descritivo para o seu PAT.
3. Em Data de expiração, mantenha a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de vencimento é padronizada para um ano a contar da data atual.

#### 4. Escolha Criar.

Você também pode criar esse token ao escolher Clonar repositório para um repositório de origem.

#### 5. Salve o segredo do PAT em um local seguro.

##### Important

O segredo do PAT só é exibido uma vez. Não é possível recuperá-lo depois de fechar a janela.

### Publicação de pacotes do Maven de terceiros

#### 1. Crie um arquivo `~/.m2/settings.xml` com o seguinte conteúdo:

```
<settings>
 <servers>
 <server>
 <id>repo_name</id>
 <username>username</username>
 <password>PAT</password>
 </server>
 </servers>
</settings>
```

#### 2. Execute o comando `mvn deploy:deploy-file`:

```
mvn deploy:deploy-file -DgroupId=commons-cli \
-DartifactId=commons-cli \
-Dversion=1.4 \
-Dfile=./commons-cli-1.4.jar \
-Dpackaging=jar \
-DrepositoryId=repo_name \
-Durl=https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/
```

 Note

O exemplo anterior publica `commons-cli 1.4`. Modifique os argumentos `groupId`, `artifactId`, `version` e `file` para publicar um JAR diferente.

Essas instruções são baseadas em exemplos no [Guia para implantação de terceiros JARs em repositórios remotos](#) da documentação do Apache Maven.

Para ter mais informações, consulte esses tópicos no site do Apache Maven Project:

- [Configurar vários repositórios](#)
- [Referência de configurações](#)
- [Gerenciamento de distribuição](#)
- [Perfis](#)

## Publicação de pacotes com curl

Esta seção mostra como usar o cliente HTTP `curl` para publicar pacotes Maven em um CodeCatalyst repositório de pacotes. A publicação de pacotes com `curl` pode ser útil se você não tiver ou quiser instalar o cliente Maven em seus ambientes.

Para publicar um pacote do Maven com **curl**

1. Você deve armazenar um token de acesso pessoal (PAT) em uma variável de ambiente com a qual se autenticar. `curl` CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um e configurar a variável de ambiente.
  - a. Crie um PAT seguindo as etapas em [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#). Copie o PAT para armazená-lo em uma variável de ambiente.
  - b. Na linha de comando da máquina local, configure uma variável de ambiente com o seu PAT.

```
export CodeCatalyst_ARTIFACTS_TOKEN=your_PAT
```



2. Use o `curl` comando a seguir para publicar o JAR em um CodeCatalyst repositório. Substitua *username*, *proj\_name*, e *repo\_name* por seu nome de CodeCatalyst usuário, nome do espaço, nome do projeto e nome do repositório do pacote.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/1.0/my-app-1.0.jar \
 --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \
 --data-binary @target/path/to/my-app-1.0.jar
```

3. Use o `curl` comando a seguir para publicar o POM em um CodeCatalyst repositório. Substitua *username*, *proj\_name*, e *repo\_name* por seu nome de CodeCatalyst usuário, nome do espaço, nome do projeto e nome do repositório do pacote.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/1.0/my-app-1.0.pom \
 --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \
 --data-binary @target/my-app-1.0.pom
```

4. Neste momento, o pacote Maven estará em seu CodeCatalyst repositório com o status de. Unfinished Para poder consumir o pacote, ele deve estar no estado Published. Você pode mover o pacote de Unfinished para Published carregando um `maven-metadata.xml` arquivo em seu pacote ou alterando o status no CodeCatalyst console.
  - a. Opção 1: use o comando `curl` a seguir para adicionar um arquivo `maven-metadata.xml` ao seu pacote. Substitua *username*, *proj\_name*, e *repo\_name* por seu nome de CodeCatalyst usuário, nome do espaço, nome do projeto e nome do repositório do pacote.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/maven-metadata.xml \
 --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \
 --data-binary @target/maven-metadata.xml
```

Um exemplo dos conteúdos em um arquivo `maven-metadata.xml` é mostrado abaixo:

```
<metadata modelVersion="1.1.0">
 <groupId>com.mycompany.app</groupId>
```

```
<artifactId>my-app</artifactId>
<versioning>
 <latest>1.0</latest>
 <release>1.0</release>
 <versions>
 <version>1.0</version>
 </versions>
 <lastUpdated>20200731090423</lastUpdated>
</versioning>
</metadata>
```

- b. Opção 2: atualize o status do pacote para Published no CodeCatalyst console. Para ter informações sobre como atualizar o status de uma versão do pacote, consulte [Atualização do status da versão do pacote](#).

Se você tiver apenas o arquivo JAR de um pacote, poderá publicar uma versão de pacote consumível em um CodeCatalyst repositório usando `mvn`. Isso pode ser útil se você não tiver acesso ao código-fonte do pacote ou ao POM. Para mais detalhes, consulte [Publicação de pacotes de terceiros](#).

## Usar somas de verificação e instantâneos do Maven

As seções a seguir descrevem como usar somas de verificação do Maven e instantâneos do Maven em CodeCatalyst.

### Usar somas de verificação do Maven

Quando um pacote Maven é publicado em um repositório de CodeCatalyst pacotes, a soma de verificação associada a cada ativo ou arquivo no pacote é usada para validar o upload. Os arquivos `jar`, `pom` e `war` são exemplos de ativos. Para cada ativo, o pacote do Maven contém vários arquivos de soma de verificação que usam o nome do ativo com uma extensão adicional, como `md5` ou `sha1`. Por exemplo, os arquivos de soma de verificação de um arquivo chamado `my-maven-package.jar` podem ser `my-maven-package.jar.md5` e `my-maven-package.jar.sha1`.

Cada pacote do Maven também contém um arquivo `maven-metadata.xml`. Esse arquivo deve ser carregado para que uma publicação seja bem-sucedida. Se uma incompatibilidade de soma de verificação for detectada durante o upload de qualquer arquivo de pacote, a publicação será interrompida. Isso pode impedir que `maven-metadata.xml` seja carregado. Quando isso acontece, o status do pacote do Maven é definido como `Unfinished`. Você não pode baixar ativos que fazem parte de um pacote com esse status.

Lembre-se do seguinte no caso de uma incompatibilidade de soma de verificação ao publicar um pacote do Maven:

- Se a incompatibilidade da soma de verificação ocorrer antes do upload de `maven-metadata.xml`, o status do pacote não será definido como `Unfinished`. O pacote não está visível e os ativos não podem ser consumidos. Quando isso acontecer, tente uma das opções a seguir e tente baixar o ativo novamente.
- Execute o comando que publica o pacote do Maven novamente. Isso pode funcionar se um problema de rede tiver corrompido o arquivo de soma de verificação durante o download. Se o problema de rede for resolvido na nova tentativa, a soma de verificação será correspondente e o download será concluído.
- Se a republicação do pacote do Maven não funcionar, exclua o pacote e, depois, publique-o novamente.
- Se a incompatibilidade da soma de verificação ocorrer depois do upload de `maven-metadata.xml`, o status do pacote será definido como `Published`. Você pode consumir qualquer ativo do pacote, incluindo aqueles com incompatibilidades de soma de verificação. Quando você baixa um ativo, a soma de verificação gerada por CodeCatalyst é baixada com ele. Se o arquivo baixado estiver associado a uma incompatibilidade de soma de verificação, o arquivo de soma de verificação baixado pode não corresponder à soma de verificação que foi carregada quando o pacote foi publicado.

## Uso de snapshots do Maven

Um snapshot do Maven é uma versão especial de um pacote do Maven que se refere ao código de ramificação de produção mais recente. É uma versão de desenvolvimento que precede a versão final de lançamento. Você pode identificar a versão de snapshot de um pacote do Maven pelo sufixo `SNAPSHOT` anexado à versão do pacote. Por exemplo, o snapshot da versão 1.1 é 1.1-SNAPSHOT. Para obter mais informações, consulte [O que é uma versão de SNAPSHOT?](#) no site do Apache Maven Project.

CodeCatalyst suporta a publicação e o consumo de instantâneos do Maven. Você pode publicar um snapshot do Maven em um CodeCatalyst repositório ou, se estiver diretamente conectado, em um repositório upstream. No entanto, não há suporte para uma versão de snapshot em um repositório de pacotes e em um dos repositórios upstream. Por exemplo, se você fizer upload de um pacote Maven com versão 1.2-SNAPSHOT para seu repositório de pacotes, CodeCatalyst não oferece suporte ao upload de um pacote Maven com a mesma versão de instantâneo em um de seus repositórios upstream. Esse cenário pode retornar resultados imprevisíveis.

Quando um snapshot do Maven é publicado, a versão anterior dele é preservada em uma nova versão chamada de compilação. Cada vez que um snapshot do Maven é publicado, uma nova versão de compilação é criada. Todas as versões anteriores de um snapshot são mantidas em suas versões de compilação. Quando um snapshot do Maven é publicado, o status é definido como `Published` e o status da compilação que contém a versão anterior é definido como `Unlisted`.

Se você solicitar um snapshot, a versão com o status `Published` será retornada. É sempre a versão mais recente de snapshot do Maven. Você também pode solicitar uma compilação específica de um snapshot.

Para excluir todas as versões de compilação de um snapshot do Maven, use o CodeCatalyst console.

## Usando NuGet

Esses tópicos descrevem como consumir e publicar NuGet pacotes usando CodeCatalyst.

### Note

CodeCatalyst suporta a [NuGet versão 4.8](#) e superior.

### Tópicos

- [Usando CodeCatalyst com o Visual Studio](#)
- [Configuração e uso da CLI nuget ou dotnet](#)
- [NuGet normalização do nome do pacote, versão e nome do ativo](#)
- [NuGet compatibilidade](#)

## Usando CodeCatalyst com o Visual Studio

Você pode consumir pacotes CodeCatalyst diretamente no Visual Studio.

Para configurar e usar NuGet com ferramentas de CLI, como `dotnet` ou `nuget`, consulte [Configuração e uso da CLI nuget ou dotnet](#)

### Sumário

- [Configurando o Visual Studio com CodeCatalyst](#)
  - [Windows](#)
  - [macOS](#)

## Configurando o Visual Studio com CodeCatalyst

### Windows

Para configurar o Visual Studio com CodeCatalyst

1. É necessário um token de acesso pessoal (PAT) para se autenticar com. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, siga as instruções em [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#) para criar um.
2. Use nuget ou dotnet para configurar seu repositório de pacotes e suas credenciais.

#### dotnet

Usuários de Linux e macOS: como a criptografia não é compatível com plataformas que não sejam Windows, você deve adicionar o sinalizador `--store-password-in-clear-text` ao comando a seguir. Observe que isso armazenará sua senha como texto simples no arquivo de configuração.

```
dotnet nuget add source https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/v3/index.json --name repo_name --password PAT --username user_name
```

#### nuget

```
nuget sources add -name repo_name -Source https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/v3/index.json -password PAT --username user_name
```

Resultado do exemplo:

```
Package source with Name: repo_name added successfully.
```

3. Configure o Visual Studio para usar sua nova origem de pacote. No Visual Studio, selecione Ferramentas e Opções.

4. No menu Opções, expanda a seção NuGet Package Manager e escolha Package Sources.
5. Na lista Fontes de pacotes disponíveis, verifique se sua *repo\_name* fonte está ativada. Se você configurou seu repositório de pacotes com uma conexão upstream com a NuGet Galeria, desative a fonte nuget.org.

## macOS

### Para configurar o Visual Studio com CodeCatalyst

1. É necessário um token de acesso pessoal (PAT) para se autenticar com. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, siga as instruções em [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#) para criar um.
2. Selecione Preferências na barra de menu.
3. Na NuGet seção, escolha Fontes.
4. Selecione Adicionar e adicione as informações do seu repositório.
  - a. Em Nome, insira o nome CodeCatalyst do repositório do pacote.
  - b. Em Localização, insira o endpoint CodeCatalyst do repositório de pacotes. O snippet de código a seguir mostra um exemplo de endpoint. Substitua *space-name* *proj-name*, e *repo-name* pelo nome do seu CodeCatalyst espaço, nome do projeto e nome do repositório.

```
https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/
```
  - c. Em Nome de usuário, insira qualquer valor válido.
  - d. Em Senha, insira o PAT.
5. Escolha Add source (Adicionar origem).
6. Se você configurou seu repositório de pacotes com uma conexão upstream com a NuGet Galeria, desative a fonte nuget.org.

Após a configuração, o Visual Studio pode consumir pacotes do seu CodeCatalyst repositório, de qualquer um de seus repositórios upstream ou do [NuGetdomínio.org](#) se você o tiver configurado como uma fonte upstream. Para obter mais informações sobre como navegar e instalar NuGet pacotes no Visual Studio, consulte [Instalar e gerenciar pacotes no Visual Studio usando o NuGet Package Manager](#) na NuGet documentação.

## Configuração e uso da CLI nuget ou dotnet

Você pode usar ferramentas de CLI, como NuGet e, dotnet para publicar e consumir pacotes do. CodeCatalyst Este documento fornece informações sobre como configurar as ferramentas de CLI e usá-las para publicar ou consumir pacotes.

### Sumário

- [Configurando com NuGet CodeCatalyst](#)
- [Consumindo NuGet pacotes de um CodeCatalyst repositório](#)
- [Consumindo NuGet pacotes de NuGet .org por meio de CodeCatalyst](#)
- [Publicando NuGet pacotes em CodeCatalyst](#)

### Configurando com NuGet CodeCatalyst

Para configurar NuGet com CodeCatalyst, adicione um endpoint do repositório e um token de acesso pessoal ao seu arquivo de NuGet configuração para permitir nuget ou conectar-se dotnet ao seu repositório de CodeCatalyst pacotes.

Para configurar NuGet com seu repositório de CodeCatalyst pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.
3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.
5. Na caixa de diálogo Conectar ao repositório, escolha NuGet ou dotnet na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para se autenticar. NuGet CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.



#### Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Configure nuget ou use dotnet o NuGet endpoint e CodeCatalyst o PAT do seu repositório. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *username* Substitua CodeCatalyst pelo seu nome de usuário.
- *PAT* Substitua por seu CodeCatalyst PAT.
- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

- a. Para nuget, use o comando `nuget sources add`.

```
nuget sources add -name "repo_name" -Source "https://
packages.region.codecatalyst.aws/nuget/space_name/proj_name/repo_name/v3/
index.json" -username "username" -password "PAT"
```

- b. Para dotnet, use o comando `dotnet nuget add source`.

Usuários de Linux e macOS: como a criptografia não é compatível com plataformas que não sejam Windows, você deve adicionar o sinalizador `--store-password-in-clear-text` ao comando a seguir. Observe que isso armazenará sua senha como texto simples no arquivo de configuração.

```
dotnet nuget add source "https://packages.region.codecatalyst.aws/
nuget/space_name/proj_name/repo_name/v3/index.json" -n "proj_name/repo_name" -u
"username" -p "PAT" --store-password-in-clear-text
```

Depois de configurar NuGet com CodeCatalyst, você pode [consumir NuGet pacotes](#) armazenados em seu CodeCatalyst repositório ou em um de seus repositórios upstream e [publicar NuGet pacotes](#) em seu repositório. CodeCatalyst



## Consumindo NuGet pacotes de um CodeCatalyst repositório

Depois de [configurar NuGet com CodeCatalyst](#), você pode consumir NuGet pacotes que estão armazenados em seu CodeCatalyst repositório ou em um de seus repositórios upstream.

Para consumir uma versão de pacote de um CodeCatalyst repositório ou de um de seus repositórios upstream com nuget ou dotnet, execute o comando a seguir. *packageName* Substitua pelo nome do pacote que você deseja consumir e *packageSourceName* pelo nome da fonte do seu repositório de CodeCatalyst pacotes no seu arquivo de NuGet configuração, que deve ser o nome do repositório.

Para instalar um pacote com **dotnet**

```
dotnet add packageName --source packageSourceName
```

Para instalar um pacote com **nuget**

```
nuget install packageName --source packageSourceName
```

Para ter mais informações, consulte [Gerenciar pacotes usando a CLI nuget](#) ou [Instalar e gerenciar pacotes usando a CLI dotnet](#) na documentação da Microsoft.

## Consumindo NuGet pacotes de NuGet .org por meio de CodeCatalyst

Você pode consumir NuGet pacotes de [NuGet.org](#) por meio de um CodeCatalyst repositório configurando o repositório com uma conexão upstream com .org. NuGet Os pacotes consumidos em NuGet.org são ingeridos e armazenados no seu CodeCatalyst repositório.

Para consumir pacotes do NuGet domínio.org

1. Se você ainda não o fez, configure seu gerenciador de NuGet pacotes com seu repositório de CodeCatalyst pacotes seguindo as etapas em [Configurando com NuGet CodeCatalyst](#).
2. Certifique-se de que seu repositório tenha adicionado NuGet.org como uma conexão upstream. Você pode verificar quais fontes upstream foram adicionadas ou adicionar o NuGet.org como uma fonte upstream seguindo as instruções [Adição de um repositório upstream](#) e escolhendo o repositório da loja. NuGet

## Publicando NuGet pacotes em CodeCatalyst

Depois de [configurar NuGet com CodeCatalyst](#), você pode usar nuget ou dotnet publicar versões de pacotes em CodeCatalyst repositórios.

Para enviar uma versão do pacote para um CodeCatalyst repositório, execute o comando a seguir com o caminho completo do seu .nupkg arquivo e o nome da fonte do seu CodeCatalyst repositório no seu arquivo de NuGet configuração.

Para publicar um pacote com **dotnet**

```
dotnet nuget push path/to/nupkg/SamplePackage.1.0.0.nupkg --source packageSourceName
```

Para publicar um pacote com **nuget**

```
nuget push path/to/nupkg/SamplePackage.1.0.0.nupkg --source packageSourceName
```

## NuGet normalização do nome do pacote, versão e nome do ativo

CodeCatalyst normaliza nomes de pacotes e ativos e versões de pacotes antes de armazená-los, o que significa que os nomes ou versões CodeCatalyst podem ser diferentes daqueles fornecidos quando o pacote ou ativo foi publicado.

Normalização do nome do pacote: CodeCatalyst normaliza os nomes dos NuGet pacotes convertendo todas as letras em minúsculas.

Normalização da versão do pacote: CodeCatalyst normaliza as versões NuGet do pacote usando o mesmo padrão do. NuGet As informações a seguir são dos [números de versão normalizados](#) da NuGet documentação.

- Os zeros iniciais são removidos dos números de versão:
  - 1.00 é tratado como 1.0
  - 1.01.1 é tratado como 1.1.1
  - 1.00.0.1 é tratado como 1.0.0.1
- Um zero na quarta parte do número da versão será omitido:
  - 1.0.0.0 é tratado como 1.0.0
  - 1.0.01.0 é tratado como 1.0.1

- SemVer Os metadados da compilação 2.0.0 foram removidos:
  - 1.0.7+r3456 é tratado como 1.0.7

Normalização do nome do ativo do pacote: CodeCatalyst constrói o nome do ativo do NuGet pacote a partir do nome do pacote normalizado e da versão do pacote.

## NuGet compatibilidade

Este guia contém informações sobre CodeCatalyst a compatibilidade com diferentes NuGet ferramentas e versões.

### Tópicos

- [NuGet Compatibilidade geral](#)
- [NuGet suporte de linha de comando](#)

### NuGet Compatibilidade geral

CodeCatalyst suporta NuGet 4.8 e superior.

CodeCatalyst suporta apenas a V3 do protocolo NuGet HTTP. Isso significa que alguns comandos da CLI que dependem da V2 do protocolo não são compatíveis. Consulte a seção [suporte ao comando nuget](#) a seguir para ter mais informações.

CodeCatalyst não suporta PowerShellGet 2.x.

### NuGet suporte de linha de comando

CodeCatalyst suporta as ferramentas de CLI NuGet (nuget) e .NET Core (dotnet).

#### suporte ao comando nuget

Como CodeCatalyst só é compatível com o protocolo HTTP V3 NuGet do, os comandos a seguir não funcionarão quando usados em CodeCatalyst recursos:

- `list`: o comando `nuget list` exibe uma lista de pacotes de uma determinada fonte. Para obter uma lista de pacotes em um repositório de CodeCatalyst pacotes, navegue até o repositório no CodeCatalyst console.

# Usar o Python

Esses tópicos descrevem como usar `pip` o gerenciador de pacotes Python e `twine` o utilitário de publicação de pacotes Python com CodeCatalyst

## Tópicos

- [Configurar `pip` e instalação de pacotes Python](#)
- [Configuração do `Twine` e publicação de pacotes Python](#)
- [Normalização do nome do pacote Python](#)
- [Compatibilidade com Python](#)

## Configurar `pip` e instalação de pacotes Python

Para usar `pip` com CodeCatalyst, você deve se conectar `pip` ao seu repositório de pacotes e fornecer um token de acesso pessoal para autenticação. Você pode ver as instruções para se conectar `pip` ao seu repositório de pacotes no CodeCatalyst console. Depois de se autenticar e se `pip` conectar CodeCatalyst, você pode executar `pip` comandos.

## Sumário

- [Instalando pacotes Python a partir do CodeCatalyst `pip`](#)
- [Consumindo pacotes Python do PyPI por meio de CodeCatalyst](#)
- [Suporte para comandos `pip`](#)
  - [Comandos compatíveis que interagem com um repositório](#)
  - [Comandos do lado do cliente compatíveis](#)

## Instalando pacotes Python a partir do CodeCatalyst `pip`

As instruções a seguir explicam como configurar `pip` para instalar pacotes Python a partir do seu CodeCatalyst repositório de pacotes ou de um de seus repositórios upstream.

Para configurar e usar **`pip`** para instalar pacotes Python a partir do seu CodeCatalyst repositório de pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.

3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.
5. Na caixa de diálogo Conectar ao repositório, selecione pip na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para autenticar o pip. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.

 Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Use o `pip config` comando para definir o URL e as credenciais do CodeCatalyst registro. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- *username* Substitua CodeCatalyst pelo seu nome de usuário.
- *PAT* Substitua por seu CodeCatalyst PAT.
- *space\_name* Substitua pelo nome do seu CodeCatalyst espaço.
- *proj\_name* Substitua pelo nome CodeCatalyst do seu projeto.
- *repo\_name* Substitua pelo nome CodeCatalyst do repositório do pacote.

```
pip config set global.index-url https://username:PAT@https://
packages.region.codecatalyst.aws/pypi/space_name/proj_name/repo_name/simple/
```

8. Supondo que um pacote esteja presente em seu repositório ou em um de seus repositórios upstream, você pode instalá-lo com `pip install`. Por exemplo, use o seguinte comando para instalar o pacote `requests`.

```
pip install requests
```

Use a `-i` opção de reverter temporariamente a instalação de pacotes do <https://pypi.org> em vez do seu repositório de CodeCatalyst pacotes.

```
pip install -i https://pypi.org/simple requests
```

## Consumindo pacotes Python do PyPI por meio de CodeCatalyst

Você pode consumir pacotes Python do Python [Package Index \(PyPI\)](#) por meio de um repositório configurando o CodeCatalyst repositório com uma conexão upstream com o PyPI. Os pacotes consumidos do PyPI são ingeridos e armazenados em seu repositório. CodeCatalyst

Para consumir pacotes do PyPI

1. Se você ainda não o fez, configure o pip com seu repositório de CodeCatalyst pacotes seguindo as etapas em [Instalando pacotes Python a partir do CodeCatalyst pip](#)
2. Certifique-se de que seu repositório tenha adicionado PyPI como uma origem upstream. Você pode verificar quais fontes upstream foram adicionadas ou adicionar o PyPI como uma fonte upstream seguindo as instruções em [Adição de um repositório upstream](#) e escolhendo o repositório do armazenamento PyPI.

Para ter mais informações sobre como solicitar pacotes de repositórios upstream, consulte [Solicitar uma versão do pacote com repositórios upstream](#).

## Suporte para comandos pip

As seções a seguir resumem os comandos pip que são suportados pelos CodeCatalyst repositórios, além dos comandos específicos que não são suportados.

### Tópicos

- [Comandos compatíveis que interagem com um repositório](#)
- [Comandos do lado do cliente compatíveis](#)

## Comandos compatíveis que interagem com um repositório

Esta seção lista os comandos `pip` em que o cliente `pip` faz uma ou mais solicitações ao registro com o qual foi configurado. Verificou-se que esses comandos funcionam corretamente quando invocados em um repositório de CodeCatalyst pacotes.

| Command                              | Description       |
|--------------------------------------|-------------------|
| <a href="#"><code>install</code></a> | Instalar pacotes. |
| <a href="#"><code>baixar</code></a>  | Baixar pacotes.   |

CodeCatalyst não implementa `pip search`. Se você configurou `pip` com um repositório de CodeCatalyst pacotes, a execução `pip search` pesquisará e mostrará pacotes do [PyPI](#).

## Comandos do lado do cliente compatíveis

Esses comandos não exigem nenhuma interação direta com um repositório, portanto, CodeCatalyst não é necessário fazer nada para suportá-los.

| Command                                | Description                                                      |
|----------------------------------------|------------------------------------------------------------------|
| <a href="#"><code>uninstall</code></a> | Desinstalar pacotes.                                             |
| <a href="#"><code>freeze</code></a>    | Emita pacotes instalados no formato de requisitos.               |
| <a href="#"><code>list</code></a>      | Listar pacotes instalados.                                       |
| <a href="#"><code>show</code></a>      | Mostre informações sobre os pacotes instalados.                  |
| <a href="#"><code>check</code></a>     | Verifique se os pacotes instalados têm dependências compatíveis. |
| <a href="#"><code>config</code></a>    | Gerencie a configuração local e global.                          |
| <a href="#"><code>wheel</code></a>     | Construa rodas conforme suas necessidades.                       |
| <a href="#"><code>hash</code></a>      | Calcule hashes de arquivos de pacotes.                           |

| Command                    | Description                              |
|----------------------------|------------------------------------------|
| <a href="#">completion</a> | Ajuda na conclusão do comando.           |
| <a href="#">debug</a>      | Mostre informações úteis para depuração. |
| help                       | Mostre ajuda para comandos.              |

## Configuração do Twine e publicação de pacotes Python

Para usar twine com CodeCatalyst, você deve se conectar twine ao seu repositório de pacotes e fornecer um token de acesso pessoal para autenticação. Você pode ver as instruções para se conectar twine ao seu repositório de pacotes no CodeCatalyst console. Depois de se autenticar e se twine conectar CodeCatalyst, você pode executar twine comandos.

### Publicando pacotes CodeCatalyst com o Twine

As instruções a seguir explicam como se autenticar e se conectar twine ao seu repositório de CodeCatalyst pacotes.

Para configurar e usar **twine** para publicar pacotes em seu repositório de CodeCatalyst pacotes

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na página de visão geral do projeto, selecione Pacotes.
3. Escolha seu repositório de pacotes na lista de repositórios de pacotes.
4. Selecione Conectar ao repositório.
5. Na caixa de diálogo Conectar ao repositório, selecione Twine na lista de clientes do gerenciador de pacotes.
6. Você precisará de um token de acesso pessoal (PAT) para autenticar o Twine. CodeCatalyst Se você já tiver um, pode usá-lo. Caso contrário, você pode criar um aqui.
  - a. Selecione Criar token.
  - b. Para copiar o PAT, selecione Copiar.



 Warning

Você não poderá ver ou copiar seu PAT novamente depois de fechar a caixa de diálogo.

7. Você pode configurar o twine com um arquivo `.pypirc` ou com variáveis de ambiente.

a. Para configurar com um arquivo `.pypirc`.

Abra `~/.pypirc` com seu editor preferido.

Adicione um servidor de indexação para CodeCatalyst, incluindo o repositório, o nome de usuário e o PAT que você criou e copiou em uma etapa anterior. Substitua os valores a seguir.

 Note

Se estiver copiando as instruções do console, os valores a seguir devem ser atualizados para você e não devem ser alterados.

- `username` Substitua CodeCatalyst pelo seu nome de usuário.
- `PAT` Substitua por seu CodeCatalyst PAT.
- `space_name` Substitua pelo nome do seu CodeCatalyst espaço.
- `proj_name` Substitua pelo nome CodeCatalyst do seu projeto.
- `repo_name` Substitua pelo nome CodeCatalyst do repositório do pacote.

```
[distutils]
index-servers = proj-name/repo-name

[proj-name/repo-name]
repository = https://packages.region.codecatalyst.aws/
pypi/space_name/proj_name/repo_name/
password = PAT
username = username
```

b. Para configurar com variáveis de ambiente.

Defina as seguintes variáveis de ambiente. No `TWINE_REPOSITORY_URL` valor `space_nameproj_name`, atualize `repo_name` com seus nomes de repositório de CodeCatalyst espaço, projeto e pacote.

```
export TWINE_USERNAME=username
```

```
export TWINE_PASSWORD=PAT
```

```
export TWINE_REPOSITORY_URL="https://packages.region.codecatalyst.aws/
pypi/space_name/proj_name/repo_name/"
```

8. Publique uma distribuição em Python com o comando `twine upload`.

## Normalização do nome do pacote Python

CodeCatalyst normaliza os nomes dos pacotes antes de armazená-los, o que significa que os nomes dos pacotes CodeCatalyst podem ser diferentes do nome fornecido quando o pacote foi publicado.

Ao realizar a normalização dos pacotes Python, o nome do pacote é colocado em minúsculas e todas as instâncias dos caracteres `.`, `-` e `_` são substituídas por um único caractere `-`. Portanto, os nomes dos pacotes `pigeon_cli` e `pigeon.cli` são normalizados e armazenados como `pigeon-cli`. O nome não normalizado pode ser usado por `pip` e `twine`. Para obter mais informações sobre a normalização de nome de pacote Python, consulte [PEP 503](#) na documentação do Python.

## Compatibilidade com Python

Embora CodeCatalyst não seja compatível com a `/simple/` API, ele suporta as operações Legacy da API. CodeCatalyst não suporta operações de PyPI XML-RPC ou JSON API.

Para obter mais informações, consulte o seguinte no repositório da Python Packaging Authority. GitHub

- [API legada](#)
- [API XML-RPC](#)
- [API JSON](#)

## Cotas para pacotes

A tabela a seguir descreve cotas e limites para pacotes na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

| Recurso                                      | Cota padrão                                                                     |
|----------------------------------------------|---------------------------------------------------------------------------------|
| Repositórios de pacotes                      | Máximo de 1000 por espaço.                                                      |
| Repositórios upstream diretos                | Máximo de 10 por repositório de pacotes.                                        |
| Repositórios upstream de pacotes pesquisados | Máximo de 25 repositórios upstream pesquisados por versão de pacote solicitada. |
| Tamanho do arquivo do ativo do pacote        | Máximo de 5 GB por ativo do pacote.                                             |
| Ativos do pacote                             | Máximo de 150 por versão do pacote.                                             |

# Compilação, teste e implantação com fluxos de trabalho

Depois de escrever o código do aplicativo em um [ambiente de CodeCatalyst desenvolvimento](#) e enviá-lo para o [repositório de CodeCatalyst origem](#), você está pronto para implantá-lo. A maneira de fazer isso automaticamente é por meio de um fluxo de trabalho.

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

## Tip

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

## Sobre o arquivo de definição do fluxo de trabalho

Arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve o fluxo de trabalho. Por padrão, o arquivo é armazenado em uma pasta `~/.codecatalyst/workflows/` na raiz do [repositório de origem](#). O arquivo pode ter uma extensão `.yaml` ou `.yml`, e a extensão deve estar em minúsculas.

Veja a seguir um exemplo de um arquivo de definição de fluxo de trabalho simples. Explicamos cada linha desse exemplo na tabela a seguir.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
```

```

Build:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: docker build -t MyApp:latest .

```

| Linha                               | Description                                                                                                                                                                                                                                                                               |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name: MyWorkflow                    | Especifica o nome do fluxo de trabalho. Para ter mais informações sobre a propriedade do Name, consulte <a href="#">Propriedades de nível superior</a> .                                                                                                                                  |
| SchemaVersion: 1.0                  | Especifica a versão do esquema de fluxo de trabalho. Para ter mais informações sobre a propriedade do SchemaVersion , consulte <a href="#">Propriedades de nível superior</a> .                                                                                                           |
| RunMode: QUEUED                     | Indica como CodeCatalyst lida com várias execuções. Para ter mais informações sobre o modo de execução, consulte <a href="#">Configurar o comportamento de enfileiramento das execuções</a> .                                                                                             |
| Triggers:                           | Especifica a lógica que fará com que a execução de um fluxo de trabalho seja iniciada. Para ter mais informações sobre gatilhos, consulte <a href="#">Início da execução automática de um fluxo de trabalho usando gatilhos</a> .                                                         |
| - Type: PUSH<br>Branches:<br>- main | Indica que o fluxo de trabalho deve ser iniciado sempre que você envia o código para a ramificação main do repositório de origem padrão. Para ter mais informações sobre a origem do fluxo de trabalho, consulte <a href="#">Conectar repositórios de origem aos fluxos de trabalho</a> . |

| Linha                                                              | Description                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>Actions:</div>                                                | Define as tarefas a serem realizadas durante a execução de um fluxo de trabalho. Neste exemplo, a seção Actions define uma única ação chamada Build. Para ter mais informações sobre as ações, consulte <a href="#">Configuração de ações de fluxo de trabalho</a> . |
| <div>Build:</div>                                                  | Define as propriedades da ação Build. Para ter mais informações sobre a ação de criação, consulte <a href="#">Criação com fluxos de trabalho</a> .                                                                                                                   |
| <div>Identifier: aws/builddev1</div>                               | Especifica o identificador exclusivo e com codificação rígida para a ação de criação.                                                                                                                                                                                |
| <div>Inputs:</div> <div>Sources:</div> <div>- WorkflowSource</div> | Indica que a ação de criação deve procurar no repositório de origem WorkflowSource os arquivos necessários para concluir o processamento. Para obter mais informações, consulte <a href="#">Conectar repositórios de origem aos fluxos de trabalho</a> .             |
| <div>Configuration:</div>                                          | Contém as propriedades de configuração específicas para a ação de criação.                                                                                                                                                                                           |
| <div>Steps:</div> <div>- Run: docker build -t MyApp:latest .</div> | Diz à ação de criação que crie uma imagem do Docker chamada MyApp e a marque com latest.                                                                                                                                                                             |

Para ver uma lista completa de todas as propriedades disponíveis no arquivo de definição de fluxo de trabalho, consulte [Definição do YAML do fluxo de trabalho](#).

## Usando os editores visual e YAML do CodeCatalyst console

Para criar e editar o arquivo de definição do fluxo de trabalho, você pode usar seu editor preferido, mas recomendamos usar o editor visual ou o editor YAML do CodeCatalyst console. Esses editores

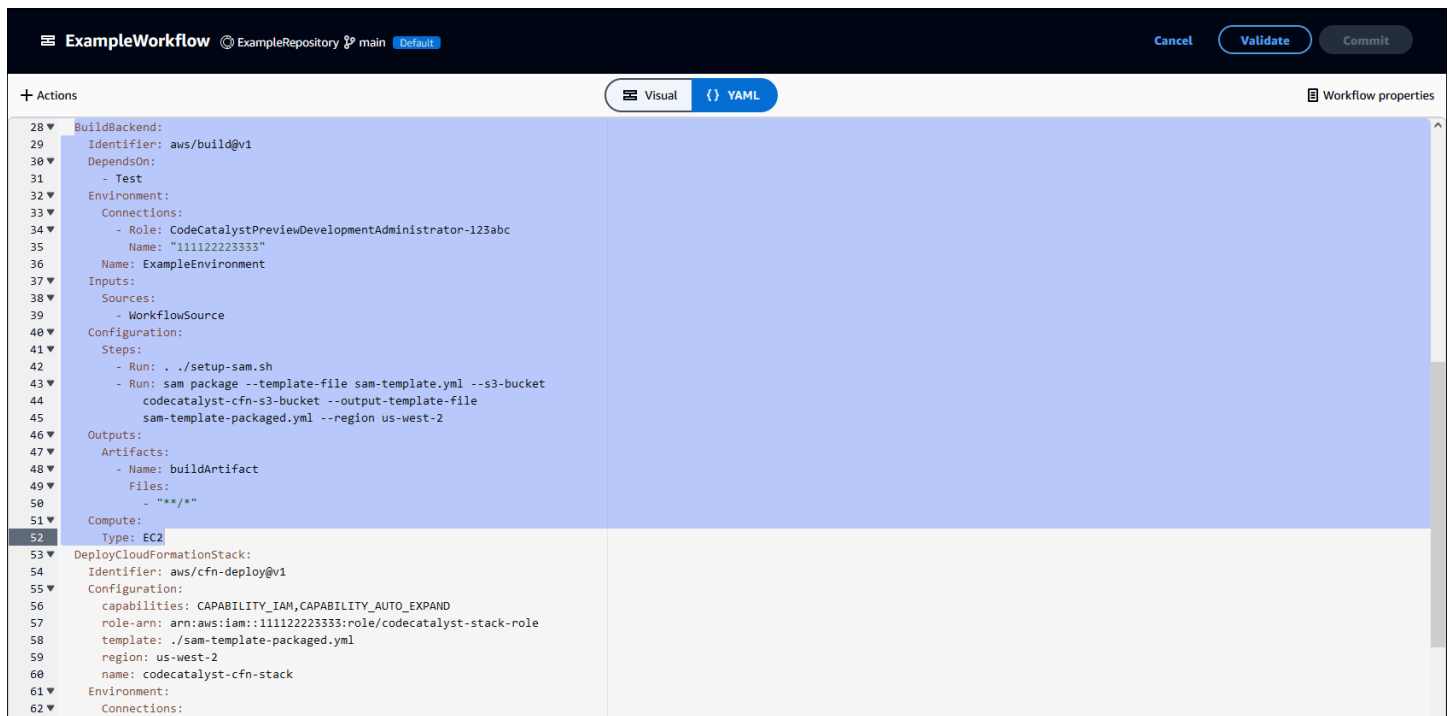
oferecem validação útil de arquivos para ajudar a garantir que nomes, valores, aninhamento, espaçamento, uso de letras maiúsculas e minúsculas, etc. das propriedades YAML estejam corretos.

A imagem a seguir mostra um fluxo de trabalho no editor visual. O editor visual oferece uma interface de usuário completa para criar e configurar seu arquivo de definição de fluxo de trabalho. O editor visual inclui um diagrama de fluxo de trabalho (1) mostrando os principais componentes do fluxo de trabalho e uma área de configuração (2).

The screenshot displays the Amazon CodeCatalyst console interface. At the top, there's a header with 'ExampleWorkflow', 'ExampleRepository', 'main', and 'Default' tabs, along with 'Cancel', 'Validate', and 'Commit' buttons. Below the header, there's a 'Visual' tab and a 'YAML' tab. The main area is divided into two sections:

- Workflow diagram (1):** A visual representation of the workflow. It starts with a 'Source' action (WorkflowSource) triggered by 'Push'. This is followed by a 'Test' action (aws/managed-test@v1), then a 'BuildBackend' action (aws/build@v1) configured with 'Environment: ExampleEnvironment' and 'Production'. Finally, there's a 'DeployCloudFormationStack' action (aws/cfn-deploy@v1) also configured with 'Environment: ExampleEnvironment' and 'Production'.
- Configuration area (2):** A panel on the right showing the configuration for the 'BuildBackend' action. It includes tabs for 'Inputs', 'Configuration', and 'Outputs'. The 'Configuration' tab is active, showing fields for 'Action name' (BuildBackend), 'Compute type' (EC2), 'Compute fleet - optional' (Choose compute), 'Environment/account/role - optional' (ExampleEnvironment), and 'AWS account connection' (111122223333).

Como alternativa, é possível usar o editor YAML, mostrado na imagem a seguir. Use o editor YAML para colar blocos de código grandes (de um tutorial, por exemplo) ou para adicionar propriedades avançadas que não são oferecidas pelo editor visual.



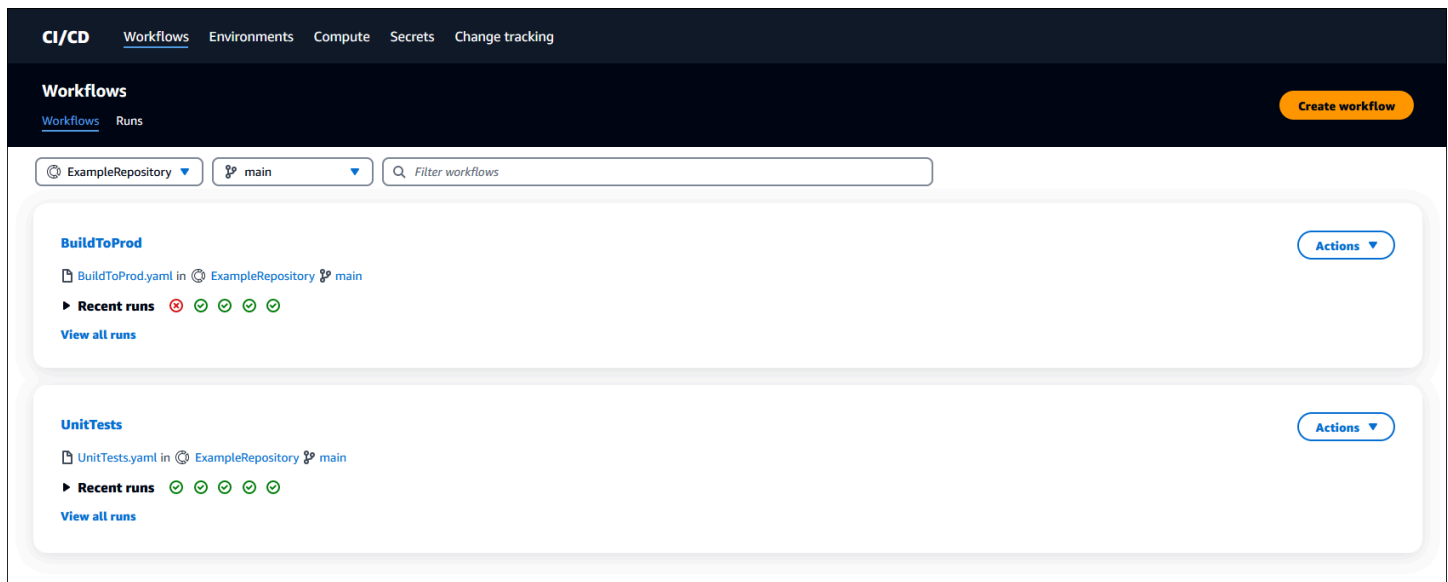
Você pode alternar do editor visual para o editor YAML para ver o efeito que suas configurações têm no código YAML subjacente.

## Descoberta de fluxos de trabalho

Você pode visualizar seu fluxo de trabalho na página de resumo dos Fluxos de trabalho, junto com outros fluxos de trabalho configurados no mesmo projeto.

A seguinte imagem mostra a página de resumo dos Fluxos de trabalho. Ele é preenchido com dois fluxos de trabalho: e. BuildToProdUnitTests Você pode ver que ambos foram executados algumas vezes. Selecione Execuções recentes para ver rapidamente o histórico de execuções ou selecione o nome do fluxo de trabalho para ver o código YAML do fluxo de trabalho e outras informações detalhadas.





## Visualização de detalhes da execução do fluxo de trabalho

Visualize os detalhes da execução de um fluxo de trabalho selecionando a execução na página de resumo dos Fluxos de trabalho.

A imagem a seguir mostra os detalhes de uma execução de fluxo de trabalho chamada Run-cc11d, iniciada automaticamente em uma confirmação na origem. O diagrama do fluxo de trabalho indica que uma ação falhou (1). Você pode navegar até os logs (2) para visualizar as mensagens de log detalhadas e solucionar problemas. Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

The screenshot displays the Amazon CodeCatalyst console for a workflow named 'Run-cc11d'. The workflow is in a 'Failed' state, indicated by a red circle with a white 'X' in the top left corner. The 'Status' section shows 'Failed' with a red circle and 'X' icon. The 'Run mode' is 'Queued', 'Trigger' is 'Started by 0f8bf654', 'Start time' is '11 minutes ago', 'Duration' is '3 minutes 17 seconds', and 'End time' is '8 minutes ago'. The workflow graph shows four actions: 'WorkflowSource' (green checkmark), 'Test' (green checkmark), 'BuildBackend' (red circle with 'X'), and 'DeployCloudFormationStack' (green checkmark). The 'BuildBackend' action is highlighted with a red circle and '1' indicating it failed. A 'Detailed log messages' panel is open on the right, showing the error message: 'Error: Template file not found at /codecatalyst/output/src3862/src/git-codecommit.us'. The log also shows the command 'sam package --template-file sam-template.yml --s3-bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml --region us-west-2'.

## Próximas etapas

Para saber mais sobre conceitos de fluxos de trabalho, consulte [Conceitos de fluxos de trabalho](#).

Para criar seu primeiro fluxo de trabalho, consulte [Conceitos básicos de fluxos de trabalho](#).

## Conceitos de fluxos de trabalho

Aqui estão alguns conceitos e termos que você deve conhecer ao criar, testar ou implantar seu código com fluxos de trabalho. CodeCatalyst

## Fluxos de trabalho

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

 Tip

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

## Arquivos de definição de fluxo de trabalho

Arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve o fluxo de trabalho. Por padrão, o arquivo é armazenado em uma pasta `~/.codecatalyst/workflows/` na raiz do [repositório de origem](#). O arquivo pode ter uma extensão `.yaml` ou `.yml`, e a extensão deve estar em minúsculas.

Para ter mais informações sobre o arquivo de definição de fluxo de trabalho, consulte [Definição do YAML do fluxo de trabalho](#).

## Ações

Uma ação é o principal componente de um fluxo de trabalho e define uma unidade lógica de trabalho, ou tarefa, a ser realizada durante a execução de um fluxo de trabalho. Normalmente, um fluxo de trabalho inclui várias ações que são executadas de modo sequencial ou paralelo, dependendo de como você as configurou.

Para ter mais informações sobre as ações, consulte [Configuração de ações de fluxo de trabalho](#).

## Grupos de ações

Um grupo de ações contém uma ou mais ações. O agrupamento de ações em grupos ajuda a manter o fluxo de trabalho organizado e também permite configurar dependências entre grupos diferentes.

Para ter mais informações sobre os grupos de ações, consulte [Agrupar ações em grupos de ações](#).

## Artefatos

Artefato é a saída de uma ação de fluxo de trabalho e geralmente consiste em uma pasta ou um arquivamento de arquivos. Os artefatos são importantes porque permitem que você compartilhe arquivos e informações entre as ações.

Para obter mais informações sobre artefatos, consulte [Compartilhar artefatos e arquivos entre ações](#).

## Computação

Computação se refere ao mecanismo de computação (CPU, memória e sistema operacional) gerenciado e mantido por CodeCatalyst para executar ações de fluxo de trabalho.

Para ter mais informações sobre computação, consulte [Configuração de imagens de computação e runtime](#).

## Ambientes do

Um CodeCatalyst ambiente, que não deve ser confundido com um [ambiente de desenvolvimento](#), define o Amazon VPC de destino Conta da AWS e opcional ao qual um CodeCatalyst [fluxo de trabalho](#) se conecta. Um ambiente também define a [função do IAM](#) que um fluxo de trabalho precisa para acessar os AWS serviços e recursos na conta de destino.

É possível configurar vários ambientes e atribuir a eles nomes, como desenvolvimento, teste, preparação e produção. Quando você implanta nesses ambientes, as informações sobre as implantações aparecem nas CodeCatalyst guias Atividade de implantação e Destinos de implantação no ambiente.

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#).

## Portões

Portão é um componente do fluxo de trabalho que pode ser usado para impedir que a execução do fluxo de trabalho continue, a menos que determinadas condições sejam atendidas. Um exemplo de porta é a porta de aprovação, na qual os usuários devem enviar uma aprovação no CodeCatalyst console antes que a execução do fluxo de trabalho possa continuar.

É possível adicionar portões entre sequências de ações a um fluxo de trabalho ou antes da primeira ação (executada imediatamente após o download da Origem). Também será possível adicionar portões após a última ação, se necessário.

Para ter mais informações sobre portões, consulte [Bloquear uma execução de fluxo de trabalho](#).

## Relatórios

Um relatório contém detalhes sobre os testes que ocorrem durante a execução de um fluxo de trabalho. É possível criar relatórios, por exemplo, de teste, de cobertura de código, de análise de composição de software e de análise estática. É possível usar um relatório para ajudar a solucionar

um problema durante um fluxo de trabalho. Se você tiver muitos relatórios de vários fluxos de trabalho, poderá usá-los para visualizar tendências e taxas de falha para ajudar a otimizar as aplicações e as configurações de implantação.

Para obter mais informações sobre os relatórios, consulte [Tipos de relatório de qualidade](#).

## Execuções

Execução é uma iteração única de um fluxo de trabalho. Durante uma execução, CodeCatalyst executa as ações definidas no arquivo de configuração do fluxo de trabalho e gera os registros, artefatos e variáveis associados.

Para ter mais informações sobre execuções, consulte [Executar um fluxo de trabalho](#).

## Fontes

Origem, também chamada de origem de entrada, é um repositório de origem ao qual uma [ação de fluxo de trabalho](#) se conecta para receber os arquivos necessários com o objetivo de realizar as operações. Por exemplo, uma ação de fluxo de trabalho pode se conectar a um repositório de origem para receber arquivos de origem a aplicação com o objetivo de criar uma aplicação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

## Variáveis

Uma variável é um par de valores-chave que contém informações que você pode consultar em seu fluxo de trabalho da Amazon CodeCatalyst. A parte do valor da variável é substituída por um valor real quando o fluxo de trabalho é executado.

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

## Gatilhos do fluxo de trabalho

Um gatilho de fluxo de trabalho, ou simplesmente um gatilho, permite que você inicie a execução automática de um fluxo de trabalho quando determinados eventos ocorrerem, como um envio de código. Talvez você queira configurar gatilhos para liberar seus desenvolvedores de software da necessidade de iniciar execuções de fluxo de trabalho manualmente por meio do CodeCatalyst console.

É possível usar três tipos de gatilho:

- **Envio:** um gatilho de envio de código faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma confirmação é enviada.
- **Solicitação pull:** um gatilho de solicitação pull faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma solicitação pull é criada, revisada ou fechada.
- **Programação:** um gatilho de programação faz com que a execução de um fluxo de trabalho comece em uma programação definida por você. Pense em usar um gatilho de programação para executar compilações noturnas do software para que a versão mais recente esteja pronta para os desenvolvedores de software trabalharem na manhã seguinte.

É possível usar gatilhos de envio, solicitação pull e programação de maneira independente ou combinados no mesmo fluxo de trabalho.

Os gatilhos são opcionais. Se você não configurar nenhum, só poderá iniciar um fluxo de trabalho manualmente.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

## Conceitos básicos de fluxos de trabalho

Neste tutorial, você aprenderá a criar e configurar o primeiro fluxo de trabalho.

### Tip

Prefere começar com um fluxo de trabalho pré-configurado? Consulte [Criar um projeto com um esquema](#), que inclui instruções para configurar um projeto com um fluxo de trabalho funcional, uma aplicação de exemplo e outros recursos.

### Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar e configurar o fluxo de trabalho](#)
- [Etapa 2: salvar o fluxo de trabalho com uma confirmação](#)
- [Etapa 3: visualizar os resultados da execução](#)
- [\(Opcional\) Etapa 4: limpar](#)

# Pré-requisitos

## Antes de começar

- Você precisa de um CodeCatalyst espaço. Para obter mais informações, consulte [Criar um espaço](#).
- Em seu CodeCatalyst espaço, você precisa de um projeto vazio chamado:

```
codecatalyst-project
```

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- No seu projeto, você precisa de um CodeCatalyst repositório chamado:

```
codecatalyst-source-repository
```

Para obter mais informações, consulte [Criar um repositório de origem](#).

### Note

Se você tiver um projeto e um repositório de origem, poderá usá-los; no entanto, a criação de outros facilita a limpeza no final deste tutorial.

## Etapa 1: criar e configurar o fluxo de trabalho

Nesta etapa, você cria e configura um fluxo de trabalho que compila e testa automaticamente o código-fonte quando as alterações são feitas.

### Como criar o fluxo de trabalho

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Selecione Criar fluxo de trabalho.

O arquivo de definição do fluxo de trabalho aparece no editor YAML do CodeCatalyst console.

## Como configurar o fluxo de trabalho

Você pode configurar o fluxo de trabalho no editor Visual ou no editor YAML. Vamos começar com o editor YAML e depois mudar para o editor visual.

1. Escolha + Ações para ver uma lista de ações do fluxo de trabalho que você pode adicionar ao fluxo de trabalho.
2. Na ação Criar, selecione + para adicionar o YAML da ação ao arquivo de definição de fluxo de trabalho. O fluxo de trabalho será semelhante ao mostrado a seguir.

```
Name: Workflow_fe47
SchemaVersion: "1.0"

Optional - Set automatic triggers.
Triggers:
 - Type: Push
 Branches:
 - main

Required - Define action configurations.
Actions:
 Build_f0:
 Identifier: aws/build@v1

 Inputs:
 Sources:
 - WorkflowSource # This specifies that the action requires this workflow as
a source

 Outputs:
 AutoDiscoverReports:
 Enabled: true
 # Use as prefix for the report files
 ReportNamePrefix: rpt

 Configuration:
 Steps:
 - Run: echo "Hello, World!"
 - Run: echo "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\" >> report.xml
report.xml
 - Run: echo "<testsuite tests=\"1\" name=\"TestAgentJunit\" >\" >>
report.xml
 - Run: echo "<testcase classname=\"TestAgentJunit\" name=\"Dummy
```



```
Test\"/></testsuite>" >> report.xml
```

O fluxo de trabalho copia os arquivos no repositório de **WorkflowSource** origem para a máquina computacional que executa a **Build\_f0** ação, imprime **Hello, World!** nos registros, descobre relatórios de teste na máquina computacional e os envia para a página Relatórios do CodeCatalyst console.

3. Selecione Visual para visualizar o arquivo de definição do fluxo de trabalho no editor visual. Os campos no editor visual permitem configurar as propriedades YAML mostradas no editor YAML.

## Etapa 2: salvar o fluxo de trabalho com uma confirmação

Nesta etapa, você salva as alterações. Como os fluxos de trabalho são armazenados como arquivos `.yaml` no repositório, você salva suas alterações com confirmações.

Como confirmar as alterações no fluxo de trabalho

1. (Opcional) Selecione Validar para garantir que o código YAML do fluxo de trabalho seja válido.
2. Selecione Confirmar.
3. Em Nome do arquivo do fluxo de trabalho, insira um nome para o arquivo de configuração do fluxo de trabalho, como **my-first-workflow**.
4. Em Mensagem de confirmação, insira uma mensagem para identificar sua confirmação, por exemplo **create my-first-workflow.yaml**.
5. Em Repositório, selecione o repositório no qual você deseja salvar o fluxo de trabalho (codecatalyst-repository).
6. Em Nome da ramificação, selecione a ramificação na qual você deseja salvar o fluxo de trabalho (main).
7. Selecione Confirmar.

Seu novo fluxo de trabalho aparece na lista de fluxos de trabalho. A exibição pode demorar alguns instantes.

Como os fluxos de trabalho são salvos com confirmações e como o fluxo de trabalho tem um gatilho de envio de código configurado, salvar o fluxo de trabalho inicia a execução do fluxo de trabalho automaticamente.

## Etapa 3: visualizar os resultados da execução

Nesta etapa, você navega até a execução que foi iniciada a partir da sua confirmação e visualiza os resultados.

Como visualizar os resultados da execução

1. Selecione o nome do fluxo de trabalho, por exemplo, `Workflow_fe47`.

Um diagrama de fluxo de trabalho mostrando o rótulo do seu repositório de origem (WorkflowSource) e a ação de criação (por exemplo, `build_F0`).

2. No diagrama de execução do fluxo de trabalho, selecione a ação de criação (por exemplo, `Build_f0`).
3. Analise o conteúdo das guias Logs, Relatórios, Configuração e Variáveis. Essas guias mostram os resultados da sua ação de criação.

Para obter mais informações, consulte [Visualização dos resultados de uma ação de criação](#).

## (Opcional) Etapa 4: limpar

Nesta etapa, você limpará os recursos que criou neste tutorial.

Como excluir recursos

- Se você criou um projeto para este tutorial, exclua-o. Para instruções, consulte [Excluir um projeto](#). A exclusão do projeto também exclui o repositório de origem e o fluxo de trabalho.

## Criação com fluxos de trabalho

Usando [CodeCatalyst fluxos de trabalho](#), você pode criar aplicativos e outros recursos.

Tópicos

- [Como faço para criar uma aplicação?](#)
- [Benefícios da ação de criação](#)
- [Alternativas à ação de criação](#)
- [Adição da ação de criação](#)
- [Visualização dos resultados de uma ação de criação](#)

- [Tutorial: Fazer upload de artefatos no Amazon S3](#)
- [Ações de criação e de teste YAML](#)

## Como faço para criar uma aplicação?

Para criar um aplicativo ou recurso em CodeCatalyst, primeiro você cria um fluxo de trabalho e, em seguida, especifica uma ação de construção dentro dele.

A ação de criação é um componente do fluxo de trabalho que compila seu código-fonte, executa testes de unidade e produz artefatos prontos para implantação.

Você adiciona uma ação de criação ao seu fluxo de trabalho usando o editor visual do CodeCatalyst console ou o editor YAML.

As etapas detalhadas para criar uma aplicação ou um recurso são as seguintes.

Como criar uma aplicação (tarefas detalhadas)

1. Em CodeCatalyst, você adiciona o código-fonte de um aplicativo que deseja criar. Para obter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#).
2. Em CodeCatalyst, você cria um fluxo de trabalho. No fluxo de trabalho, você define como criar, testar e implantar a aplicação. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#).
3. (Opcional) No fluxo de trabalho, você adiciona um gatilho que indica os eventos que farão com que o fluxo de trabalho seja iniciado automaticamente. Para obter mais informações, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
4. No fluxo de trabalho, você adiciona uma ação de criação que compila e empacota o código-fonte da aplicação ou do recurso. Se desejar, você também poderá fazer com que a ação de criação execute testes de unidade, gere relatórios e implante a aplicação se você não quiser usar uma ação de teste ou de implantação para essas finalidades. Para ter mais informações sobre as ações de teste e de implantação, consulte [Adição da ação de criação](#).
5. (Opcional) No fluxo de trabalho, você adiciona uma ação de teste e uma ação de implantação para testar e implantar a aplicação ou o recurso. Você pode escolher entre várias ações pré-configuradas para implantar a aplicação em diferentes destinos, como o Amazon ECS. Para ter mais informações, consulte [Teste com fluxos de trabalho](#) e [Implantar com fluxos de trabalho](#).

6. Você inicia o fluxo de trabalho manual ou automaticamente por meio de um gatilho. O fluxo de trabalho executa as ações de criação, teste e implantação em sequência para criar, testar e implantar a aplicação e seus recursos no destino. Para obter mais informações, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Benefícios da ação de criação

O uso da ação de criação em um fluxo de trabalho fornece os seguintes benefícios:

- Totalmente gerenciado: a ação de criação elimina a necessidade de configurar, aplicar patches e atualizações e gerenciar os próprios servidores de compilação.
- Sob demanda: a ação de criação escala sob demanda, para atender às necessidades da compilação. Você paga somente pela quantidade de minutos de compilação que consumir. Para obter mais informações, consulte [Configuração de imagens de computação e runtime](#).
- Pronto para uso — CodeCatalyst inclui imagens Docker de ambiente de tempo de execução pré-empacotadas que são usadas para executar todas as ações do seu fluxo de trabalho, incluindo ações de construção. Essas imagens vêm pré-configuradas com ferramentas úteis para criar aplicativos como o AWS CLI e o Node.js. Você pode configurar CodeCatalyst para usar uma imagem de compilação fornecida por você a partir de um registro público ou privado. Para obter mais informações, consulte [Especificação de imagens de ambiente de runtime](#).

## Alternativas à ação de criação

Se você estiver usando uma ação de criação para implantar seu aplicativo, considere usar uma ação de CodeCatalyst implantação em vez disso. As ações de implantação realizam behind-the-scenes configurações que, de outra forma, você precisaria escrever manualmente se estivesse usando uma ação de compilação. Para ter mais informações sobre as ações de implantação disponíveis, consulte [Lista de ações de implantação](#).

Você também pode usar AWS CodeBuild para criar seus aplicativos. Para obter mais informações, consulte [O que é o CodeBuild?](#).

## Adição da ação de criação

Use o procedimento a seguir para adicionar uma ação de criação ao seu CodeCatalyst fluxo de trabalho.

## Visual

Para adicionar uma ação de criação usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione Visual.
6. Escolha Ações.
7. Em Ações, escolha Criar.
8. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [Ações de criação e de teste YAML](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como adicionar uma ação de criação usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione YAML.
6. Escolha Ações.
7. Em Ações, escolha Criar.

8. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [Ações de criação e de teste YAML](#).
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Definição de ação de criação

A ação de criação é definida como um conjunto de propriedades YAML dentro do arquivo de definição do fluxo de trabalho. Para ter mais informações sobre essas propriedades, consulte [Ações de criação e de teste YAML](#) na [Definição do YAML do fluxo de trabalho](#).

## Visualização dos resultados de uma ação de criação

Use as instruções a seguir para visualizar os resultados de uma ação de criação, incluindo os logs, relatórios e variáveis gerados.

Para visualizar os resultados de uma ação de criação

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
3. No diagrama do fluxo de trabalho, escolha o nome da ação de criação, por exemplo, Compilação.
4. Para ver os logs da execução da criação, selecione Logs. Os logs das várias fases de criação são exibidos. Você pode expandir ou recolher os logs conforme necessário.
5. Para visualizar os relatórios de teste produzidos pela ação de criação, selecione Relatórios ou, no painel de navegação, selecione Relatórios. Para obter mais informações, consulte [Tipos de relatório de qualidade](#).
6. Para ver a configuração usada para a ação de criação, selecione Configuração. Para obter mais informações, consulte [Adição da ação de criação](#).
7. Para ver as variáveis usadas pela ação de criação, selecione Variáveis. Para obter mais informações, consulte [Uso de variáveis em fluxos de trabalho](#).

## Tutorial: Fazer upload de artefatos no Amazon S3

Neste tutorial, você aprende a fazer upload de artefatos em um bucket do Amazon S3 usando um fluxo de trabalho da CodeCatalyst [Amazon](#) que inclui algumas ações [de](#) criação. Essas ações são executadas em série quando o fluxo de trabalho é iniciado. A primeira ação de criação gera dois arquivos, `Hello.txt` e `Goodbye.txt`, e os agrupa em um artefato de criação. A segunda ação de criação carrega o artefato no Amazon S3. Você configurará o fluxo de trabalho para ser executado sempre que enviar uma confirmação para o repositório de origem.

### Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar uma AWS função](#)
- [Etapa 2: criar um bucket do Amazon S3](#)
- [Etapa 3: criar um repositório de origem](#)
- [Etapa 4: criar um fluxo de trabalho](#)
- [Etapa 5: verificar os resultados](#)
- [Limpeza](#)

### Pré-requisitos

Antes de começar, você precisará fazer o seguinte:

- Você precisa de um CodeCatalyst espaço com uma AWS conta conectada. Para obter mais informações, consulte [Criar um espaço](#).
- Em seu espaço, você precisa de um projeto vazio chamado:

```
codecatalyst-artifact-project
```

Use a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- Em seu projeto, você precisa de um CodeCatalyst ambiente chamado:

```
codecatalyst-artifact-environment
```

Configure esse ambiente da seguinte forma:

- Escolha qualquer tipo, como Desenvolvimento.
- Conecte sua AWS conta a ela.
- Para o Perfil do IAM padrão, escolha qualquer perfil. Você especificará um perfil diferente posteriormente.

Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).

## Etapa 1: criar uma AWS função

Nesta etapa, você cria uma função AWS do IAM que posteriormente atribuirá à ação de criação em seu fluxo de trabalho. Essa função concede à ação de CodeCatalyst construção permissão para acessar sua AWS conta e gravar no Amazon S3, onde seu artefato será armazenado. O perfil é chamado de Perfil de criação.

### Note

Se você já tiver um perfil de criação para outro tutorial, também poderá usá-lo para este tutorial. Apenas garanta que ele tenha as permissões e a política de confiança mostradas no procedimento a seguir.

Para obter mais informações sobre as funções do IAM, consulte [Funções do IAM](#) no Guia AWS AWS Identity and Access Management do usuário.

### Como criar um perfil de criação

1. Crie uma política para o perfil da seguinte maneira:
  - a. Faça login em AWS.
  - b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
  - c. No painel de navegação, selecione Políticas.
  - d. Escolha Criar política.
  - e. Escolha a guia JSON.
  - f. Exclua o código existente.
  - g. Cole o seguinte código:



## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor0",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject",
 "s3:ListBucket"
],
 "Resource": "*"
 }
]
}
```

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-s3-build-policy
```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

- 2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.

- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure `codecatalyst-s3-build-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

**`codecatalyst-s3-build-role`**

- i. Em Descrição do perfil, insira:

**`CodeCatalyst build role`**

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

## Etapa 2: criar um bucket do Amazon S3

Nesta etapa, você cria um bucket do Amazon S3 no qual os artefatos `Hello.txt` e `Goodbye.txt` serão carregados.

Como criar um bucket do Amazon S3

- 1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
- 2. No painel principal, selecione Criar bucket.
- 3. Em Nome do bucket, insira:

**`codecatalyst-artifact-bucket`**

- 4. Na região da AWS, escolha uma região. Este tutorial pressupõe que você tenha escolhido o Oeste dos EUA (Oregon) `us-west-2`. Para ter informações sobre regiões com suporte do

Amazon S3, consulte [Endpoints e cotas do Amazon Simple Storage Service](#) na Referência geral da AWS.

5. Selecione Criar bucket na parte inferior da página.
6. Copie o nome do bucket que você acabou de criar, por exemplo:

```
codecatalyst-artifact-bucket
```

Agora você criou um bucket chamado **codecatalyst-artifact-bucket** na região Oeste dos EUA (Oregon), us-west-2.

### Etapa 3: criar um repositório de origem

Nesta etapa, você cria um repositório de origem no CodeCatalyst. Esse repositório é usado para armazenar o arquivo de definição do fluxo de trabalho do tutorial.

Para ter mais informações sobre repositórios de origem, consulte [Criar um repositório de origem](#).

Como criar um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto, `codecatalyst-artifact-project`.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha Adicionar repositório e selecione Criar repositório.
5. Em Nome do repositório, insira:

```
codecatalyst-artifact-source-repository
```

6. Escolha Criar.

Agora você criou um repositório chamado `codecatalyst-artifact-source-repository`.

### Etapa 4: criar um fluxo de trabalho

Nesta etapa, você cria um fluxo de trabalho que consiste nos seguintes componentes que são executados sequencialmente:

- Um gatilho – Esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de criação chamada GenerateFiles — No gatilho, a ação GenerateFiles cria dois arquivos, Hello.txt e Goodbye.txt, os empacota em um artefato de saída chamado codecatalystArtifact.
- Outra ação de criação chamada Upload — Ao concluir a ação GenerateFiles, a ação Upload executa o comando da AWS CLI `aws s3 sync` para carregar os arquivos no codecatalystArtifact e no repositório de origem para o bucket do Amazon S3. AWS CLI Ele vem pré-instalado e pré-configurado na plataforma de CodeCatalyst computação, então você não precisa instalá-lo ou configurá-lo.

Para obter mais informações sobre o software pré-empacotado na plataforma de CodeCatalyst computação, consulte [Especificação de imagens de ambiente de runtime](#). Para obter mais informações sobre o `aws s3 sync` comando AWS CLI's, consulte [sync](#) na Referência de AWS CLI Comandos.

Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).

Como criar um fluxo de trabalho

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Selecione Criar fluxo de trabalho.
3. Exclua o código de amostra YAML.
4. Adicione o seguinte código YAML:

 Note

No código YAML a seguir, você pode omitir a seção `Connections:` se quiser. Se você omitir essa seção, deverá garantir que o perfil especificado no campo Perfil do IAM padrão em seu ambiente inclua as permissões e as políticas de confiança descritas em [Etapa 1: criar uma AWS função](#). Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#).

Name: codecatalyst-artifact-workflow

```

SchemaVersion: 1.0

Triggers:
 - Type: Push
 Branches:
 - main
Actions:
 GenerateFiles:
 Identifier: aws/build@v1
 Configuration:
 Steps:
 # Create the output files.
 - Run: echo "Hello, World!" > "Hello.txt"
 - Run: echo "Goodbye!" > "Goodbye.txt"
 Outputs:
 Artifacts:
 - Name: codecatalystArtifact
 Files:
 - "**/*"
 Upload:
 Identifier: aws/build@v1
 DependsOn:
 - GenerateFiles
 Environment:
 Name: codecatalyst-artifact-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-s3-build-role
 Inputs:
 Artifacts:
 - codecatalystArtifact
 Configuration:
 Steps:
 # Upload the output artifact to the S3 bucket.
 - Run: aws s3 sync . s3://codecatalyst-artifact-bucket

```

No código acima, substitua:

- *codecatalyst-artifact-environment* com o nome do ambiente em que você criou [Pré-requisitos](#).
- *codecatalyst-account-connection* com o nome da conexão da conta que você criou [Pré-requisitos](#).

- *codecatalyst-s3-build-role* pelo nome do perfil de criação criado em [Etapa 1: criar uma AWS função](#).
- *codecatalyst-artifact-bucket* com o nome do Amazon S3 em que você criou. [Etapa 2: criar um bucket do Amazon S3](#)

Para ter informações sobre as propriedades nesse arquivo, consulte a [Ações de criação e de teste YAML](#).

5. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
6. Selecione Confirmar.
7. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
  - a. Em Nome do arquivo do fluxo de trabalho, deixe o padrão, `codecatalyst-artifact-workflow`.
  - b. Em Confirmar mensagem, insira:

**add initial workflow file**

- c. Em Repositório, selecione `codecatalyst-artifact-source-repository`.
- d. Em Nome da ramificação, selecione principal.
- e. Selecione Confirmar.

Agora você criou um fluxo de trabalho. A execução de um fluxo de trabalho é iniciada automaticamente devido ao gatilho definido na parte superior do fluxo de trabalho. Especificamente, quando você confirmou (e enviou) o arquivo `codecatalyst-artifact-workflow.yaml` ao repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para visualizar a execução do fluxo de trabalho em andamento

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar: `codecatalyst-artifact-workflow`.
3. Escolha GenerateFilesver o progresso da primeira ação de construção.
4. Selecione Carregar para ver o progresso da segunda ação de criação.
5. Quando a ação Carregar terminar, faça o seguinte:
  - Se a execução do fluxo de trabalho for bem-sucedida, vá para o próximo procedimento.

- Se a execução do fluxo de trabalho falhar, selecione Logs para solucionar o problema.

## Etapa 5: verificar os resultados

Depois que o fluxo de trabalho for executado, acesse o serviço Amazon S3 e examine seu *codecatalyst-artifact-bucket* bucket. Agora ele deve incluir os seguintes arquivos e pastas:

```
.
|- .aws/
|- .git/
|Goodbye.txt
|Hello.txt
|README.md
```

Os arquivos `Goodbye.txt` e `Hello.txt` foram enviados porque faziam parte do artefato `codecatalystArtifact`. Os arquivos `.aws/`, `.git/` e `README.md` foram enviados porque estavam no repositório de origem.

## Limpeza

Limpe CodeCatalyst e evite AWS ser cobrado por esses serviços.

Para limpar CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Exclua o repositório de origem `codecatalyst-artifact-source-repository`.
3. Exclua o fluxo de trabalho `codecatalyst-artifact-workflow`.

Para limpar AWS

1. Limpe no Amazon S3, da seguinte forma:
  - a. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
  - b. Exclua os arquivos no bucket `codecatalyst-artifact-bucket`.
  - c. Exclua o bucket `codecatalyst-artifact-bucket`.
2. Limpe no IAM, da seguinte forma:
  - a. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

- b. Exclua o `codecatalyst-s3-build-policy`.
- c. Exclua o `codecatalyst-s3-build-role`.

## Ações de criação e de teste YAML

Veja a seguir a definição YAML das ações de criação e de teste. Há uma referência para duas ações porque as propriedades YAML de cada uma são muito semelhantes.

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

Escolha uma propriedade YAML no código a seguir para ver uma descrição dela.

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
action-name:
 Identifier: aws/build@v1 | aws/managed-test@v1
 DependsOn:
 - dependent-action-name-1
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Environment:
 Name: environment-name
 Connections:
```



- Name: *account-connection-name*
- Role: *iam-role-name*

Caching:FileCaching:key-name-1:Path: *file1.txt*RestoreKeys:

- *restore-key-1*

Inputs:Sources:

- *source-name-1*
- *source-name-2*

Artifacts:

- *artifact-name*

Variables:

- Name: *variable-name-1*  
Value: *variable-value-1*
- Name: *variable-name-2*  
Value: *variable-value-2*

Outputs:Artifacts:

- Name: *output-artifact-1*

Files:

- build-output/artifact-1.jar
- "build-output/build\*"

- Name: *output-artifact-2*

Files:

- build-output/artifact-2.1.jar
- build-output/artifact-2.2.jar

Variables:

- *variable-name-1*
- *variable-name-2*

AutoDiscoverReports:Enabled: *true | false*ReportNamePrefix: *AutoDiscovered*IncludePaths:

- *\*\*/\**

ExcludePaths:

- node\_modules/cdk/junit.xml

SuccessCriteria:PassRate: *percent*LineCoverage: *percent*BranchCoverage: *percent*Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisBug:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisSecurity:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisQuality:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisFinding:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

Reports:

report-name-1:

Format: *format*

IncludePaths:

- *"\*.xml"*

ExcludePaths:

- *report2.xml*

- *report3.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisBug:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisSecurity:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisQuality:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisFinding:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

Configuration:

Container:

Registry: *registry*

```

 Image: image
 Steps:
 - Run: "step 1"
 - Run: "step 2"
 Packages:
 NpmConfiguration:
 PackageRegistries:
 - PackageRepository: package-repository
 Scopes:
 - "@scope"
 ExportAuthorizationToken: true | false

```

## action-name

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Interface de usuário correspondente: guia Configuração/Nome da ação

## Identifier

(*action-name*/Identifier)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Use *aws/build@v1* para ações de criação.

Use *aws/managed-test@v1* para ações de teste.

UI correspondente: diagrama de fluxo de trabalho/nome da ação/ *aws/build@v1|aws/managed-test@v1* rótulo

## DependsOn

(*action-name*/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

## Compute

(*action-name*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

## Type

(*action-name*/Compute/Tipo)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

Interface de usuário correspondente: guia Configuração/Tipo de computação

## Fleet

(*action-name*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

Interface de usuário correspondente: guia Configuração/Frota de computação

## Timeout

(*action-name*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Environment

(*action-name*/Environment)

(Optional)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM

especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

#### Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

## Name

(*action-name*/Environment/Name)

(Optional)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

## Connections

(*action-name*/Environment/Connections)

(Optional)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

UI correspondente: a configuração tab/Environment/What está pronta *my-environment*? /menu de três pontos/ Mudar função

## Name

(*action-name*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

UI correspondente: a configuração tab/Environment/What está pronta *my-environment*? /menu de três pontos/ Mudar função

## Role

(*action-name*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome do perfil do IAM que essa ação usa para acessar e operar em serviços da AWS, como o Amazon S3 e o Amazon ECR. Esse perfil deve ser adicionado à conexão da Conta da AWS no espaço. Para adicionar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#).

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

### Note

É possível usar o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* com essa ação. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que

you use this profile only in tutorials and in scenarios in which security is less of a concern.

#### Warning

Limit permissions to those required by the creation and testing actions. Using a profile with more permissions can represent a security risk.

UI correspondent: the configuration tab/Environment/What is ready *my-environment*? /menu of three dots/ Change function

## Caching

(*action-name*/Caching)

(Optional)

A section in which you can specify a cache to save files on disk and restore them from this cache in subsequent workflow executions.

For more information about caching files, consult [Armazenar arquivos em cache entre execuções de fluxo de trabalho](#).

User interface correspondent: guide Configuration/Storage of files in cache - optional

## FileCaching

(*action-name*/Caching/FileCaching)

(Optional)

A section that specifies the configuration of a sequence of caches.

UI correspondent: tab/File Cache configuration - optional/Add cache

key-name-1

(*action-name*/Caching/FileCaching/*key-name-1*)



(Optional)

Especifique o nome da propriedade de cache principal. Os nomes de propriedade de cache devem ser exclusivos no fluxo de trabalho. Cada ação pode ter até cinco entradas em FileCaching.

UI correspondente: tab/File Cache de configuração - opcional/Adicionar cache/chave

## Path

(*action-name*/Caching/FileCaching/*key-name-1*/Path)

(Optional)

Especifique o caminho associado ao cache.

UI correspondente: tab/File Cache de configuração - opcional/Adicionar cache/caminho

## RestoreKeys

(*action-name*/Caching/FileCaching/*key-name-1*/RestoreKeys)

(Optional)

Especifique a chave de restauração a ser usada como fallback quando a propriedade do cache principal não puder ser encontrada. Os nomes de chave de restauração devem ser exclusivos no fluxo de trabalho. Cada cache pode ter até cinco entradas em RestoreKeys.

UI correspondente: tab/File Cache de configuração - opcional/Adicionar cache/Restaurar chaves - opcional

## Inputs

(*action-name*/Inputs)

(Optional)

A seção Inputs define os dados de que uma ação precisa durante a execução de um fluxo de trabalho.

### Note

São permitidas no máximo quatro entradas (uma origem e três artefatos) por ação de criação ou de teste. As variáveis não são contadas nesse total.

Se você precisar consultar arquivos que residem em entradas diferentes (digamos, uma origem e um artefato), a entrada de origem será a entrada primária e o artefato será a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferirem das primárias. Para obter detalhes, consulte [Exemplo: referência de arquivos em vários artefatos](#).

Interface de usuário correspondente: guia Entradas

## Sources

(*action-name*/Inputs/Sources)

(Optional)

Especifique os rótulos que representam os repositórios de origem que serão necessários para a ação. Atualmente, o único rótulo compatível é `WorkflowSource`, que representa o repositório de origem em que o arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma origem, deverá especificar pelo menos um artefato de entrada em *action-name*/Inputs/Artifacts.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: nenhuma

## Artifacts - input

(*action-name*/Inputs/Artifacts)

(Optional)

Especifique artefatos de ações anteriores que você deseja fornecer como entrada para essa ação. Esses artefatos já devem estar definidos como artefatos de saída em ações anteriores.

Se você não especificar nenhum artefato de entrada, deverá especificar pelo menos um repositório de origem em *action-name*/Inputs/Sources.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

 Note

Se a lista suspensa Artefatos – opcional não estiver disponível (editor visual) ou se você receber erros ao validar o YAML (editor YAML), talvez seja porque a ação permite apenas uma entrada. Nesse caso, tente remover a entrada de origem.

Interface de usuário correspondente: guia Entradas/Artefatos - opcional

## Variables - input

(*action-name*/Inputs/Variables)

(Optional)

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Variáveis - opcional

## Outputs

(*action-name*/Outputs)

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

## Artifacts - output

(*action-name*/Outputs/Artifacts)

(Optional)

Especifique o nome de um artefato gerado pela ação. Os nomes de artefato devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados

(`_`). Espaços, hifens (-) e outros caracteres especiais não são permitidos. Não é possível usar aspas para habilitar espaços, hifens e outros caracteres especiais em nomes de artefato de saída.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Saídas/Artefatos

## Name

(*action-name*/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique o nome de um artefato gerado pela ação. Os nomes de artefato devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (`_`). Espaços, hifens (-) e outros caracteres especiais não são permitidos. Não é possível usar aspas para habilitar espaços, hifens e outros caracteres especiais em nomes de artefato de saída.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

UI correspondente: tab/Artifacts/New Saídas, saída/nome do artefato de construção

## Files

(*action-name*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique os arquivos CodeCatalyst incluídos no artefato que é gerado pela ação. Esses arquivos são gerados pela ação do fluxo de trabalho quando ela é executada e também estão disponíveis no repositório de origem. Os caminhos de arquivo podem residir em um repositório de origem ou em um artefato de uma ação anterior e são relativos ao repositório de origem ou à raiz do artefato. É possível usar padrões glob para especificar caminhos. Exemplos:

- Para especificar um único arquivo que esteja na raiz do local de compilação ou do local do repositório de origem, use `my-file.jar`.
- Para especificar um único arquivo em um subdiretório, use `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Para especificar todos os arquivos, use `**/*`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.

- Para especificar todos os arquivos e diretórios em um diretório chamado `directory`, use `"directory/**/*"`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.
- Para especificar todos os arquivos em um diretório chamado `directory`, mas não em nenhum de seus subdiretórios, use `"directory/*"`.

#### Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outro caractere especial, coloque o caminho entre aspas duplas (""). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

#### Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou origem encontrá-lo. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

UI correspondente: `tab/Artifacts/New` saídas de saída/arquivos produzidos pela compilação

Variables - output

(*action-name*/Outputs/Variables)

(Optional)

Especifique as variáveis que você deseja que a ação exporte para que estejam disponíveis para uso em ações subsequentes.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Saídas/Variáveis/Adicionar variável

## variable-name-1

(*action-name*/Outputs/Variables/*variable-name-1*)

(Optional)

Especifique o nome de uma variável a ser exportada pela ação. Essa variável já deve estar definida na seção Inputs ou Steps da mesma ação.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

UI correspondente: tab/Variables/Add Variável de saída/nome

## AutoDiscoverReports

(*action-name*/Outputs/AutoDiscoverReports)

(Optional)

Define a configuração do recurso de descoberta automática.

Quando você ativa a descoberta automática, CodeCatalyst pesquisa todos os dados Inputs passados para a ação, bem como todos os arquivos gerados pela própria ação, procurando relatórios de teste, cobertura de código e análise de composição de software (SCA). Para cada relatório encontrado, CodeCatalyst transforma-o em um CodeCatalyst relatório. Um CodeCatalyst relatório é um relatório totalmente integrado ao CodeCatalyst serviço e que pode ser visualizado e manipulado por meio do CodeCatalyst console.

### Note

Por padrão, o recurso de descoberta automática inspeciona todos os arquivos. É possível limitar quais arquivos são inspecionados usando as propriedades [IncludePaths](#) ou [ExcludePaths](#).

Interface de usuário correspondente: guia Saídas/Relatórios/Relatórios de descoberta automática

## Enabled

(*action-name*/Outputs/AutoDiscoverReports/Enabled)

(Optional)

Habilite ou desabilite o recurso de descoberta automática.

Os valores válidos são `true` ou `false`.

Se `Enabled` for omitido, o padrão será `true`.

Interface de usuário correspondente: guia Saídas/Relatórios/Relatórios de descoberta automática

## ReportNamePrefix

(*action-name*/Outputs/AutoDiscoverReports/ReportNamePrefix)

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e habilitado)

Especifique um prefixo que seja CodeCatalyst anexado a todos os relatórios encontrados para nomear seus relatórios associados. CodeCatalyst Por exemplo, se você especificar um prefixo `deAutoDiscovered`, e CodeCatalyst descobrir automaticamente dois relatórios de teste, `TestSuiteOne.xml` e `TestSuiteTwo.xml`, os CodeCatalyst relatórios associados serão nomeados `e. AutoDiscoveredTestSuiteOne` `AutoDiscoveredTestSuiteTwo`

Interface de usuário correspondente: guia Saídas/Relatórios/Nome do prefixo

## IncludePaths

(*action-name*/Outputs/AutoDiscoverReports/IncludePaths)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/IncludePaths)

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e habilitado ou se [Reports](#) estiver incluído)

Especifique os arquivos e os caminhos de arquivo CodeCatalyst incluídos na pesquisa de relatórios brutos. Por exemplo, se você especificar `/test/report/*`, CodeCatalyst pesquisa toda a [imagem de compilação](#) usada pela ação procurando o `/test/report/*` diretório. Quando encontra esse diretório CodeCatalyst , procura relatórios nesse diretório.

### Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outros caracteres especiais, coloque o caminho entre aspas duplas ("). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Se essa propriedade for omitida, o padrão será "\*\*/\*", o que significa que a pesquisa inclui todos os arquivos em todos os caminhos.

 Note

Em relatórios configurados manualmente, IncludePaths deverá ser um padrão global que corresponda a um único arquivo.

Interface de usuário correspondente:

- Caminhos de tab/Reports/Auto-discover reports/Include/exclude saída/ Caminhos de inclusão
- Saídas tab/Reports/Manually configuram relatórios/ /incluir/excluir **report-name-1** caminhos/  
Incluir caminhos

## ExcludePaths

(**action-name**/Outputs/AutoDiscoverReports/ExcludePaths)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/ExcludePaths)

(Optional)

Especifique os arquivos e os caminhos de arquivo que são CodeCatalyst excluídos ao pesquisar relatórios brutos. Por exemplo, se você especificar "/test/my-reports/\*\*/\*", não CodeCatalyst procurará arquivos no /test/my-reports/ diretório. Para ignorar todos os arquivos em um diretório, use o padrão de glob "\*\*/\*".

 Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outros caracteres especiais, coloque o caminho entre aspas duplas (""). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Interface de usuário correspondente:

- Caminhos de tab/Reports/Auto-discover reports/Include/exclude saída/ caminhos de exclusão



- As saídas tab/Reports/Manually configuram relatórios/ /incluir/excluir **report-name-1** caminhos/  
Excluir caminhos

## SuccessCriteria

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria)

(Optional)

Especifique os critérios de sucesso para os relatórios de teste, cobertura de código, análise de composição de software (SCA) e análise estática (SA).

Para obter mais informações, consulte [Configuração de critérios de sucesso para relatórios](#).

Interface de usuário correspondente: guia Saída/Relatórios/Critérios de sucesso

## PassRate

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/PassRate)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria/PassRate)

(Optional)

Especifique a porcentagem de testes em um relatório de teste que devem ser aprovados para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para ter mais informações sobre os relatórios de teste, consulte [Relatórios de teste](#).

UI correspondente: tab/Reports/Success critérios de saída/taxa de aprovação

## LineCoverage

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/LineCoverage)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/LineCoverage)

(Optional)

Especifique a porcentagem de linhas em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente: tab/Reports/Success critérios de saída/cobertura de linha

## BranchCoverage

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/BranchCoverage)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/BranchCoverage)

(Optional)

Especifique a porcentagem de filiais em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de ramificações são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente: tab/Reports/Success critérios de saída/cobertura da filial

## Vulnerabilities

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/Vulnerabilities)

(Optional)

Especifique o número máximo e a gravidade das vulnerabilidades permitidas no relatório SCA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar vulnerabilidades, é necessário especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as vulnerabilidades HIGH e CRITICAL serão contabilizadas.

- O número máximo de vulnerabilidades da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de vulnerabilidade são aplicados somente aos relatórios de SCA. Para ter mais informações sobre os relatórios de SCA, consulte [Relatórios de análise de composição de software](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

UI correspondente: `tab/Reports/Success` critérios de saída/vulnerabilidades

## StaticAnalysisBug

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisBug)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisBug)

(Optional)

Especifique o número máximo e a gravidade dos bugs permitidos no relatório do SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar bugs, é necessário especificar:

- A gravidade mínima dos bugs que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, os bugs HIGH e CRITICAL serão contabilizados.

- O número máximo de bugs da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de bugs são aplicados somente PyLint aos relatórios do ESLint SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

UI correspondente: `tab/Reports/Success` critérios de saída/bugs

## StaticAnalysisSecurity

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisSecurity)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisSecurity)

(Optional)

Especifique o número máximo e a gravidade das vulnerabilidades de segurança permitidas no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar vulnerabilidades de segurança, é necessário especificar:

- A gravidade mínima das vulnerabilidades de segurança que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as vulnerabilidades de segurança HIGH e CRITICAL serão contabilizadas.

- O número máximo de vulnerabilidades de segurança da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de vulnerabilidades de segurança são aplicados somente aos PyLint relatórios do ESLint SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

UI correspondente: `tab/Reports/Success` critérios de saída/vulnerabilidades de segurança

## StaticAnalysisQuality

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisQuality)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisQuality)

(Optional)

Especifique o número máximo e a gravidade dos problemas de qualidade permitidos no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar problemas de qualidade, é necessário especificar:

- A gravidade mínima dos problemas de qualidade que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, os problemas de qualidade HIGH e CRITICAL serão contabilizados.

- O número máximo de problemas de qualidade da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de problemas de qualidade são aplicados somente PyLint aos relatórios ESLint da SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

UI correspondente: tab/Reports/Success critérios de saída/problemas de qualidade

## StaticAnalysisFinding

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisFinding)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisFinding)

(Optional)

Especifique o número máximo e a severidade das descobertas permitidas no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar descobertas, é necessário especificar:

- A gravidade mínima das descobertas que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as descobertas HIGH e CRITICAL serão contabilizadas.

- O número máximo de descobertas da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

As descobertas são aplicadas somente aos relatórios de SARIF SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

UI correspondente: `tab/Reports/Success` critérios de saída/descobertas

## Reports

(*action-name*/Outputs/Reports )

(Optional)

Uma seção que especifica a configuração dos relatórios de teste.

Interface de usuário correspondente: guia Saídas/Relatórios

report-name-1

(nome do *action-name* /Outputs/Reports/ relatório-1)

(Obrigatório se [Reports](#) for incluído)

O nome que você deseja dar ao CodeCatalyst relatório que será gerado a partir de seus relatórios brutos.

UI correspondente: saídas `tab/Reports/Manually` configuram relatórios/nome do relatório

## Format

(*action-name*/Outputs/Reports/*report-name-1*/Format)

(Obrigatório se [Reports](#) for incluído)

Especifique o formato de arquivo utilizado para os relatórios. Veja a seguir os valores possíveis.

- Para relatórios de teste:
  - Para Cucumber JSON, especifique Cucumber (editor visual) ou CUCUMBERJSON (editor YAML).
  - Para JUnit XML, especifique JUnit(editor visual) ou JUNITXML (editor YAML).
  - Para NUnit XML, especifique NUnit(editor visual) ou NUNITXML (editor YAML).
  - Para NUnit 3 XML, especifique NUnit3(editor visual) ou NUNIT3XML (editor YAML).
  - Para Visual Studio TRX, especifique Visual Studio TRX (editor visual) ou VISUALSTUDIOTRX (editor YAML).
  - Para TestNG XML, especifique TestNG (editor visual) ou TESTNGXML (editor YAML).
- Para relatórios de cobertura de código:
  - Para Clover XML, especifique Clover (editor visual) ou CLOVERXML (editor YAML).
  - Para Cobertura XML, especifique Cobertura (editor visual) ou COBERTURAXML (editor YAML).
  - Para JaCoCo XML, especifique JaCoCo(editor visual) ou JACOCOXML (editor YAML).
  - Para SimpleCov JSON gerado por [simplecov](#), [não simplecov-json](#), especifique Simplecov (editor visual) ou (editor YAML). SIMPLECOV
- Para relatórios de análise de composição de software (SCA):
  - Para SARIF, especifique SARIF (editor visual) ou SARIFSCA (editor YAML).

UI correspondente: tab/Reports/Manually configure reports/Add/configure relatórios de saídas/**report-name-1**/Tipo de relatório e formato do relatório

## Configuration

(**action-name**/Configuration)

(Obrigatório) Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## Container

(**action-name**/Configuration/Container)

(Optional)

Especifique a imagem do Docker, ou contêiner, que a ação usa para concluir o processamento. Você pode especificar uma das [imagens ativas](#) que vêm com CodeCatalyst ela ou usar sua própria imagem. Se você optar por usar a própria imagem, ela poderá residir no Amazon ECR, no Docker Hub ou em outro registro. Se você não especificar uma imagem do Docker, a ação usará uma das imagens ativas para processamento. Para ter informações sobre qual imagem ativa é usada por padrão, consulte [Imagens ativas](#).

Para ter mais informações sobre como especificar a própria imagem do Docker, consulte [Atribuir uma imagem do Docker de um ambiente de runtime personalizada a uma ação](#).

Interface de usuário correspondente: Imagem do Docker do runtime - opcional

## Registry

(*action-name*/Configuration/Container/Registry)

(Obrigatório se Container for incluído)

Especifique o registro em que a imagem está armazenada. Os valores válidos são:

- CODECATALYST (editor YAML)

A imagem é armazenada no CodeCatalyst registro.

- Docker Hub (editor visual) ou DockerHub (editor YAML)

A imagem é armazenada no registro de imagens do Docker Hub.

- Outro registro (editor visual) ou Other (editor YAML)

A imagem é armazenada em um registro de imagem personalizado. Qualquer registro disponível publicamente pode ser usado.

- Amazon Elastic Container Registry (editor visual) ou ECR (editor YAML)

A imagem é armazenada em um repositório de imagens do Amazon Elastic Container Registry. Para usar uma imagem em um repositório do Amazon ECR, essa ação precisa acessar o Amazon ECR. Para habilitar esse acesso, é necessário criar um [perfil do IAM](#) que inclua as permissões e a política de confiança personalizada a seguir. (Se desejar, você poderá modificar um perfil existente para incluir as permissões e a política.)

O perfil do IAM deve incluir as seguintes permissões na política:

- `ecr:BatchCheckLayerAvailability`



- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

O perfil do IAM deve incluir a seguinte política de confiança personalizada:

Para ter mais informações sobre como criar perfis do IAM, consulte [Criar um perfil usando políticas de confiança personalizadas \(console\)](#) no Guia do usuário do IAM.

Depois que você criar o perfil, deverá atribuí-lo à ação por meio de um ambiente. Para obter mais informações, consulte [Associação de um ambiente a uma ação](#).

Interface de usuário correspondente: opções Amazon Elastic Container Registry, Docker Hub e Outro registro

## Image

(*action-name*/Configuration/Container/Image)

(Obrigatório se Container for incluído)

Especifique um dos seguintes:

- Se você estiver usando um registro CODECATALYST, defina a imagem como uma das seguintes [imagens ativas](#):
  - `CodeCatalystLinux_x86_64:2024_03`
  - `CodeCatalystLinux_x86_64:2022_11`
  - `CodeCatalystLinux_Arm64:2024_03`
  - `CodeCatalystLinux_Arm64:2022_11`
  - `CodeCatalystLinuxLambda_x86_64:2024_03`
  - `CodeCatalystLinuxLambda_x86_64:2022_11`
  - `CodeCatalystLinuxLambda_Arm64:2024_03`
  - `CodeCatalystLinuxLambda_Arm64:2022_11`
  - `CodeCatalystWindows_x86_64:2022_11`
- Se você estiver usando um registro do Docker Hub, defina a imagem como o nome da imagem do Docker Hub e a tag opcional.

Exemplo: `postgres:latest`

- Se estiver usando um registro do Amazon ECR, defina a imagem como o URI do registro do Amazon ECR.

Exemplo: `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Se estiver usando um registro personalizado, defina a imagem como o valor esperado pelo registro personalizado.

Interface de usuário correspondente: imagem do Docker do ambiente de runtime (se o registro for CODECATALYST), imagem do Docker Hub (se o registro for Docker Hub), URL da imagem do ECR (se o registro for Amazon Elastic Container Registry) e URL da imagem (se o registro for Outro registro).

## Steps

(*action-name*/Configuration/Steps)

(Obrigatório)

Especifique os comandos do shell que você deseja executar durante a ação para instalar, configurar e executar as ferramentas de compilação.

Confira um exemplo de como criar um projeto npm:

Steps:

- Run: `npm install`
- Run: `npm run build`

Veja a seguir um exemplo de como especificar caminhos de arquivo:

Steps:

- Run: `cd $ACTION_BUILD_SOURCE_PATH_WorkflowSource/app && cat file2.txt`
- Run: `cd $ACTION_BUILD_SOURCE_PATH_MyBuildArtifact/build-output/ && cat file.txt`

Para ter mais informações sobre como especificar caminhos de arquivo, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

Interface de usuário correspondente: guia Configuração/Comandos de shell

## Packages

(*action-name*/Configuration/Packages)

(Optional)

Uma seção na qual você pode especificar um repositório de pacotes que a ação usa para resolver dependências. Os pacotes permitem que você armazene e compartilhe com segurança pacotes de software usados para desenvolvimento de aplicações.

Para ter mais informações sobre pacotes, consulte [Publique e compartilhe pacotes de software no CodeCatalyst](#).

Interface de usuário correspondente: guia Configuração/Pacotes

## NpmConfiguration

(*action-name*/Configuration/Packages/NpmConfiguration)

(Obrigatório se [Packages](#) for incluído)

Uma seção que define a configuração para o formato do pacote npm. Essa configuração é usada por uma ação durante a execução de um fluxo de trabalho.

Para ter mais informações sobre a configuração do pacote npm, consulte [Uso de npm](#).

UI correspondente: configuração de tab/Packages/Add configuração/npm

## PackageRegistries

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries)

(Obrigatório se [Packages](#) for incluído)

Uma seção na qual você pode definir as propriedades de configuração de uma sequência de repositórios de pacotes.

UI correspondente: tab/Packages/Add configuration/npm Configuração/ Adicionar repositório de pacotes

## PackagesRepository

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries/PackagesRepository)

(Obrigatório se [Packages](#) for incluído)

Especifique o nome do repositório de CodeCatalyst pacotes que você deseja que a ação use.

Se você especificar vários repositórios padrão, o último repositório terá prioridade.

Para ter mais informações sobre repositórios de pacote, consulte [Repositórios de pacotes](#).

UI correspondente: repositório de tab/Packages/Add configuration/npm/Add pacotes de configuração/  
repositório de pacotes

## Scopes

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries/Scopes)

(Optional)

Especifique uma sequência de escopos que você deseja definir no registro do pacote. Na definição de escopos, o repositório de pacotes especificado é configurado como o registro para todos os escopos listados. Se um pacote com o escopo for solicitado por meio do cliente npm, ele usará esse repositório em vez do padrão. Cada nome de escopo deve ter o prefixo “@”.

Se você incluir escopos substitutos, o último repositório terá prioridade.

Se Scopes for omitido, o repositório de pacotes especificado será configurado como o registro padrão para todos os pacotes usados pela ação.

Para ter mais informações sobre escopos, consulte [Namespaces de pacote](#) e [Pacotes com escopo definido](#).

UI correspondente: tab/Packages/Add configuration/npm/Add repositório/escopos do pacote de configuração - opcional

## ExportAuthorizationToken

(*action-name*/Configuration/Packages/ExportAuthorizationToken)

(Optional)

Habilite ou desabilite o recurso de token de autorização de exportação. Se habilitados, os tokens de autorização exportados podem ser usados para configurar manualmente um gerenciador de pacotes para se autenticar com repositórios de CodeCatalyst pacotes. É possível usar o token como uma variável de ambiente que pode ser referida nas ações.

Os valores válidos são `true` ou `false`.

Se `ExportAuthorizationToken` for omitido, o padrão será `false`.

Para ter mais informações sobre o token de autorização de exportação, consulte [Usar tokens de autorização em ações de fluxo de trabalho](#).

Interface de usuário correspondente: guia Configuração/Pacotes/Token de autorização de exportação

## Teste com fluxos de trabalho

Em CodeCatalyst, você pode executar testes como parte de diferentes ações do fluxo de trabalho, como criar e testar. Todas essas ações de fluxo de trabalho podem gerar relatórios de qualidade. Uma ação de teste é uma ação de fluxo de trabalho que produz relatórios de teste, cobertura de código, análise de composição de software e análise estática. Esses relatórios são exibidos no CodeCatalyst console.

### Tópicos

- [Tipos de relatório de qualidade](#)
- [Adição da ação de teste](#)
- [Visualização dos resultados de uma ação de teste](#)
- [Omissão dos testes com falha em uma ação](#)
- [Integrando com universal-test-runner](#)
- [Configuração de relatórios de qualidade em uma ação](#)
- [Melhores práticas de teste](#)
- [Propriedades SARIF compatíveis](#)

## Tipos de relatório de qualidade

A ação CodeCatalyst de teste da Amazon oferece suporte aos seguintes tipos de relatórios de qualidade. Para ver um exemplo de como formatar esses relatórios em seu YAML, consulte [Exemplo de relatórios de qualidade YAML](#).

### Tópicos

- [Relatórios de teste](#)
- [Relatórios de cobertura de código](#)
- [Relatórios de análise de composição de software](#)

- [Relatórios de análise estática](#)

## Relatórios de teste

Em CodeCatalyst, você pode configurar testes de unidade, testes de integração e testes de sistema que são executados durante as compilações. Em seguida, CodeCatalyst pode criar relatórios que contêm os resultados dos seus testes.

Você pode usar um relatório de teste para ajudar a solucionar problemas com seus testes. Se tiver muitos relatórios de teste de várias compilações, você poderá usar seus relatórios de teste para visualizar taxas de falha para ajudá-lo a otimizar suas compilações.

É possível usar os seguintes formatos de arquivo de relatório de testes:

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

## Relatórios de cobertura de código

Em CodeCatalyst, você pode gerar relatórios de cobertura de código para seus testes. CodeCatalyst fornece as seguintes métricas de cobertura de código:

### Cobertura de linha

Mede quantas declarações seus testes abrangem. Declaração é uma instrução única, que não inclui comentários.

$$\text{line coverage} = (\text{total lines covered}) / (\text{total number of lines})$$

### Cobertura de ramificações

Mede quantas ramificações seus testes cobrem de cada ramificação possível de uma estrutura de controle, como uma declaração if ou case.

$$\text{branch coverage} = (\text{total branches covered}) / (\text{total number of branches})$$

Os seguintes formatos de arquivo de relatório de cobertura de código são compatíveis:

- JaCoCo XML (.xml)
- SimpleCov JSON (gerado por [simplecov](#), [não simplecov-json](#), .json)
- Clover XML (versão 3, .xml)
- Cobertura XML (.xml)
- LCOV (.info)

## Relatórios de análise de composição de software

Em CodeCatalyst, você pode usar ferramentas de análise de composição de software (SCA) para analisar componentes do seu aplicativo e verificar se há vulnerabilidades de segurança conhecidas. Você pode descobrir e analisar relatórios do SARIF que detalham vulnerabilidades com severidades variadas e formas de corrigi-las. Os valores de gravidade válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Os seguintes formatos de arquivo de relatório SCA são compatíveis:

- SARIF (.sarif, .json)

## Relatórios de análise estática

Você pode usar relatórios de análise estática (SA) para identificar defeitos no código de origem. Em CodeCatalyst, você pode gerar relatórios de SA para ajudar a resolver problemas em seu código antes de implantá-lo. Esses problemas incluem bugs, vulnerabilidades de segurança, problemas de qualidade e outras vulnerabilidades. Os valores de gravidade válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

CodeCatalyst fornece as seguintes métricas de SA:

### Bugs

Identifica vários possíveis bugs encontrados em seu código de origem. Esses bugs podem incluir problemas relacionados à segurança da memória. Veja a seguir um exemplo de um bug.

```
// The while loop will inadvertently index into array x out-of-bounds
int x[64];
while (int n = 0; n <= 64; n++) {
```

```
x[n] = 0;
}
```

## Vulnerabilidades de segurança

Identifica várias vulnerabilidades de segurança possíveis encontradas em seu código de origem. Essas vulnerabilidades de segurança podem incluir problemas, como armazenar seus tokens secretos em texto simples.

## Problemas de qualidade

Identifica vários possíveis problemas de qualidade encontrados em seu código de origem. Esses problemas de qualidade podem incluir problemas relacionados às convenções de estilo. Veja a seguir um exemplo de um problema de qualidade.

```
// The function name doesn't adhere to the style convention of camelCase
int SUBTRACT(int x, int y) {
 return x-y
}
```

## Outras vulnerabilidades

Identifica várias outras vulnerabilidades possíveis encontradas em seu código de origem.

CodeCatalyst suporta os seguintes formatos de arquivo de relatório SA:

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

## Adição da ação de teste

Use o procedimento a seguir para adicionar uma ação de teste ao seu CodeCatalyst fluxo de trabalho.

### Visual

Como adicionar uma ação de teste usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.



2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione Visual.
6. Escolha Ações.
7. Em Ações, selecione Teste.
8. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [Ações de criação e de teste YAML](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como adicionar uma ação de criação usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione YAML.
6. Escolha Ações.
7. Em Ações, selecione Teste.
8. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [Ações de criação e de teste YAML](#).
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Definição da ação de teste

A ação de teste é definida como um conjunto de propriedades YAML dentro do arquivo de definição do fluxo de trabalho. Para ter mais informações sobre essas propriedades, consulte [Ações de criação e de teste YAML](#) em [Definição do YAML do fluxo de trabalho](#).

## Visualização dos resultados de uma ação de teste

Use as instruções a seguir para visualizar os resultados de uma ação de teste, incluindo os logs, relatórios e variáveis gerados.

Para visualizar os resultados de uma ação de teste

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
3. No diagrama do fluxo de trabalho, selecione o nome da ação de teste, por exemplo, Teste.
4. Para visualizar os logs gerados por uma ação, selecione Logs. Os logs das várias fases de ação são exibidos. Você pode expandir ou recolher os logs conforme necessário.
5. Para visualizar os relatórios de teste produzidos pela ação de teste, selecione Relatórios ou, no painel de navegação, selecione Relatórios. Para obter mais informações, consulte [Tipos de relatório de qualidade](#).
6. Para visualizar a configuração usada para a ação de teste, selecione Configuração. Para obter mais informações, consulte [Adição da ação de teste](#).
7. Para visualizar as variáveis usadas pela ação de teste, selecione Variáveis. Para obter mais informações, consulte [Uso de variáveis em fluxos de trabalho](#).

## Omissão dos testes com falha em uma ação

Se sua ação tiver mais de um comando de teste, talvez você queira permitir que comandos de teste subsequentes na ação sejam executados mesmo se um comando anterior falhar. Por exemplo, nos comandos a seguir, talvez você queira que `test2` execute sempre, mesmo se `test1` falhar.

**Steps:**

- Run: `npm install`
- Run: `npm run test1`
- Run: `npm run test2`

Normalmente, quando uma etapa retorna um erro, a Amazon CodeCatalyst interrompe a ação do fluxo de trabalho e a marca como falhada. Você pode permitir que as etapas da ação continuem sendo executadas redirecionando a saída de erro para `null`. É possível fazer isso adicionando `2>/dev/null` ao comando. Com essa modificação, o exemplo anterior seria semelhante ao seguinte.

**Steps:**

- Run: `npm install`
- Run: `npm run test1 2>/dev/null`
- Run: `npm run test2`

No segundo trecho de código, o status do comando `npm install` será respeitado, mas qualquer erro retornado pelo comando `npm run test1` será ignorado. Como resultado, o comando `npm run test2` é executado. Ao fazer isso, você pode visualizar ambos os relatórios ao mesmo tempo, independentemente de ocorrer um erro.

## Integrando com universal-test-runner

As ações de teste são integradas à ferramenta de linha de comando de código aberto `universal-test-runner`. O `universal-test-runner` usa o [Protocolo de execução de teste](#) para executar seus testes para qualquer linguagem em uma determinada estrutura. O `universal-test-runner` é compatível com as seguintes estruturas:

- [Gradle](#)
- [Jest](#)
- [Maven](#)
- [pytest](#)
- [.NET](#)

O `universal-test-runner` é instalado somente nas imagens selecionadas para ações de teste. Se você configurar uma ação de teste para usar um Docker Hub ou Amazon ECR personalizado, deverá instalar manualmente o `universal-test-runner` para habilitar recursos de teste avançados. Para fazer isso, instale o Node.js (14 ou superior) na imagem e, depois, instale

`universal-test-runner` por meio do npm usando o comando shell - Run: `npm install -g @aws/universal-test-runner`. Para ter mais informações sobre como instalar o Node.js no contêiner por meio de comandos shell, consulte [Installing and Updating Node Version Manager](#).

Para obter mais informações sobre o `universal-test-runner`, consulte [O que é universal-test-runner?](#)

## Visual

Para usar `universal-test-runner` no editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho.
4. Escolha Editar.
5. Selecione Visual.
6. Escolha Ações.
7. Em Ações, selecione Teste.
8. Na guia Configuração, preencha o campo Comandos do Shell atualizando o código de amostra com as estruturas compatíveis de sua escolha. Por exemplo, para usar uma estrutura compatível, você usaria um comando Run semelhante ao seguinte.

```
- Run: run-tests <framework>
```

Se a estrutura que você deseja não for compatível, considere contribuir com um adaptador ou executor personalizado. Para ver uma descrição do campo de comandos do Shell, consulte [Steps](#).

9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para usar `universal-test-runner` no editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho.
4. Escolha Editar.
5. Selecione YAML.
6. Escolha Ações.
7. Em Ações, selecione Teste.
8. Modifique o código YAML de acordo com as suas necessidades. Por exemplo, para usar uma estrutura compatível, você usaria um comando Run semelhante ao seguinte.

```
Configuration:
Steps:
 - Run: run-tests <framework>
```

Se a estrutura que você deseja não for compatível, considere contribuir com um adaptador ou executor personalizado. Para ver uma descrição da propriedade Etapas, consulte [Steps](#).

9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Configuração de relatórios de qualidade em uma ação

Esta seção descreve como configurar um relatório de qualidade em uma ação.

### Tópicos

- [Descoberta automática e relatórios manuais](#)
- [Configuração de critérios de sucesso para relatórios](#)
- [Exemplo de relatórios de qualidade YAML](#)

### Descoberta automática e relatórios manuais

Quando a descoberta automática está ativada, CodeCatalyst pesquisa todas as entradas passadas para a ação e todos os arquivos gerados pela própria ação, procurando relatórios de teste, cobertura de código, análise de composição de software (SCA) e análise estática (SA). Você pode visualizar e manipular cada um desses relatórios em CodeCatalyst.

Você também pode configurar manualmente quais relatórios são gerados. Você pode especificar o tipo de relatório que deseja gerar, bem como o formato do arquivo. Para obter mais informações, consulte [Tipos de relatório de qualidade](#).

## Configuração de critérios de sucesso para relatórios

Defina os valores que determinam os critérios de sucesso para um teste, cobertura de código, análise de composição de software (SCA) ou relatório de análise estática (SA).

Os critérios de sucesso são limites que determinam se um relatório é aprovado ou reprovado. CodeCatalyst primeiro gera seu relatório, que pode ser um relatório de teste, cobertura de código, SCA ou SA, e depois aplica os critérios de sucesso aos relatórios gerados. Em seguida, mostra se os critérios de sucesso foram atendidos e em que medida. Se algum relatório não atender aos critérios de sucesso especificados, a CodeCatalyst ação que especificou os critérios de sucesso falhará.

Por exemplo, quando você define os critérios de sucesso para seu relatório de SCA, os valores de vulnerabilidade válidos que variam do mais ao menos grave são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL. Se você definir os critérios para verificar uma vulnerabilidade com a gravidade HIGH, o relatório falhará se houver pelo menos uma vulnerabilidade com a gravidade HIGH ou nenhuma vulnerabilidade com a gravidade HIGH, mas pelo menos uma vulnerabilidade em um nível de gravidade mais alto, como uma vulnerabilidade em uma gravidade CRITICAL.

Se você não especificar os critérios de sucesso, então:

- O CodeCatalyst relatório gerado com base em seus relatórios brutos não exibirá critérios de sucesso.
- Os critérios de sucesso não serão usados para determinar se a ação do fluxo de trabalho associada será aprovada ou falhará.

## Visual

Para configurar critérios de sucesso

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha um fluxo de trabalho com uma ação que gere um relatório. Este é o relatório ao qual você deseja aplicar os critérios de sucesso. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

3. Escolha Editar.
4. Selecione Visual.
5. No diagrama do fluxo de trabalho, escolha a ação que você configurou para gerar CodeCatalyst relatórios.
6. Escolha a guia Outputs.
7. Em Relatórios de descoberta automática ou em Configurar relatórios manualmente, selecione Critérios de sucesso.

Os critérios de sucesso são exibidos. Dependendo das suas seleções anteriores, você pode ver uma ou todas essas opções:

#### Taxa de aprovação

Especifique a porcentagem de testes em um relatório de teste que devem ser aprovados para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para ter mais informações sobre os relatórios de teste, consulte [Relatórios de teste](#).

#### Cobertura de linha

Especifique a porcentagem de linhas em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

#### Cobertura de ramificações

Especifique a porcentagem de filiais em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de ramificações são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

#### Vulnerabilidades (SCA)

Especifique o número máximo e a gravidade das vulnerabilidades permitidas no relatório SCA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar vulnerabilidades, é necessário especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as vulnerabilidades HIGH e CRITICAL serão contabilizadas.

- O número máximo de vulnerabilidades da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de vulnerabilidade são aplicados somente aos relatórios de SCA. Para ter mais informações sobre os relatórios de SCA, consulte [Relatórios de análise de composição de software](#).

## Bugs

Especifique o número máximo e a gravidade dos bugs permitidos no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar bugs, é necessário especificar:

- A gravidade mínima dos bugs que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, os bugs HIGH e CRITICAL serão contabilizados.

- O número máximo de bugs da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de bugs são aplicados somente PyLint aos relatórios do ESLint SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

## Vulnerabilidades de segurança



Especifique o número máximo e a gravidade das vulnerabilidades de segurança permitidas no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar vulnerabilidades de segurança, é necessário especificar:

- A gravidade mínima das vulnerabilidades de segurança que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as vulnerabilidades de segurança HIGH e CRITICAL serão contabilizadas.

- O número máximo de vulnerabilidades de segurança da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de vulnerabilidades de segurança são aplicados somente aos PyLint relatórios do ESLint SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

### Problemas de qualidade

Especifique o número máximo e a gravidade dos problemas de qualidade permitidos no relatório SA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar problemas de qualidade, é necessário especificar:

- A gravidade mínima dos problemas de qualidade que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, os problemas de qualidade HIGH e CRITICAL serão contabilizados.

- O número máximo de problemas de qualidade da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de problemas de qualidade são aplicados somente PyLint aos relatórios ESLint da SA. Para ter mais informações sobre os relatórios SA, consulte [Relatórios de análise estática](#).

8. Selecione Confirmar.
9. Execute seu fluxo de trabalho para CodeCatalyst aplicar critérios de sucesso aos seus relatórios brutos e regenere os CodeCatalyst relatórios associados com as informações dos critérios de sucesso incluídas. Para obter mais informações, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## YAML

Para configurar critérios de sucesso

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha um fluxo de trabalho com uma ação que gere um relatório. Este é o relatório ao qual você deseja aplicar os critérios de sucesso. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
3. Escolha Editar.
4. Selecione YAML.
5. No diagrama do fluxo de trabalho, escolha a ação que você configurou para gerar CodeCatalyst relatórios.
6. No painel de detalhes, selecione a guia Saídas.
7. Na ação, na AutoDiscoverReports seção ou na Reports seção, adicione uma SuccessCriteria propriedade junto com PassRate, LineCoverage, BranchCoverage, Vulnerabilities, StaticAnalysisBugStaticAnalysisSecurity, e StaticAnalysisQuality propriedades.

Para ver uma explicação de cada uma dessas propriedades, consulte [Ações de criação e de teste YAML](#).

8. Selecione Confirmar.
9. Execute seu fluxo de trabalho para CodeCatalyst aplicar critérios de sucesso aos seus relatórios brutos e regenere os CodeCatalyst relatórios associados com as informações dos critérios de sucesso incluídas. Para ter mais informações sobre como iniciar um fluxo de trabalho, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Exemplo de relatórios de qualidade YAML

O exemplo a seguir mostra como configurar manualmente quatro relatórios: um relatório de teste, um relatório de cobertura de código, um relatório de análise de composição de software e um relatório de análise estática.

```
Reports:
 MyTestReport:
 Format: JUNITXML
 IncludePaths:
 - "*.xml"
 ExcludePaths:
 - report1.xml
 SuccessCriteria:
 PassRate: 90
 MyCoverageReport:
 Format: CLOVERXML
 IncludePaths:
 - output/coverage/jest/clover.xml
 SuccessCriteria:
 LineCoverage: 75
 BranchCoverage: 75
 MySCARReport:
 Format: SARIFSCA
 IncludePaths:
 - output/sca/reports.xml
 SuccessCriteria:
 Vulnerabilities:
 Number: 5
 Severity: HIGH
 MySARReport:
 Format: ESLINTJSON
 IncludePaths:
 - output/static/eslint.xml
 SuccessCriteria:
 StaticAnalysisBug:
 Number: 10
 Severity: MEDIUM
 StaticAnalysisSecurity:
 Number: 5
 Severity: CRITICAL
 StaticAnalysisQuality:
 Number: 0
```

Severity: INFORMATIONAL

## Melhores práticas de teste

Ao usar os recursos de teste fornecidos pelo CodeCatalyst, recomendamos que você siga essas melhores práticas.

### Tópicos

- [Descoberta automática](#)
- [Critérios de sucesso](#)
- [Inclusão/exclusão de caminhos](#)

### Descoberta automática

Ao configurar ações CodeCatalyst, a descoberta automática permite que você descubra automaticamente os resultados de várias ferramentas, como relatórios de JUnit teste, e gere CodeCatalyst relatórios relevantes a partir deles. A descoberta automática ajuda a garantir que os relatórios continuem sendo gerados, mesmo se os nomes ou caminhos para as saídas descobertas mudarem. Quando novos arquivos são adicionados, os descobre CodeCatalyst automaticamente e produz relatórios relevantes. No entanto, se você usa a descoberta automática, é importante pensar em alguns dos seguintes aspectos desse recurso:

- Ao ativar a descoberta automática em sua ação, todos os relatórios descobertos automaticamente do mesmo tipo compartilharão os mesmos critérios de sucesso. Por exemplo, um critério compartilhado, como taxa mínima de aprovação, se aplicaria a todos os relatórios de teste descobertos automaticamente. Se você precisar de critérios diferentes para relatórios do mesmo tipo, configure explicitamente cada um desses relatórios.
- A descoberta automática também pode encontrar relatórios produzidos por suas dependências e, se os critérios de sucesso estiverem configurados, poderá falhar na ação desses relatórios. Esse problema pode ser resolvido atualizando a configuração do caminho de exclusão.
- Não é garantido que a descoberta automática produza sempre a mesma lista de relatórios, pois ela verifica a ação em runtime. Se você quiser que um relatório específico seja sempre produzido, configure os relatórios explicitamente. Por exemplo, se os testes parassem de ser executados como parte de sua compilação, a estrutura de teste não produziria nenhuma saída e, como resultado, nenhum relatório de teste seria produzido e a ação poderia ser bem-sucedida. Se você

quiser que o sucesso de sua ação dependa desse teste específico, configure explicitamente esse relatório.

#### Tip

Ao começar um projeto novo ou existente, use a descoberta automática para todo o diretório do projeto (incluir `**/*`). Isso invoca a geração de relatórios em todos os arquivos do seu projeto, incluindo aqueles em subdiretórios.

Para obter mais informações, consulte [Configuração de relatórios de qualidade em uma ação](#).

## Critérios de sucesso

Você pode impor limites de qualidade em seus relatórios configurando critérios de sucesso. Por exemplo, se dois relatórios de cobertura de código foram descobertos automaticamente, um com uma cobertura de linha de 80% e outro com uma cobertura de linha de 60%, você tem as seguintes opções:

- Defina os critérios de sucesso da descoberta automática para cobertura de linha em 80%. Isso faria com que o primeiro relatório fosse aprovado e o segundo falhasse, o que resultaria na falha geral da ação. Para desbloquear o fluxo de trabalho, adicione novos testes ao seu projeto até que a cobertura da linha para o segundo relatório exceda 80%.
- Defina os critérios de sucesso da descoberta automática para cobertura de linha em 60%. Isso faria com que ambos os relatórios fossem aprovados, o que resultaria no sucesso da ação. Você poderia então trabalhar para aumentar a cobertura do código no segundo relatório. No entanto, com essa abordagem, você não pode garantir que a cobertura no primeiro relatório não caia abaixo de 80%.
- Configure explicitamente um ou ambos os relatórios usando o editor visual ou adicionando uma seção e um caminho YAML explícitos para cada relatório. Isso permitiria que você configurasse critérios de sucesso e nomes personalizados separados para cada relatório. No entanto, com essa abordagem, a ação pode falhar se os caminhos do relatório mudarem.

Para obter mais informações, consulte [Configuração de critérios de sucesso para relatórios](#).

## Inclusão/exclusão de caminhos

Ao analisar os resultados da ação, você pode ajustar a lista de relatórios que são gerados CodeCatalyst pela configuração e. `IncludePaths` `ExcludePaths`

- Use `IncludePaths` para especificar os arquivos e caminhos de arquivo que você CodeCatalyst deseja incluir ao pesquisar relatórios. Por exemplo, se você especificar `"/test/report/*"`, CodeCatalyst pesquisa toda a imagem de compilação usada pela ação procurando o `/test/report/` diretório. Quando encontra esse diretório CodeCatalyst , procura relatórios nesse diretório.

### Note

Em relatórios configurados manualmente, `IncludePaths` deverá ser um padrão global que corresponda a um único arquivo.

- Use `ExcludePaths` para especificar os arquivos e caminhos de arquivo que você CodeCatalyst deseja excluir ao pesquisar relatórios. Por exemplo, se você especificar `"/test/reports/**/*"`, não CodeCatalyst procurará arquivos no `/test/reports/` diretório. Para ignorar todos os arquivos em um diretório, use o padrão de glob `**/*`.

Veja a seguir alguns exemplos de padrões glob possíveis.

| Padrão                   | Description                                                                                                        |
|--------------------------|--------------------------------------------------------------------------------------------------------------------|
| <code>*.*</code>         | Corresponde a todos os nomes de objetos no diretório atual que contêm um ponto                                     |
| <code>*.xml</code>       | Corresponde a todos os nomes de objetos no diretório atual que terminam com <code>.xml</code>                      |
| <code>*.{xml,txt}</code> | Corresponde a todos os nomes de objetos no diretório atual que terminam com <code>.xml</code> ou <code>.txt</code> |
| <code>**/*.xml</code>    | Corresponde aos nomes dos objetos em todos os diretórios que terminam com <code>.xml</code>                        |

| Padrão                      | Description                                                                                                                                                                                           |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>testFolder</code>     | Corresponde a um objeto chamado <code>testFolder</code> , tratando-o como um arquivo                                                                                                                  |
| <code>testFolder/*</code>   | Corresponde a objetos em um nível de subpasta de <code>testFolder</code> , como <code>testFolder/file.xml</code>                                                                                      |
| <code>testFolder/*/*</code> | Corresponde a objetos em dois níveis da subpasta de <code>testFolder</code> , como <code>testFolder/reportsFolder/file.xml</code>                                                                     |
| <code>testFolder/**</code>  | Corresponde a subpasta <code>testFolder</code> bem como aos arquivos abaixo de <code>testFolder</code> , por exemplo, <code>testFolder/file.xml</code> e <code>testFolder/otherFolder/file.xml</code> |

CodeCatalyst interpreta os padrões globais da seguinte forma:

- O caractere de barra (/) separa os diretórios nos caminhos dos arquivos.
- 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.

 Note

`ExcludePaths` tem precedência sobre `IncludePaths`. Se ambos `IncludePaths` e `ExcludePaths` incluírem a mesma pasta, essa pasta não será examinada em busca de relatórios.

## Propriedades SARIF compatíveis

O Static Analysis Results Interchange Format (SARIF) é um formato de arquivo de saída que está disponível em análise de composição de software (SCA) e relatórios de análise estática na Amazon. CodeCatalyst O exemplo a seguir mostra como configurar manualmente o SARIF em um relatório de análise estática:

```
Reports:
MySAReport:
Format: SARIFSA
IncludePaths:
 - output/sa_report.json
SuccessCriteria:
 StaticAnalysisFinding:
 Number: 25
 Severity: HIGH
```

CodeCatalyst suporta as seguintes propriedades SARIF, que podem ser usadas para otimizar a forma como os resultados da análise aparecerão em seus relatórios.

### Tópicos

- [Objeto sarifLog](#)
- [Objeto run](#)
- [Objeto toolComponent](#)
- [Objeto reportingDescriptor](#)
- [Objeto result](#)
- [Objeto location](#)
- [Objeto physicalLocation](#)
- [Objeto logicalLocation](#)
- [Objeto fix](#)



## Objeto **sarifLog**

| Nome                  | Obrigatório | Descrição                                                                                                                    |
|-----------------------|-------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>\$schema</code> | Sim         | O URI do esquema SARIF JSON para a versão <a href="#">2.1.0</a> .                                                            |
| <code>version</code>  | Sim         | CodeCatalyst suporta apenas a versão 2.1.0 do SARIF.                                                                         |
| <code>runs[]</code>   | Sim         | Um arquivo SARIF contém uma ou mais execuções, cada uma das quais representando uma única execução da ferramenta de análise. |

## Objeto **run**

| Nome                     | Obrigatório | Descrição                                                                      |
|--------------------------|-------------|--------------------------------------------------------------------------------|
| <code>tool.driver</code> | Sim         | Um objeto <code>toolComponent</code> que descreve a ferramenta de análise.     |
| <code>tool.name</code>   | Não         | Uma propriedade que indica o nome da ferramenta usada para realizar a análise. |
| <code>results[]</code>   | Sim         | Os resultados da ferramenta de análise que são exibidos em CodeCatalyst.       |

## Objeto **toolComponent**

| Nome                                    | Obrigatório | Descrição                                                                                                                                                                    |
|-----------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>name</code>                       | Sim         | O nome da ferramenta de análise.                                                                                                                                             |
| <code>properties.artifactScanned</code> | Não         | Um número total de artefatos analisados pela ferramenta.                                                                                                                     |
| <code>rules[]</code>                    | Sim         | Uma matriz de objetos <code>reportingDescriptor</code> que representam regras. Com base nessas regras, a ferramenta de análise encontra problemas no código que é analisado. |

## Objeto **reportingDescriptor**

| Nome                               | Obrigatório | Descrição                                                                                                               |
|------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>id</code>                    | Sim         | O identificador exclusivo da regra que é usada para referenciar uma descoberta.<br><br>Tamanho máximo: 1.024 caracteres |
| <code>name</code>                  | Não         | O nome de exibição da regra.<br><br>Tamanho máximo: 1.024 caracteres                                                    |
| <code>shortDescription.text</code> | Não         | Uma descrição resumida da regra.                                                                                        |

| Nome                                        | Obrigatório | Descrição                                                                                                                                       |
|---------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             |             | Tamanho máximo: 3.000 caracteres                                                                                                                |
| <code>fullDescription.text</code>           | Não         | <p>Uma descrição completa da regra.</p> <p>Tamanho máximo: 3.000 caracteres</p>                                                                 |
| <code>helpUri</code>                        | Não         | <p>Uma string que pode ser localizada para conter o URI absoluto da documentação primária da regra.</p> <p>Tamanho máximo: 3.000 caracteres</p> |
| <code>properties.unscore</code>             | Não         | Um sinalizador que indica se a descoberta do escaneamento foi pontuada.                                                                         |
| <code>properties.score.severity</code>      | Não         | <p>Um conjunto fixo de strings que especificam o nível de gravidade da descoberta.</p> <p>Tamanho máximo: 1.024 caracteres</p>                  |
| <code>properties.cvssv3_baseSeverity</code> | Não         | Uma classificação de gravidade qualitativa do <a href="#">Sistema de pontuação de vulnerabilidades comuns v3.1</a> .                            |
| <code>properties.cvssv3_baseScore</code>    | Não         | Uma pontuação básica do CVSS v3 variando de <a href="#">0,0 a 10,0</a> .                                                                        |

| Nome                                    | Obrigatório | Descrição                                                                                                               |
|-----------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>properties.cvssv2_severity</code> | Não         | Se os valores do CVSS v3 não estiverem disponíveis, CodeCatalyst pesquisa os valores do CVSS v2.                        |
| <code>properties.cvssv2_score</code>    | Não         | Uma pontuação básica do CVSS v2 variando de <a href="#">0,0 a 10,0</a> .                                                |
| <code>properties.severity</code>        | Não         | Um conjunto fixo de strings que especificam o nível de gravidade da descoberta.<br><br>Tamanho máximo: 1.024 caracteres |
| <code>defaultConfiguration.level</code> | Não         | A gravidade padrão de uma regra.                                                                                        |

## Objeto **result**

| Nome                      | Obrigatório | Descrição                                                                                                               |
|---------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>ruleId</code>       | Sim         | O identificador exclusivo da regra que é usada para referenciar uma descoberta.<br><br>Tamanho máximo: 1.024 caracteres |
| <code>ruleIndex</code>    | Sim         | O índice da regra associada no componente da ferramenta <code>rules[]</code> .                                          |
| <code>message.text</code> | Sim         | Uma mensagem que descreve o resultado e exibe a                                                                         |

| Nome                           | Obrigatório | Descrição                                                                                                                                                                                   |
|--------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                |             | mensagem de cada descoberta.<br><br>Tamanho máximo: 3.000 caracteres                                                                                                                        |
| rank                           | Não         | Um valor entre 0,0 e 100,0, inclusive, que representa a prioridade ou importância do resultado. Essa escala considera 0,0 como a prioridade mais baixa e 100,0 como a prioridade mais alta. |
| level                          | Não         | A gravidade do resultado.<br><br>Tamanho máximo: 1.024 caracteres                                                                                                                           |
| properties.unscore             | Não         | Um sinalizador que indica se a descoberta do escaneamento foi pontuada.                                                                                                                     |
| properties.score.severity      | Não         | Um conjunto fixo de strings que especificam o nível de gravidade da descoberta.<br><br>Tamanho máximo: 1.024 caracteres                                                                     |
| properties.cvssv3_baseSeverity | Não         | Uma classificação de gravidade qualitativa do <a href="#">Sistema de pontuação de vulnerabilidades comuns v3.1</a> .                                                                        |

| Nome                                     | Obrigatório | Descrição                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>properties.cvssv3_baseScore</code> | Não         | Uma pontuação básica do CVSS v3 variando de <a href="#">0,0 a 10,0</a> .                                                                                                                                                                                                                                                     |
| <code>properties.cvssv2_severity</code>  | Não         | Se os valores do CVSS v3 não estiverem disponíveis, CodeCatalyst pesquisa os valores do CVSS v2.                                                                                                                                                                                                                             |
| <code>properties.cvssv2_score</code>     | Não         | Uma pontuação básica do CVSS v2 variando de <a href="#">0,0 a 10,0</a> .                                                                                                                                                                                                                                                     |
| <code>properties.severity</code>         | Não         | Um conjunto fixo de strings que especificam o nível de gravidade da descoberta.<br><br>Tamanho máximo: 1.024 caracteres                                                                                                                                                                                                      |
| <code>locations[]</code>                 | Sim         | O conjunto de locais em que o resultado foi detectado . Somente um local deve ser incluído, a menos que o problema só possa ser corrigido fazendo uma alteração em cada local especificado. CodeCatalyst usa o primeiro valor na matriz de localização para anotar o resultado.<br><br>Número máximo de objetos location: 10 |

| Nome                            | Obrigatório | Descrição                                                                                                                                                                                |
|---------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>relatedLocations[]</code> | Não         | Uma lista de referências de localizações adicionais na descoberta.<br><br>Número máximo de objetos <code>location</code> : 50                                                            |
| <code>fixes[]</code>            | Não         | Uma matriz de <code>fix</code> objetos que representam a recomendação fornecida pela ferramenta de digitalização. CodeCatalyst usa a primeira recomendação na <code>fixes</code> matriz. |

## Objeto **location**

| Nome                            | Obrigatório | Descrição                                                            |
|---------------------------------|-------------|----------------------------------------------------------------------|
| <code>physicalLocation</code>   | Sim         | Identifica o artefato e a região.                                    |
| <code>logicalLocations[]</code> | Não         | O conjunto de locais descritos pelo nome sem referência ao artefato. |

## Objeto **physicalLocation**

| Nome                              | Obrigatório | Descrição                                                                                                             |
|-----------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>artifactLocation.uri</code> | Sim         | O URI que indica a localização de um artefato, geralmente um arquivo no repositório ou gerado durante uma compilação. |

| Nome                            | Obrigatório | Descrição                                                                                                                 |
|---------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>fileLocation.uri</code>   | Não         | O URI de retorno indicando a localização do arquivo. Isso será usado se <code>artifactLocation.uri</code> retornar vazio. |
| <code>region.startLine</code>   | Sim         | O número da linha do primeiro caractere na região.                                                                        |
| <code>region.startColumn</code> | Sim         | O número da coluna do primeiro caractere na região.                                                                       |
| <code>region.endLine</code>     | Sim         | O número da linha do último caractere na região.                                                                          |
| <code>region.endColumn</code>   | Sim         | O número da coluna do último caractere na região.                                                                         |

## Objeto **logicalLocation**

| Nome                             | Obrigatório | Description                                                                                                     |
|----------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------|
| <code>fullyQualified_name</code> | Não         | <p>Informações adicionais que descrevem a localização do resultado.</p> <p>Tamanho máximo: 1.024 caracteres</p> |



## Objeto **fix**

| Nome                                     | Obrigatório | Description                                                                                           |
|------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------|
| description.text                         | Não         | Uma mensagem que exibe uma recomendação para cada descoberta.<br><br>Tamanho máximo: 3.000 caracteres |
| artifactChanges.[0].artifactLocation.uri | Não         | O URI que indica a localização do artefato que precisa ser atualizado.                                |

## Implantar com fluxos de trabalho

Usando [CodeCatalyst fluxos de trabalho](#), você pode implantar aplicativos e outros recursos em vários destinos, como Amazon ECS e muito mais. AWS Lambda

### Como faço para implantar uma aplicação?

Para implantar um aplicativo ou recurso CodeCatalyst, primeiro você cria um fluxo de trabalho e, em seguida, especifica uma ação de implantação dentro dele. Uma ação de implantação é um componente básico do fluxo de trabalho que define o que você deseja implantar, onde e como deseja implantá-lo (por exemplo, usando um blue/green esquema). Você adiciona uma ação de implantação ao seu fluxo de trabalho usando o editor visual do CodeCatalyst console ou o editor YAML.

As etapas detalhadas para implantar uma aplicação ou um recurso são as seguintes.

Como implantar uma aplicação (tarefas detalhadas)

1. No seu CodeCatalyst projeto, você adiciona o código-fonte de um aplicativo que deseja implantar. Para obter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#).

2. Em seu CodeCatalyst projeto, você adiciona um ambiente que define a Amazon Virtual Private Cloud (VPC) de destino Conta da AWS e opcional na qual você deseja implantar. Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).
3. No seu CodeCatalyst projeto, você cria um fluxo de trabalho. No fluxo de trabalho, você define como criar, testar e implantar a aplicação. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#).
4. No fluxo de trabalho, você adiciona um gatilho, uma ação de criação e, se desejar, uma ação de teste. Para obter mais informações, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#), [Adição da ação de criação](#) e [Adição da ação de teste](#).
5. No fluxo de trabalho, você adiciona uma ação de implantação. Você pode escolher entre várias ações de implantação CodeCatalyst fornecidas para seu aplicativo em diferentes destinos, como o Amazon ECS. (Você também pode usar uma ação de compilação ou uma GitHub ação para implantar seu aplicativo. Para obter mais informações sobre a ação de criação e GitHub as ações, consulte [Alternativas para ações de implantação](#).)
6. Você inicia o fluxo de trabalho manual ou automaticamente por meio de um gatilho. O fluxo de trabalho executa as ações de criação, teste e implantação em sequência para implantar a aplicação e os recursos no destino. Para obter mais informações, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Lista de ações de implantação

As seguintes ações de implantação estão disponíveis:

- Implantar CloudFormation pilha — Essa ação cria uma CloudFormation pilha AWS com base em um [CloudFormation modelo ou AWS Serverless Application Model modelo](#) fornecido por você. Para obter mais informações, consulte [Implantação de uma pilha CloudFormation](#).
- Implantar no Amazon ECS - Essa ação registra um arquivo de [definição de tarefa](#) que você fornece. Para obter mais informações, consulte [Implantação no Amazon ECS com um fluxo de trabalho](#).
- Implantar no cluster do Kubernetes - Essa ação implanta uma aplicação em um cluster do Amazon Elastic Kubernetes Service. Para obter mais informações, consulte [Implantar no Amazon EKS com um fluxo de trabalho](#).
- AWS CDK implantar — Essa ação implanta um [AWS CDK aplicativo](#) em AWS. Para obter mais informações, consulte [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#).

 Note

Há outras CodeCatalyst ações que podem implantar recursos; no entanto, elas não são consideradas ações de implantação porque suas informações de implantação não aparecem na página Ambientes. Para saber mais sobre a página Ambientes e ver as implantações, consulte [Implantação em e Contas da AWS VPCs](#) e [Visualizar informações de implantação](#).

## Benefícios das ações de implantação

O uso de ações de implantação em um fluxo de trabalho fornece os seguintes benefícios:

- **Histórico de implantação:** visualize um histórico de suas implantações para ajudar a gerenciar e comunicar as alterações no software implantado.
- **Rastreabilidade** — Acompanhe o status de suas implantações por meio do CodeCatalyst console e veja quando e onde cada revisão do aplicativo foi implantada.
- **Reversões:** reverta as implantações automaticamente se houver erros. Também é possível configurar alarmes para ativar reversões de implantação.
- **Monitoramento** — Observe sua implantação conforme ela progride nos vários estágios do fluxo de trabalho.
- **Integração com outros CodeCatalyst recursos** — armazene o código-fonte e, em seguida, crie, teste e implante, tudo em um único aplicativo.

## Alternativas para ações de implantação

Você não precisa usar ações de implantação, embora elas sejam recomendadas porque oferecem os benefícios descritos na seção anterior. Em vez disso, você pode usar as seguintes [CodeCatalyst ações](#):

- Uma ação de criação.

Normalmente, você usa ações de criação quando quer implantar em um destino para o qual não existe uma ação de implantação correspondente ou quando quer ter mais controle sobre o procedimento de implantação. Para ter mais informações sobre como usar ações de criação para implantar recursos, consulte [Criação com fluxos de trabalho](#).

- Uma GitHub ação.

Você pode usar uma [GitHub ação](#) dentro de um CodeCatalyst fluxo de trabalho para implantar aplicativos e recursos (em vez de uma CodeCatalyst ação). Para obter informações sobre como usar GitHub ações em um CodeCatalyst fluxo de trabalho, consulte [Integração com GitHub ações](#)

Você também pode usar os seguintes AWS serviços para implantar seu aplicativo, se não quiser usar um CodeCatalyst fluxo de trabalho para fazer isso:

- AWS CodeDeploy — veja [O que é CodeDeploy?](#)
- AWS CodeBuild e AWS CodePipeline — veja [O que é AWS CodeBuild?](#) e [o que é AWS CodePipeline?](#)
- CloudFormation — veja [O que é CloudFormation?](#)

Use CodeDeploy, CodeBuild CodePipeline, e CloudFormation serviços para implantações corporativas complexas.

## Tópicos

- [Implantação no Amazon ECS com um fluxo de trabalho](#)
- [Implantar no Amazon EKS com um fluxo de trabalho](#)
- [Implantação de uma pilha CloudFormation](#)
- [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#)
- [Inicializando um AWS CDK aplicativo com um fluxo de trabalho](#)
- [Publicação de arquivos no Amazon S3 com um fluxo de trabalho](#)
- [Implantação em e Contas da AWS VPCs](#)
- [Exibir o URL da aplicação no diagrama do fluxo de trabalho](#)
- [Remoção de um destino de implantação](#)
- [Rastreamento do status de implantação por confirmação](#)
- [Visualização dos logs de implantação](#)
- [Visualizar informações de implantação](#)

## Implantação no Amazon ECS com um fluxo de trabalho

Esta seção descreve como implantar um aplicativo em contêiner em um cluster do Amazon Elastic Container Service usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar

a ação Implantar no Amazon ECS ao seu fluxo de trabalho. Essa ação registra um arquivo de [definição de tarefa](#) que você fornece. Após o registro, a definição da tarefa é instanciada pelo [serviço do Amazon ECS](#) em execução no [cluster do Amazon ECS](#). “Instanciar uma definição de tarefa” é equivalente a implantar uma aplicação no Amazon ECS.

Para usar essa ação, você deve ter um cluster, serviço e arquivo de definição de tarefas do Amazon ECS pronto.

Para ter mais informações sobre o Amazon ECS, consulte o Guia do desenvolvedor do Amazon Elastic Container Service.

 Tip

Para ver um tutorial que mostra como usar a ação Implantar no Amazon ECS, consulte [Tutorial: Implantar uma aplicação no Amazon ECS](#).

 Tip

Para ter um exemplo prático da ação Implantar no Amazon ECS, crie um projeto com o esquema API Node.js com AWS Fargate ou a API Java com AWS Fargate. Para obter mais informações, consulte [Criar um projeto com um esquema](#).

## Tópicos

- [Imagem de runtime usada pela ação “Implantar no Amazon ECS”](#)
- [Tutorial: Implantar uma aplicação no Amazon ECS](#)
- [Adição da ação “Implantar no Amazon ECS”](#)
- [Variáveis de “Implantar no Amazon ECS”](#)
- [YAML de ação “Implantar no Amazon ECS”](#)

## Imagem de runtime usada pela ação “Implantar no Amazon ECS”

A ação Implantar no Amazon ECS é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Tutorial: Implantar uma aplicação no Amazon ECS

Neste tutorial, você aprende a implantar um aplicativo sem servidor no Amazon Elastic Container Service (Amazon ECS) usando um fluxo de trabalho, o Amazon ECS e alguns outros serviços. AWS A aplicação implantada é um site simples do Hello World construído em uma imagem do Docker do servidor da Web Apache. O tutorial explica o trabalho de preparação necessário, como a configuração de um cluster, e descreve como criar um fluxo de trabalho para criar e implantar a aplicação.

### Tip

Em vez de seguir este tutorial, você pode usar um esquema que faz uma configuração completa do Amazon ECS para você. Você precisará usar o esquema API Node.js com AWS Fargate ou API Java com AWS Fargate. Para obter mais informações, consulte [Criar um projeto com um esquema](#).

### Tópicos

- [Pré-requisitos](#)
- [Etapa 1: configurar um AWS usuário e AWS CloudShell](#)
- [Etapa 2: implantar uma aplicação de espaço reservado no Amazon ECS](#)
- [Etapa 3: criar uma imagem de repositório do Amazon ECR](#)
- [Etapa 4: criar AWS funções](#)
- [Etapa 5: adicionar AWS funções a CodeCatalyst](#)
- [Etapa 6: criar um repositório de origem](#)
- [Etapa 7: adicionar arquivos de origem](#)
- [Etapa 8: criar e executar um fluxo de trabalho](#)
- [Etapa 9: realizar uma alteração nos arquivos de origem](#)
- [Limpeza](#)

### Pré-requisitos

#### Antes de começar

- Você precisa de um CodeCatalyst espaço com uma AWS conta conectada. Para obter mais informações, consulte [Criar um espaço](#).

- Em seu espaço, você precisa de um projeto vazio chamado:

```
codecatalyst-ecs-project
```

Use a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- Em seu projeto, você precisa de um CodeCatalyst ambiente chamado:

```
codecatalyst-ecs-environment
```

Configure esse ambiente da seguinte forma:

- Escolha qualquer tipo, como Não produção.
- Conecte sua AWS conta a ela.
- Para o Perfil do IAM padrão, escolha qualquer perfil. Você especificará um perfil diferente posteriormente.

Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).

## Etapa 1: configurar um AWS usuário e AWS CloudShell

A primeira etapa deste tutorial é criar um usuário e executar uma AWS CloudShell instância como esse usuário. Centro de Identidade do AWS IAM Durante este tutorial, CloudShell é o seu computador de desenvolvimento e é onde você configura AWS recursos e serviços. Exclua esse usuário depois de concluir o tutorial.

### Note

Não use seu usuário-raiz neste tutorial. Você deve criar um usuário separado, caso contrário poderá ter problemas ao executar ações na AWS Command Line Interface (CLI) posteriormente.

Para obter mais informações sobre os usuários do IAM Identity Center e CloudShell consulte o Guia Centro de Identidade do AWS IAM do usuário e o Guia AWS CloudShell do usuário.

## Como criar um usuário do Centro de Identidade do IAM

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

### Note

Certifique-se de fazer login usando o Conta da AWS que está conectado ao seu CodeCatalyst espaço. Você pode verificar qual conta está conectada navegando até seu espaço e escolhendo a guia Contas da AWS. Para obter mais informações, consulte [Criar um espaço](#).

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em Nome de usuário, digite:

**CodeCatalystECSUser**

4. Em Senha, selecione Gerar uma senha de uso único que você possa compartilhar com esse usuário.
5. Em Endereço de e-mail e Confirmar endereço de e-mail, insira um endereço de e-mail que ainda não exista no IAM Identity Center.
6. Em Nome e Sobrenome, digite:

**CodeCatalystECSUser**

7. Em Nome de exibição, mantenha o nome gerado automaticamente:

**CodeCatalystECSUser CodeCatalystECSUser**

8. Escolha Próximo.
9. Na página Adicionar usuário a grupos, selecione Avançar.
10. Na página Revisar e adicionar usuário, revise as informações e selecione Adicionar usuário.

Uma caixa de diálogo Senha de uso único é exibida.

11. Selecione Copiar e cole as informações de login, incluindo o URL do portal de acesso da AWS e a senha de uso único.
12. Escolha Fechar.



## Para criar um conjunto de permissões

Você atribuirá esse conjunto de permissões a CodeCatalystECSUser posteriormente.

1. No painel de navegação, escolha Conjuntos de permissões e selecione Criar conjunto de permissões.
2. Escolha Conjunto de permissões predefinido e, em seguida, selecione AdministratorAccess. Esta política fornece permissões totais para todos os Serviços da AWS.
3. Escolha Próximo.
4. Em Nome do conjunto de permissões, digite:

**CodeCatalystECSPermissionSet**

5. Escolha Próximo.
6. Na página Revisar e criar, revise as informações e selecione Criar.

## Para atribuir o conjunto de permissões a CodeCatalyst ECSUser

1. No painel de navegação Contas da AWS, escolha e marque a caixa de seleção ao lado da caixa de seleção na Conta da AWS qual você está conectado no momento.
2. Escolha Atribuir usuários ou grupos.
3. Escolha a guia Users.
4. Marque a caixa de seleção ao lado dos logs do CodeCatalystECSUser.
5. Escolha Próximo.
6. Marque a caixa de seleção ao lado dos logs do CodeCatalystECSPermissionSet.
7. Escolha Próximo.
8. Revise suas informações e selecione Enviar.

Agora você designou CodeCatalystECSUser e CodeCatalystECSPermissionSet para o seu Conta da AWS, unindo-os.

## Para sair e entrar novamente como CodeCatalyst ECSUser

1. Antes de sair, verifique se você tem a URL do portal de AWS acesso, o nome de usuário e a senha de uso único para CodeCatalystECSUser. Você deveria ter copiado essas informações para um editor de texto anteriormente.

**Note**

Se você não tiver essas informações, acesse a página de detalhes do CodeCatalystECSUser no IAM Identity Center, selecione Redefinir senha, Gerar uma senha de uso único [...] e Redefinir senha novamente para exibir as informações na tela.

2. Sair de AWS.
3. Cole o URL do portal de AWS acesso na barra de endereço do seu navegador.
4. Faça login com o nome de usuário e a senha de uso único para CodeCatalystECSUser.
5. Em Nova senha, insira uma senha e selecione Definir nova senha.

Uma caixa de Conta da AWS aparece na tela.

6. Escolha e Conta da AWS, em seguida, escolha o nome do Conta da AWS ao qual você atribuiu o CodeCatalystECSUser usuário e o conjunto de permissões.
7. Ao lado de CodeCatalystECSPermissionSet, selecione Console de gerenciamento.

O Console de gerenciamento da AWS aparece. Agora você está conectado como CodeCatalystECSUser com as permissões apropriadas.

Para iniciar uma AWS CloudShell instância

1. ComoCodeCatalystECSUser, na barra de navegação superior, escolha o AWS ícone



).

A página principal do Console de gerenciamento da AWS aparece.

2. Na barra de navegação superior, escolha o AWS CloudShell ícone



).

CloudShell abre. Aguarde enquanto o CloudShell ambiente é criado.

**Note**

Se você não vê o CloudShell ícone, verifique se você está em uma [região suportada pelo CloudShell](#). Este tutorial pressupõe que você esteja na região do Oeste dos EUA (Oregon).

## Para verificar se o AWS CLI está instalado

1. No CloudShell terminal, digite:

```
aws --version
```

2. Confira se uma versão é exibida.

O já AWS CLI está configurado para o usuário atual `CodeCatalystECSUser`, portanto, não há necessidade de configurar AWS CLI chaves e credenciais, como é normalmente o caso.

## Etapa 2: implantar uma aplicação de espaço reservado no Amazon ECS

Nesta seção, você implanta manualmente uma aplicação de espaço reservado no Amazon ECS. Essa aplicação de espaço reservado será substituída pela aplicação Hello World implantada pelo fluxo de trabalho. A aplicação de espaço reservado é o Apache Web Server.

Para ter mais informações sobre o Amazon ECS, consulte o Guia do desenvolvedor do Amazon Elastic Container Service.

Conclua a série de procedimentos a seguir para implantar a aplicação de espaço reservado.

### Como criar o perfil de execução de tarefas

Essa função concede ao Amazon ECS AWS Fargate permissão para fazer chamadas de API em seu nome.

1. Crie uma política de confiança:
  - a. Em AWS CloudShell, digite o seguinte comando:

```
cat > codecatalyst-ecs-trust-policy.json
```

Um aviso piscando aparece no CloudShell terminal.

- b. No prompt, insira o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
```

```
"Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "ecs-tasks.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

- c. Coloque o cursor após o último colchete (}).
  - d. Pressione **Enter** e, depois, **Ctrl+d** para salvar o arquivo e sair do cat.
2. Crie um perfil de execução de tarefas:

```
aws iam create-role \
 --role-name codecatalyst-ecs-task-execution-role \
 --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

3. Anexe a AmazonECSTaskExecutionRolePolicy política AWS gerenciada à função:

```
aws iam attach-role-policy \
 --role-name codecatalyst-ecs-task-execution-role \
 --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
```

4. Exiba os detalhes do perfil:

```
aws iam get-role \
 --role-name codecatalyst-ecs-task-execution-role
```

5. Observe o valor "Arn": do perfil, por exemplo, arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role. Você precisará desse nome do recurso da Amazon (ARN) posteriormente.

Para criar um cluster do Amazon ECS

Esse cluster conterá a aplicação de espaço reservado Apache e, posteriormente, a aplicação Hello World.

1. ComoCodeCatalystECSUser, em AWS CloudShell, crie um cluster vazio:

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

2. (Opcional) Verifique se o cluster foi criado:

```
aws ecs list-clusters
```

O ARN do cluster codecatalyst-ecs-cluster deve aparecer na lista, indicando uma criação bem-sucedida.

Para criar um arquivo de definição de tarefa

O arquivo de definição da tarefa indica a execução da imagem Docker (httpd:2.4) [do servidor Web Apache 2.4](#), da qual foi extraída. DockerHub

1. ComoCodeCatalystECSUser, em AWS CloudShell, crie um arquivo de definição de tarefa:

```
cat > taskdef.json
```

2. No prompt, cole o seguinte código:

```
{
 "executionRoleArn": "arn:aws:iam::111122223333:role/codecatalyst-ecs-task-
execution-role",
 "containerDefinitions": [
 {
 "name": "codecatalyst-ecs-container",
 "image": "httpd:2.4",
 "essential": true,
 "portMappings": [
 {
 "hostPort": 80,
 "protocol": "tcp",
 "containerPort": 80
 }
]
 }
],
 "requiresCompatibilities": [
 "FARGATE"
]
}
```

```
],
 "cpu": "256",
 "family": "codecatalyst-ecs-task-def",
 "memory": "512",
 "networkMode": "awsvpc"
}
```

No código anterior, substitua *arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role*

pelo ARN do perfil de execução da tarefa que você anotou em [Como criar o perfil de execução de tarefas](#).

3. Coloque o cursor após o último colchete (}).
4. Pressione **Enter** e, depois, **Ctrl+d** para salvar o arquivo e sair do cat.

Para registrar o arquivo definição de tarefa com o Amazon ECS

1. ComoCodeCatalystECSUser, em AWS CloudShell, registre a definição da tarefa:

```
aws ecs register-task-definition \
 --cli-input-json file://taskdef.json
```

2. (Opcional) Verifique se a definição da tarefa foi registrada:

```
aws ecs list-task-definitions
```

A definição da tarefa `codecatalyst-ecs-task-def` deve aparecer na lista.

Para criar o serviço do Amazon ECS

O serviço Amazon ECS executa as tarefas (e os contêineres do Docker associados) da aplicação de espaço reservado Apache e, posteriormente, da aplicação Hello World.

1. Como CodeCatalystECSUser, mude para o console do Amazon Elastic Container Service, se você ainda não tiver feito isso.
2. Selecione o cluster que você criou antes, `codecatalyst-ecs-cluster`.
3. Na guia Serviços, selecione Criar.
4. Na página Criar, faça o seguinte:

- a. Mantenha todas as configurações padrão, exceto as listadas a seguir.
- b. Para Launch type (Tipo de inicialização), escolha FARGATE.
- c. Em Definição de tarefa, na lista suspensa Família, escolha:

`codecatalyst-ecs-task-def`

- d. Em Nome do serviço, digite:

`codecatalyst-ecs-service`

- e. Em Tarefas desejadas, digite:

`3`

Neste tutorial, cada tarefa inicia um único contêiner do Docker.

- f. Expanda a seção Rede.
- g. Em VPC, escolha qualquer VPC.
- h. Em Sub-redes, escolha qualquer sub-rede.

 Note

Especifique somente uma sub-rede. Isso é tudo o que é necessário para este tutorial.

 Note

Se você não tiver uma VPC e uma sub-rede, crie-as. Consulte [Criar uma VPC](#) e [Criar uma sub-rede na VPC](#) no Guia do usuário da Amazon VPC.

- i. Em Grupo de segurança, selecione Criar um grupo de segurança e faça o seguinte:
  - i. Em Nome do grupo de segurança, digite:

`codecatalyst-ecs-security-group`

- ii. Em Descrição do grupo de segurança, digite:

**CodeCatalyst ECS security group**

- iii. Escolha Adicionar regra. Em Tipo, selecione HTTP e, em Origem, selecione Qualquer lugar.
  - j. Selecione Criar na parte inferior.
  - k. Aguarde enquanto o serviço é criado. Isso pode levar alguns minutos.
5. Escolha a guia Tarefas e escolha o botão de atualização. Verifique se as três tarefas têm a coluna Último status definida como Em execução.

(Opcional) Para verificar se a aplicação de espaço reservado Apache está em execução

1. Na guia Tarefas, escolha qualquer uma das três tarefas.
2. No campo IP público, escolha endereço aberto.

Uma página *It Works!* é exibida. Isso indica que o serviço Amazon ECS iniciou uma tarefa que executou um contêiner do Docker com a imagem do Apache.

Neste ponto do tutorial, você implantou manualmente uma definição de cluster, serviço e tarefa do Amazon ECS, bem como uma aplicação de espaço reservado do Apache. Com todos esses itens prontos, agora você já pode criar um fluxo de trabalho que substituirá a aplicação de espaço reservado Apache pela aplicação Hello World do tutorial.

### Etapa 3: criar uma imagem de repositório do Amazon ECR

Nesta seção, você cria um repositório de imagens privadas no Amazon Elastic Container Registry (Amazon ECR). Esse repositório armazena a imagem do Docker do tutorial que substituirá a imagem de espaço reservado do Apache que você implantou anteriormente.

Para obter mais informações sobre o Amazon ECR, consulte o Guia do usuário do Amazon Elastic Container Registry.

Para criar um repositório de imagens no Amazon ECR

1. Por `CodeCatalystECSUser` exemplo AWS CloudShell, crie um repositório vazio no Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-ecs-image-repo
```



## 2. Exiba os detalhes do repositório do Amazon ECR:

```
aws ecr describe-repositories \
 --repository-names codecatalyst-ecs-image-repo
```

## 3. Observe o valor “repositoryUri”: , por exemplo, 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo.

Você precisará dele mais tarde ao adicionar o repositório ao seu fluxo de trabalho.

## Etapa 4: criar AWS funções

Nesta seção, você cria funções AWS do IAM que seu CodeCatalyst fluxo de trabalho precisará para funcionar. Esses perfis são:

- Criar função — Concede permissão à ação de CodeCatalyst criação (no fluxo de trabalho) para acessar sua AWS conta e gravar no Amazon ECR e na Amazon EC2.
- Função de implantação — concede à ação CodeCatalyst Deploy to ECS (no fluxo de trabalho) permissão para acessar sua AWS conta, o Amazon ECS e alguns outros AWS serviços.

Para ter informações sobre perfis do IAM, consulte [Perfis do IAM](#) no Guia do usuário do AWS Identity and Access Management .

### Note

Para economizar tempo, você pode criar um único perfil, chamado CodeCatalystWorkflowDevelopmentRole-*spaceName*, em vez dos dois perfis listados anteriormente. Para obter mais informações, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* tem permissões bem amplas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando os dois perfis listados anteriormente.

Para criar as funções de criação e implantação, você pode usar Console de gerenciamento da AWS o. ou AWS CLI o.

## Console de gerenciamento da AWS

Para criar os perfis de criação e implantação, conclua a série de procedimentos a seguir.

### Como criar um perfil de criação

1. Crie uma política para o perfil da seguinte maneira:
  - a. Faça login em AWS.
  - b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
  - c. No painel de navegação, selecione Políticas.
  - d. Escolha Criar política.
  - e. Escolha a guia JSON.
  - f. Exclua o código existente.
  - g. Cole o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecr:*",
 "ec2:*"
],
 "Resource": "*"
 }
]
}
```

#### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-ecs-build-policy

```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure `codecatalyst-ecs-build-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

```
codecatalyst-ecs-build-role

```

- i. Em Descrição do perfil, insira:

```
CodeCatalyst ECS build role

```

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de permissões e uma política de confiança.

3. Obtenha o ARN do perfil de criação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (codecatalyst-ecs-build-role).
- c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN. Você precisará disso mais tarde.

### Para criar um perfil de implantação

1. Crie uma política para o perfil da seguinte maneira:

- a. Faça login em AWS.
- b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, selecione Políticas.
- d. Escolha Create Policy.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Action": [
 "ecs:DescribeServices",
 "ecs:CreateTaskSet",
 "ecs>DeleteTaskSet",
 "ecs:ListClusters",
 "ecs:RegisterTaskDefinition",
 "ecs:UpdateServicePrimaryTaskSet",
 "ecs:UpdateService",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeListeners",
 "elasticloadbalancing:ModifyListener",
 "elasticloadbalancing:DescribeRules",
```

```

 "elasticloadbalancing:ModifyRule",
 "lambda:InvokeFunction",
 "lambda:ListFunctions",
 "cloudwatch:DescribeAlarms",
 "sns:Publish",
 "sns:ListTopics",
 "s3:GetObject",
 "s3:GetObjectVersion",
 "codedeploy:CreateApplication",
 "codedeploy:CreateDeployment",
 "codedeploy:CreateDeploymentGroup",
 "codedeploy:GetApplication",
 "codedeploy:GetDeployment",
 "codedeploy:GetDeploymentGroup",
 "codedeploy:ListApplications",
 "codedeploy:ListDeploymentGroups",
 "codedeploy:ListDeployments",
 "codedeploy:StopDeployment",
 "codedeploy:GetDeploymentTarget",
 "codedeploy:ListDeploymentTargets",
 "codedeploy:GetDeploymentConfig",
 "codedeploy:GetApplicationRevision",
 "codedeploy:RegisterApplicationRevision",
 "codedeploy:BatchGetApplicationRevisions",
 "codedeploy:BatchGetDeploymentGroups",
 "codedeploy:BatchGetDeployments",
 "codedeploy:BatchGetApplications",
 "codedeploy:ListApplicationRevisions",
 "codedeploy:ListDeploymentConfigs",
 "codedeploy:ContinueDeployment"
],
 "Resource": "*",
 "Effect": "Allow"
 }, { "Action": [
 "iam:PassRole"
],
 "Effect": "Allow",
 "Resource": "*",
 "Condition": { "StringLike": { "iam:PassedToService": [
 "ecs-tasks.amazonaws.com",
 "codedeploy.amazonaws.com"
]
 }
 }
}

```

```
}]
}
```

 **Note**

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos. Em seguida, você pode definir o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-ecs-deploy-policy
```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

- 2. Crie o perfil de implantação, da seguinte forma:
  - a. No painel de navegação, escolha Perfis e Criar perfil.
  - b. Selecione Política de confiança personalizada.
  - c. Exclua a política de confiança personalizada existente.
  - d. Adicione a política de confiança personalizada a seguir:
  - e. Escolha Próximo.
  - f. Em Políticas de permissões, procure `codecatalyst-ecs-deploy-policy` e marque a caixa de seleção.
  - g. Escolha Próximo.
  - h. Em Nome do perfil, insira:

```
codecatalyst-ecs-deploy-role
```

- i. Em Descrição do perfil, insira:

```
CodeCatalyst ECS deploy role
```

- j. Selecione Criar perfil.

Agora você criou um perfil de implantação com uma política de confiança.

3. Obtenha o ARN do perfil de implantação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (codecatalyst-ecs-deploy-role).
- c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN. Você precisará disso mais tarde.

## AWS CLI

Para criar os perfis de criação e implantação, conclua a série de procedimentos a seguir.

Para criar uma política de confiança para os dois perfis

Por CodeCatalystECSUser exemplo AWS CloudShell, crie um arquivo de política de confiança:

1. Crie o arquivo:

```
cat > codecatalyst-ecs-trust-policy.json
```

2. No prompt do terminal, cole o seguinte código:
3. Coloque o cursor após o último colchete (}).
4. Pressione **Enter** e, depois, **Ctrl+d** para salvar o arquivo e sair do cat.

Para criar a política de criação e o perfil de criação

1. Crie a política de criação:

- a. Por CodeCatalystECSUser exemplo AWS CloudShell, crie um arquivo de política de compilação:

```
cat > codecatalyst-ecs-build-policy.json
```

- b. No prompt, insira o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecr:*",
 "ec2:*"
],
 "Resource": "*"
 }
]
}
```

- c. Coloque o cursor após o último colchete (}).
  - d. Pressione **Enter** e, depois, **Ctrl+d** para salvar o arquivo e sair do cat.
2. Adicione a política de compilação a AWS:

```
aws iam create-policy \
 --policy-name codecatalyst-ecs-build-policy \
 --policy-document file:///codecatalyst-ecs-build-policy.json
```

3. Na saída do comando, observe o valor "arn":, por exemplo, arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy. Esse ARN será necessário posteriormente.
4. Crie o perfil de criação e anexe a política de confiança a ele:

```
aws iam create-role \
 --role-name codecatalyst-ecs-build-role \
 --assume-role-policy-document file:///codecatalyst-ecs-trust-policy.json
```



5. Anexe a política de criação à função de criação:

```
aws iam attach-role-policy \
 --role-name codecatalyst-ecs-build-role \
 --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy
```

Onde *arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy* é substituído pelo ARN da política de construção que você observou anteriormente.

6. Exiba os detalhes do perfil de criação:

```
aws iam get-role \
 --role-name codecatalyst-ecs-build-role
```

7. Observe o valor "Arn": do perfil, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role`. Esse ARN será necessário posteriormente.

Para criar a política e o perfil de implantação

1. Crie uma política de implantação:

a. Em AWS CloudShell, crie um arquivo de política de implantação:

```
cat > codecatalyst-ecs-deploy-policy.json
```

b. No prompt, insira o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Action": [
 "ecs:DescribeServices",
 "ecs:CreateTaskSet",
 "ecs>DeleteTaskSet",
 "ecs:ListClusters",
 "ecs:RegisterTaskDefinition",
 "ecs:UpdateServicePrimaryTaskSet",
 "ecs:UpdateService",
]
 }]
```

```

 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeListeners",
 "elasticloadbalancing:ModifyListener",
 "elasticloadbalancing:DescribeRules",
 "elasticloadbalancing:ModifyRule",
 "lambda:InvokeFunction",
 "lambda:ListFunctions",
 "cloudwatch:DescribeAlarms",
 "sns:Publish",
 "sns:ListTopics",
 "s3:GetObject",
 "s3:GetObjectVersion",
 "codedeploy:CreateApplication",
 "codedeploy:CreateDeployment",
 "codedeploy:CreateDeploymentGroup",
 "codedeploy:GetApplication",
 "codedeploy:GetDeployment",
 "codedeploy:GetDeploymentGroup",
 "codedeploy:ListApplications",
 "codedeploy:ListDeploymentGroups",
 "codedeploy:ListDeployments",
 "codedeploy:StopDeployment",
 "codedeploy:GetDeploymentTarget",
 "codedeploy:ListDeploymentTargets",
 "codedeploy:GetDeploymentConfig",
 "codedeploy:GetApplicationRevision",
 "codedeploy:RegisterApplicationRevision",
 "codedeploy:BatchGetApplicationRevisions",
 "codedeploy:BatchGetDeploymentGroups",
 "codedeploy:BatchGetDeployments",
 "codedeploy:BatchGetApplications",
 "codedeploy:ListApplicationRevisions",
 "codedeploy:ListDeploymentConfigs",
 "codedeploy:ContinueDeployment"
],
 "Resource": "*",
 "Effect": "Allow"
}, {"Action": [
 "iam:PassRole"
],
 "Effect": "Allow",
 "Resource": "*",
 "Condition": {"StringLike": {"iam:PassedToService": [
 "ecs-tasks.amazonaws.com",

```

```

 "codedeploy.amazonaws.com"
]
}
}
}]
}

```

 **Note**

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- c. Coloque o cursor após o último colchete (}).
- d. Pressione **Enter** e, depois, **Ctrl+d** para salvar o arquivo e sair do cat.

2. Adicione a política de implantação a AWS:

```
aws iam create-policy \
 --policy-name codecatalyst-ecs-deploy-policy \
 --policy-document file://codecatalyst-ecs-deploy-policy.json
```

3. Na saída do comando, observe o valor "arn": da política de implantação, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy`. Esse ARN será necessário posteriormente.
4. Crie o perfil de implantação e anexe a política de confiança a ele:

```
aws iam create-role \
 --role-name codecatalyst-ecs-deploy-role \
 --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

5. Anexe a política de implantação à função de implantação, que `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` é substituída pelo ARN da política de implantação que você anotou anteriormente.

```
aws iam attach-role-policy \
 --role-name codecatalyst-ecs-deploy-role \
```

```
--policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy
```

6. Exiba os detalhes do perfil de implantação:

```
aws iam get-role \
 --role-name codecatalyst-ecs-deploy-role
```

7. Observe o valor "Arn": do perfil, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role`. Esse ARN será necessário posteriormente.

## Etapa 5: adicionar AWS funções a CodeCatalyst

Nesta etapa, você adiciona a função de criação (`codecatalyst-ecs-build-role`) e a função de implantação (`codecatalyst-ecs-deploy-role`) à conexão da CodeCatalyst conta em seu espaço.

Para adicionar perfis de criação e implantação à conexão da sua conta

1. Dentro CodeCatalyst, navegue até seu espaço.
2. Selecione Contas da AWS . Uma lista de conexões de conta é exibida.
3. Escolha a conexão de conta que representa a AWS conta em que você criou suas funções de criação e implantação.
4. Escolha Gerenciar funções no console de AWS gerenciamento.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é exibida. Talvez seja necessário fazer login para acessá-la.

5. Selecione Adicionar um perfil existente que você criou no IAM.

Uma lista suspensa aparece. A lista exibe todos os perfis do IAM com uma política de confiança que inclui as entidades principais de serviço `codecatalyst-runner.amazonaws.com` e `codecatalyst.amazonaws.com`.

6. Na lista suspensa, selecione `codecatalyst-ecs-build-role` e Adicionar perfil.

### Note

Se você vir `The security token included in the request is invalid`, pode ser porque você não tem as permissões corretas. Para corrigir esse problema,

saia e faça login novamente com a AWS conta que você usou ao criar seu CodeCatalyst espaço. AWS

- Escolha Adicionar perfil do IAM, selecione Adicionar um perfil existente que você criou no IAM e, na lista suspensa, escolha `codecatalyst-ecs-deploy-role`. Escolha Add role (adicionar função).

Agora você adicionou os perfis de criação e implantação ao seu espaço.

- Copie o valor do nome de CodeCatalyst exibição da Amazon. Esse valor será necessário posteriormente, ao criar seu fluxo de trabalho.

## Etapa 6: criar um repositório de origem

Nesta etapa, você cria um repositório de origem no CodeCatalyst. Esse repositório armazena os arquivos de origem do tutorial, como o arquivo de definição da tarefa.

Para ter mais informações sobre repositórios de origem, consulte [Criar um repositório de origem](#).

### Como criar um repositório de origem

- Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
- Navegue até o projeto, `codecatalyst-ecs-project`.
- No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
- Escolha Adicionar repositório e selecione Criar repositório.
- Em Nome do repositório, insira:

**`codecatalyst-ecs-source-repository`**

- Escolha Criar.

## Etapa 7: adicionar arquivos de origem

Nesta seção, você adiciona os arquivos de origem do Hello World ao seu CodeCatalyst repositório, `codecatalyst-ecs-source-repository`. Eles consistem em:

- Um arquivo `index.html` - Exibe uma mensagem do Hello World no navegador.
- Um Dockerfile: descreve a imagem básica a ser usada para sua imagem do Docker e os comandos do Docker a serem aplicados a ela.

- Um arquivo `taskdef.json` - Define a imagem do Docker a ser usada ao iniciar tarefas em seu cluster.

A estrutura de pastas é esta:

```
.
|- public-html
| |- index.html
|- Dockerfile
|- taskdef.json
```

#### Note

As instruções a seguir mostram como adicionar os arquivos usando o CodeCatalyst console, mas você pode usar o Git se preferir. Para obter detalhes, consulte [Clonar um repositório de origem](#).

## Tópicos

- [index.html](#)
- [Dockerfile](#)
- [taskdef.json](#)

## index.html

O arquivo `index.html` exibe uma mensagem do Hello World no navegador.

Como adicionar o arquivo `index.html`

1. No CodeCatalyst console, acesse seu repositório de origem, `codecatalyst-ecs-source-repository`.
2. Em Arquivos, selecione Criar arquivo.
3. Em Nome do arquivo, insira:

```
public-html/index.html
```

**⚠ Important**

Inclua o prefixo `public-html/` para criar uma pasta com o mesmo nome. Espera-se que `index.html` esteja nessa pasta.

4. Na caixa de texto, insira o código a seguir:

```
<html>
 <head>
 <title>Hello World</title>
 <style>
 body {
 background-color: black;
 text-align: center;
 color: white;
 font-family: Arial, Helvetica, sans-serif;
 }
 </style>
 </head>
 <body>
 <h1>Hello World</h1>
 </body>
</html>
```

5. Selecione Confirmar e, depois, escolha Confirmar novamente.

O `index.html` é adicionado ao seu repositório em uma pasta `public-html`.

## Dockerfile

O Dockerfile descreve a imagem base do Docker a ser usada e os comandos do Docker a serem aplicados a ela. Para ter mais informações sobre o Dockerfile, consulte a [Referência de Dockerfile](#).

O Dockerfile especificado aqui indica o uso da imagem base do Apache 2.4 (httpd). Também inclui instruções para copiar um arquivo fonte chamado `index.html` para uma pasta no servidor Apache que serve páginas da Web. A instrução `EXPOSE` no Dockerfile diz ao Docker que o contêiner está escutando na porta 80.

## Para adicionar o Dockerfile

1. No repositório de origem, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

**Dockerfile**

Não inclua uma extensão de arquivo.

### Important

O Dockerfile deve residir na pasta raiz do repositório. O comando `Docker build` do fluxo de trabalho espera que ele esteja lá.

3. Na caixa de texto, insira o código a seguir:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

O Dockerfile é adicionado ao repositório.

## taskdef.json

O arquivo `taskdef.json` que você adiciona nessa etapa é o mesmo que você já especificou em [Etapa 2: implantar uma aplicação de espaço reservado no Amazon ECS](#) com a seguinte diferença:

Em vez de especificar um nome de imagem do Docker codificado no campo `image:` (`httpd:2.4`), a definição da tarefa aqui usa algumas variáveis para indicar a imagem: `$REPOSITORY_URI` e `$IMAGE_TAG`. Essas variáveis serão substituídas por valores reais gerados pela ação de criação do fluxo de trabalho quando você executar o fluxo de trabalho em uma etapa posterior.

Para saber detalhes sobre os parâmetros de definição de tarefa, consulte [Parâmetros de definição de tarefa](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

## Como adicionar o arquivo taskdef.json

1. No repositório de origem, selecione Criar arquivo.



2. Em Nome do arquivo, insira:

```
taskdef.json
```

3. Na caixa de texto, insira o código a seguir:

```
{
 "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
execution-role",
 "containerDefinitions": [
 {
 "name": "codecatalyst-ecs-container",
 # The $REPOSITORY_URI and $IMAGE_TAG variables will be replaced
 # by the workflow at build time (see the build action in the
 # workflow)
 "image": $REPOSITORY_URI:$IMAGE_TAG,
 "essential": true,
 "portMappings": [
 {
 "hostPort": 80,
 "protocol": "tcp",
 "containerPort": 80
 }
]
 }
],
 "requiresCompatibilities": [
 "FARGATE"
],
 "networkMode": "awsvpc",
 "cpu": "256",
 "memory": "512",
 "family": "codecatalyst-ecs-task-def"
}
```

No código anterior, substitua

*arn:aws:iam::account\_ID:role/codecatalyst-ecs-task-execution-role*

pelo ARN do perfil de execução da tarefa que você anotou em [Como criar o perfil de execução de tarefas](#).

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

O arquivo `taskdef.json` é adicionado ao repositório.

## Etapa 8: criar e executar um fluxo de trabalho

Nesta etapa, você cria um fluxo de trabalho que pega seus arquivos de origem, os cria em uma imagem do Docker e, depois, implanta a imagem no seu cluster do Amazon ECS. Essa implantação substitui a aplicação de espaço reservado Apache existente.

O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de criação (BuildBackend) – No gatilho, a ação cria a imagem do Docker usando o Dockerfile e envia a imagem para o Amazon ECR. A ação de criação também atualiza o `taskdef.json` com o valor de campo `image` correto e, depois, cria um artefato de saída desse arquivo. Esse artefato é usado como entrada para a ação de implantação, que será a seguir.

Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).

- Uma ação de implantação (DeployToECS): ao concluir a ação de criação, a ação de implantação procura o artefato de saída gerado pela ação de criação (TaskDefArtifact), encontra o `taskdef.json` dentro dele e o registra no seu serviço do Amazon ECS. Em seguida, o serviço segue as instruções no arquivo `taskdef.json` para executar três tarefas do Amazon ECS – e os contêineres do Docker do Hello World associados – dentro do seu cluster do Amazon ECS.

Para criar um fluxo de trabalho

1. No CodeCatalyst console, no painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Selecione Criar fluxo de trabalho.
3. Em Repositório de origem, selecione `codecatalyst-ecs-source-repository`.
4. Em Ramificação, selecione `main`.
5. Escolha Criar.
6. Exclua o código de amostra YAML.

## 7. Adicione o seguinte código YAML:

### Note

No código YAML a seguir, você pode omitir as seções `Connections:` se quiser. Se você omitir essas seções, deverá garantir que o perfil especificado no campo `Perfil` do IAM padrão em seu ambiente inclua as permissões e as políticas de confiança dos dois perfis descritas em [Etapa 5: adicionar AWS funções a CodeCatalyst](#). Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#).

```
Name: codecatalyst-ecs-workflow
SchemaVersion: 1.0

Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 BuildBackend:
 Identifier: aws/build@v1
 Environment:
 Name: codecatalyst-ecs-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-ecs-build-role
 Inputs:
 Sources:
 - WorkflowSource
 Variables:
 - Name: REPOSITORY_URI
 Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo
 - Name: IMAGE_TAG
 Value: ${WorkflowSource.CommitId}
 Configuration:
 Steps:
 #pre_build:
 - Run: echo Logging in to Amazon ECR...
 - Run: aws --version
```

```

- Run: aws ecr get-login-password --region us-west-2 | docker login --
username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com
#build:
- Run: echo Build started on `date`
- Run: echo Building the Docker image...
- Run: docker build -t $REPOSITORY_URI:latest .
- Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
#post_build:
- Run: echo Build completed on `date`
- Run: echo Pushing the Docker images...
- Run: docker push $REPOSITORY_URI:latest
- Run: docker push $REPOSITORY_URI:$IMAGE_TAG
Replace the variables in taskdef.json
- Run: find taskdef.json -type f | xargs sed -i "s|\\$REPOSITORY_URI|
$REPOSITORY_URI|g"
- Run: find taskdef.json -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|
g"
- Run: cat taskdef.json
The output artifact will be a zip file that contains a task definition
file.
 Outputs:
 Artifacts:
 - Name: TaskDefArtifact
 Files:
 - taskdef.json
 DeployToECS:
 DependsOn:
 - BuildBackend
 Identifier: aws/ecs-deploy@v1
 Environment:
 Name: codecatalyst-ecs-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-ecs-deploy-role
 Inputs:
 Sources: []
 Artifacts:
 - TaskDefArtifact
 Configuration:
 region: us-west-2
 cluster: codecatalyst-ecs-cluster
 service: codecatalyst-ecs-service
 task-definition: taskdef.json

```

No código anterior, substitua:

- Ambas as instâncias de *codecatalyst-ecs-environment* com o nome do ambiente em que você criou [Pré-requisitos](#).
- Ambas as instâncias *codecatalyst-account-connection* com o nome de exibição da conexão da sua conta. O nome de exibição pode ser um número. Para obter mais informações, consulte [Etapa 5: adicionar AWS funções a CodeCatalyst](#).
- *codecatalyst-ecs-build-role* com o nome da função de construção em que você criou [Etapa 4: criar AWS funções](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo* (na Value: propriedade) com o URI do repositório Amazon ECR em que você criou. [Etapa 3: criar uma imagem de repositório do Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (no Run: aws ecr comando) com o URI do repositório Amazon ECR sem o sufixo da imagem (). /codecatalyst-ecs-image-repo
- *codecatalyst-ecs-deploy-role* com o nome da função de implantação na qual você criou [Etapa 4: criar AWS funções](#).
- Ambas as instâncias *us-west-2* com seu código de AWS região. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.

#### Note

Se você decidiu não criar e implantar funções, *codecatalyst-ecs-deploy-role* substitua *codecatalyst-ecs-build-role* e pelo nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função. Para obter mais informações sobre essa função, consulte [Etapa 4: criar AWS funções](#).

#### Tip

Em vez de usar os comandos find e sed mostrados no código do fluxo de trabalho anterior para atualizar o nome do repositório e da imagem, você pode usar a ação de definição de tarefas Renderizar Amazon ECS para essa finalidade. Para obter mais informações, consulte [Modificação de uma definição de tarefa do Amazon ECS](#).

8. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
9. Selecione Confirmar.
10. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
  - a. Em Mensagem de confirmação, remova o texto e digite:

**Add first workflow**

- b. Em Repositório, selecione `codecatalyst-ecs-source-repository`.
- c. Em Nome da ramificação, selecione a principal.
- d. Selecione Confirmar.

Agora você criou um fluxo de trabalho. A execução de um fluxo de trabalho é iniciada automaticamente devido ao gatilho definido na parte superior do fluxo de trabalho. Especificamente, quando você confirmou (e enviou) o arquivo `workflow.yaml` ao repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para visualizar o andamento da execução do fluxo de trabalho

1. No painel de navegação do CodeCatalyst console, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar, `codecatalyst-ecs-workflow`.
3. Escolha BuildBackendver o progresso da compilação.
4. Escolha o DeployToECS para ver o progresso da implantação.

Para ter mais informações sobre como visualizar os detalhes da execução, consulte o [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Como verificar a implantação

1. Abra o console clássico do Amazon ECS em <https://console.aws.amazon.com/ecs/>.
2. Escolha seu cluster, `codecatalyst-ecs-cluster`.
3. Escolha a guia Tasks (Tarefas).
4. Escolha qualquer uma das três tarefas.
5. No campo IP público, escolha endereço aberto.

Uma página “Hello World” aparece no navegador, indicando que o serviço Amazon ECS implantou a aplicação.

### Etapa 9: realizar uma alteração nos arquivos de origem

Nesta seção, você faz uma alteração no arquivo `index.html` no repositório de origem. Essa alteração faz com que o fluxo de trabalho crie uma nova imagem do Docker, a marque com um ID de confirmação, a envie por push ao Amazon ECR e a implante no Amazon ECS.

Para alterar o `index.html`

1. No CodeCatalyst console, no painel de navegação, escolha Código, escolha Repositórios de origem e, em seguida, escolha seu repositório. `codecatalyst-ecs-source-repository`
2. Escolha `public-html` e, depois, escolha `index.html`.

O conteúdo de `index.html` aparece.

3. Escolha Editar.
4. Na linha 14, altere o texto `Hello World` para `Tutorial complete!`.
5. Selecione Confirmar e, depois, escolha Confirmar novamente.

A confirmação faz com que uma nova execução do fluxo de trabalho seja iniciada.

6. (Opcional) Acesse a página principal do repositório de origem, selecione Exibir confirmações e anote o ID da confirmação da alteração em `index.html`.
7. Assista ao andamento da implantação:
  - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - b. Escolha `codecatalyst-ecs-workflow` para ver a última execução.
  - c. Escolha BuildBackende DeployToECS para ver o progresso da execução do fluxo de trabalho.
8. Verifique se a aplicação foi atualizada, da seguinte forma:
  - a. Abra o console clássico do Amazon ECS em <https://console.aws.amazon.com/ecs/>.
  - b. Escolha seu cluster, `codecatalyst-ecs-cluster`.
  - c. Escolha a guia Tasks (Tarefas).
  - d. Escolha qualquer uma das três tarefas.

- e. No campo IP público, escolha endereço aberto.

Uma página Tutorial `complete!` é exibida.

9. (Opcional) Em AWS, mude para o console do Amazon ECR e verifique se a nova imagem do Docker foi marcada com o ID de confirmação da etapa 6.

## Limpeza

Limpe os arquivos e serviços usados neste tutorial para evitar cobranças por eles.

No Console de gerenciamento da AWS, limpe nesta ordem:

1. No Amazon ECS, faça o seguinte:
  - a. Exclua `codecatalyst-ecs-service`.
  - b. Exclua `codecatalyst-ecs-cluster`.
  - c. Cancelar registro de `codecatalyst-ecs-task-definition`.
2. No Amazon ECR, exclua `codecatalyst-ecs-image-repo`.
3. Na Amazon EC2, exclua `codecatalyst-ecs-security-group`.
4. No IAM Identity Center, exclua:
  - a. `CodeCatalystECSUser`
  - b. `CodeCatalystECSPermissionSet`

No CodeCatalyst console, limpe da seguinte forma:

1. Exclua `codecatalyst-ecs-workflow`.
2. Exclua `codecatalyst-ecs-environment`.
3. Exclua `codecatalyst-ecs-source-repository`.
4. Exclua `codecatalyst-ecs-project`.

Neste tutorial, você aprendeu como implantar um aplicativo em um serviço do Amazon ECS usando um CodeCatalyst fluxo de trabalho e uma ação Deploy to Amazon ECS.

## Adição da ação “Implantar no Amazon ECS”

Use as instruções a seguir para adicionar a ação Implantar no Amazon ECS ao seu fluxo de trabalho.



## Visual

Para adicionar a ação “Implantar no Amazon ECS” usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Implantar no Amazon ECS e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Selecione Implantar no Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [YAML de ação “Implantar no Amazon ECS”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação “Implantar no Amazon ECS” usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Implantar no Amazon ECS e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Selecione Implantar no Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [YAML de ação “Implantar no Amazon ECS”](#).
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Variáveis de “Implantar no Amazon ECS”

A ação Implantar no Amazon ECS produz e define as seguintes variáveis em runtime. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave               | Valor                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cluster             | <p>O nome do cluster do Amazon ECS no qual foi implantado durante a execução do fluxo de trabalho.</p> <p>Exemplo: codecatalyst-ecs-cluster</p>                                                                                                                                             |
| deployment-platform | <p>O nome da plataforma de implantação.</p> <p>Codificado para AWS : ECS.</p>                                                                                                                                                                                                               |
| service             | <p>O nome do serviço do Amazon ECS no qual foi implantado durante a execução do fluxo de trabalho.</p> <p>Exemplo: codecatalyst-ecs-service</p>                                                                                                                                             |
| task-definition-arn | <p>O nome do recurso da Amazon (ARN) da definição de tarefa que foi registrada durante a execução do fluxo de trabalho.</p> <p>Exemplo: arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</p> <p>O :8 no exemplo anterior indica a revisão que foi registrada.</p> |
| deployment-url      | <p>Um link para a guia Eventos do console do Amazon ECS, onde você pode ver detalhes</p>                                                                                                                                                                                                    |

| Chave  | Valor                                                                                                                                                                                                                                     |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | da implantação do Amazon ECS associada à execução do fluxo de trabalho.<br><br>Exemplo: <code>https://console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/codecatalyst-ecs-cluster/services/codecatalyst-ecs-service/events</code> |
| region | O código da região em Região da AWS que foi implantado durante a execução do fluxo de trabalho.<br><br>Exemplo: <code>us-west-2</code>                                                                                                    |

## YAML de ação “Implantar no Amazon ECS”

Confira a seguir a definição de YAML da ação Implantar no Amazon ECS. Para saber como usar essa ação, consulte [Implantação no Amazon ECS com um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:
```

```
The action definition starts here.
ECSDeployAction_nn:
 Identifier: aws/ecs-deploy@v1
 DependsOn:
 - build-action
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - task-definition-artifact
 Configuration:
 region: us-east-1
 cluster: ecs-cluster
 service: ecs-service
 task-definition: task-definition-path
 force-new-deployment: false|true
 codedeploy-appspec: app-spec-file-path
 codedeploy-application: application-name
 codedeploy-deployment-group: deployment-group-name
 codedeploy-deployment-description: deployment-description
```

## ECSDeployAction

### (Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: ECSDeployAction\_nn.

Interface de usuário correspondente: guia Configuração/Nome de exibição da ação

## Identifier

(*ECSDeployAction*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/ecs-deploy@v1.

UI correspondente: diagrama de fluxo de ECSDeploy trabalho/Action\_NN/aws/ecs-deploy @v1 label

## DependsOn

(*ECSDeployAction*/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

## Compute

(*ECSDeployAction*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

## Type

(*ECSDeployAction*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: Configuração tab/Advanced - opcional/Tipo de computação

Fleet

(*ECSDeployAction*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

UI correspondente: configuração tab/Advanced - opcional/frota de computação

Timeout

(*ECSDeployAction*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

Environment

(*ECSDeployAction*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*ECSDeployAction*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*ECSDeployAction*/Environment/Connections)



(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em `Environment`.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Name

(*ECSDeployAction*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

## Role

(*ECSDeployAction*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome do perfil do IAM que a ação Implantar no Amazon ECS usa para acessar a AWS. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

### Warning

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Action": [
 "ecs:DescribeServices",
 "ecs:CreateTaskSet",
 "ecs>DeleteTaskSet",
 "ecs:ListClusters",
 "ecs:RegisterTaskDefinition",
 "ecs:UpdateServicePrimaryTaskSet",
 "ecs:UpdateService",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeListeners",
 "elasticloadbalancing:ModifyListener",
 "elasticloadbalancing:DescribeRules",
 "elasticloadbalancing:ModifyRule",
 "lambda:InvokeFunction",
 "lambda:ListFunctions",
```

```

 "cloudwatch:DescribeAlarms",
 "sns:Publish",
 "sns:ListTopics",
 "s3:GetObject",
 "s3:GetObjectVersion",
 "codedeploy:CreateApplication",
 "codedeploy:CreateDeployment",
 "codedeploy:CreateDeploymentGroup",
 "codedeploy:GetApplication",
 "codedeploy:GetDeployment",
 "codedeploy:GetDeploymentGroup",
 "codedeploy:ListApplications",
 "codedeploy:ListDeploymentGroups",
 "codedeploy:ListDeployments",
 "codedeploy:StopDeployment",
 "codedeploy:GetDeploymentTarget",
 "codedeploy:ListDeploymentTargets",
 "codedeploy:GetDeploymentConfig",
 "codedeploy:GetApplicationRevision",
 "codedeploy:RegisterApplicationRevision",
 "codedeploy:BatchGetApplicationRevisions",
 "codedeploy:BatchGetDeploymentGroups",
 "codedeploy:BatchGetDeployments",
 "codedeploy:BatchGetApplications",
 "codedeploy:ListApplicationRevisions",
 "codedeploy:ListDeploymentConfigs",
 "codedeploy:ContinueDeployment"
],
 "Resource": "*",
 "Effect": "Allow"
 }, {
 "Action": [
 "iam:PassRole"
],
 "Effect": "Allow",
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "iam:PassedToService": [
 "ecs-tasks.amazonaws.com",
 "codedeploy.amazonaws.com"
]
 }
 }
 }
]
}

```

**Note**

Na primeira vez em que o perfil for usado, use o caractere curinga a seguir na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A política de confiança personalizada a seguir:

**Note**

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' / Função Environment/account/role

**Inputs**

(*ECSDeployAction*/Inputs)

(Optional)

A seção Inputs define os dados que `ECSDeployAction` precisa durante a execução de um fluxo de trabalho.

 Note

Somente uma entrada (uma origem ou um artefato) é permitida por ação Implantar no Amazon ECS.

Interface de usuário correspondente: guia Entradas

Sources

(*ECSDeployAction*/Inputs/Sources)

(Obrigatório se o arquivo de definição de tarefas estiver armazenado em um repositório de origem)

Se seu arquivo de definição de tarefa estiver armazenado em um repositório de origem, especifique o rótulo do repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seu arquivo de definição de tarefa não estiver em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

Artifacts - input

(*ECSDeployAction*/Inputs/Artifacts)

(Obrigatório se o arquivo de definição de tarefa estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se o arquivo de definição de tarefa que você deseja implantar estiver em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se seu arquivo de definição de tarefa não estiver em um artefato, ele deverá residir em seu repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

## Configuration

(*ECSDeployAction*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

region

(Configuration/region)

(Obrigatório)

Especifique a AWS região em que seu cluster e serviço do Amazon ECS residem. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.

Interface de usuário correspondente: guia Configuração/Região

cluster

(*ECSDeployAction*/Configuration/cluster)

(Obrigatório)

Especifique o nome de um cluster existente do Amazon ECS. A ação Implantar no Amazon ECS implantará a aplicação em contêineres como uma tarefa nesse cluster. Para ter mais informações sobre clusters do Amazon ECS, consulte [Clusters](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Cluster

service

(*ECSDeployAction*/Configuration/service)

(Obrigatório)

Especifique o nome de um serviço do Amazon ECS existente que instanciará o arquivo de definição de tarefas. Esse serviço deve residir no cluster especificado no campo `cluster`. Para ter mais informações sobre serviços do Amazon ECS, consulte [Serviços do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Serviço

task-definition

(*ECSDeployAction*/Configuration/task-definition)

(Obrigatório)

Especifique o caminho para um arquivo de definição de tarefa existente. Se o arquivo residir em seu repositório de origem, o caminho é relativo à pasta raiz do repositório de origem. Se o arquivo residir em um artefato de uma ação anterior do fluxo de trabalho, o caminho é relativo à pasta raiz do artefato. Para ter mais informações sobre os arquivos de definição de tarefa, consulte [Definições de tarefa](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Definição de tarefa

force-new-deployment

(*ECSDeployAction*/Configuration/force-new-deployment)

(Obrigatório)

Se habilitado, o serviço do Amazon ECS poderá iniciar novas implantações sem alterações na definição do serviço. Forçar uma implantação faz com que o serviço interrompa todas as tarefas atualmente em execução e inicie novas tarefas. Para ter mais informações sobre como forçar novas implantações, consulte [Atualização de um serviço](#), no Guia do desenvolvedor do Amazon Elastic Container Service.

Padrão: false

Interface de usuário correspondente: guia Configuração/Forçar uma nova implantação do serviço

codedeploy-appspec

(*ECSDeployAction*/Configuration/codedeploy-appspec)

(Obrigatório se você configurou seu serviço Amazon ECS para usar blue/green implantações; caso contrário, omita)

Especifique o nome e o caminho para um arquivo de especificação de CodeDeploy aplicativo existente (AppSpec). Esse arquivo deve residir na raiz do seu repositório de CodeCatalyst origem. Para obter mais informações sobre AppSpec arquivos, consulte os [arquivos de especificação do CodeDeploy aplicativo \(AppSpec\)](#) no Guia AWS CodeDeploy do usuário.

**Note**

Forneça CodeDeploy informações somente se você tiver configurado seu serviço Amazon ECS para realizar blue/green implantações. Para implantações de atualizações contínuas (o padrão), CodeDeploy omite as informações. Para ter mais informações sobre implantações do Amazon ECS, consulte [Tipos de implantação do Amazon ECS](#), no Guia do Desenvolvedor do Amazon Elastic Container Service.

**Note**

Os CodeDeploy campos podem estar ocultos no editor visual. Para fazer com que apareçam, consulte [Por que os campos do CodeDeploy estão ausentes no editor visual?](#).

UI correspondente: guia Configuração/ CodeDeploy AppSpec

codedeploy-application

(*ECSDeployAction*/Configuration/codedeploy-application)

(Obrigatório se codedeploy-appspec for incluído)

Especifique o nome de um CodeDeploy aplicativo existente. Para obter mais informações sobre CodeDeploy aplicativos, consulte Como [trabalhar com aplicativos CodeDeploy no](#) Guia do AWS CodeDeploy usuário.

UI correspondente: guia de configuração/aplicativo CodeDeploy

codedeploy-deployment-group

(*ECSDeployAction*/Configuration/codedeploy-deployment-group)

(Obrigatório se codedeploy-appspec for incluído)

Especifique o nome de um grupo CodeDeploy de implantação existente. Para obter mais informações sobre grupos de CodeDeploy implantação, consulte [Trabalhando com grupos de implantação CodeDeploy no](#) Guia AWS CodeDeploy do Usuário.

UI correspondente: guia de configuração/grupo CodeDeploy de implantação



codedeploy-deployment-description

(*ECSDeployAction*/Configuration/codedeploy-deployment-description)

(Optional)

Especifique uma descrição da implantação que essa ação criará. Para obter mais informações, consulte Como [trabalhar com implantações CodeDeploy no](#) Guia do AWS CodeDeploy usuário.

UI correspondente: guia de configuração/descrição CodeDeploy da implantação

## Implantar no Amazon EKS com um fluxo de trabalho

### Tip

Para ver um tutorial que mostra como usar a ação Implantar no cluster do Kubernetes, consulte [Tutorial: Implantar uma aplicação no Amazon EKS](#).

Esta seção descreve como implantar um aplicativo em contêiner em um cluster Kubernetes usando um fluxo de trabalho. CodeCatalyst Para fazer isso, você deve adicionar a ação Implantar no cluster do Kubernetes ao seu fluxo de trabalho. Essa ação implanta a aplicação em um cluster do Kubernetes que você configurou no Amazon Elastic Kubernetes Service (EKS) usando um ou mais arquivos de manifesto do Kubernetes. Para um exemplo de manifesto, consulte [deployment.yaml](#) em [Tutorial: Implantar uma aplicação no Amazon EKS](#).

Para ter mais informações sobre o Kubernetes, consulte a [documentação do Kubernetes](#).

Para ter mais informações sobre o Amazon EKS, consulte [O que é o Amazon EKS?](#) no Guia do usuário do Amazon EKS.

### Tópicos

- [Como funciona a ação “Implantar no cluster do Kubernetes”](#)
- [Imagem de runtime usada pela ação “Implantar no Amazon EKS”](#)
- [Tutorial: Implantar uma aplicação no Amazon EKS](#)
- [Adição da ação “Implantar no cluster do Kubernetes”](#)
- [Variáveis de “Implantar no cluster do Kubernetes”](#)
- [YAML da ação “Implantar no cluster do Kubernetes”](#)

## Como funciona a ação “Implantar no cluster do Kubernetes”

A ação Implantar no cluster do Kubernetes funciona da seguinte forma:

1. Em tempo de execução, a ação instala o `kubectl` utilitário Kubernetes na máquina CodeCatalyst computacional em que a ação está sendo executada. A ação configura `kubectl` para apontar para o cluster do Amazon EKS que você forneceu ao configurar a ação. O utilitário `kubectl` é necessário para executar o comando `kubectl apply` a seguir.
2. A ação executa o `kubectl apply -f my-manifest.yaml` comando, que executa as instruções `my-manifest.yaml` para implantar seu aplicativo como um conjunto de contêineres e pods no cluster configurado. Para ter mais informações sobre esse comando, consulte o tópico [kubectl apply](#) na documentação de referência do Kubernetes.

## Imagem de runtime usada pela ação “Implantar no Amazon EKS”

A ação Implantar no Amazon EKS é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Tutorial: Implantar uma aplicação no Amazon EKS

Neste tutorial, você aprende a implantar um aplicativo em contêineres no Amazon Elastic Kubernetes Service usando CodeCatalyst um fluxo de trabalho da Amazon, o Amazon EKS e alguns outros serviços. AWS A aplicação implantada é um site simples “Hello, World!” criado em uma imagem do Docker do servidor da Web Apache. O tutorial mostra o trabalho de preparação necessário, como configurar uma máquina de desenvolvimento e um cluster do Amazon EKS, e depois descreve como criar um fluxo de trabalho para criar a aplicação e implantá-la no cluster.

Após a conclusão da implantação inicial, o tutorial instrui você a fazer uma alteração na origem da aplicação. Essa alteração faz com que uma nova imagem do Docker seja criada e enviada ao seu repositório de imagens do Docker com novas informações de revisão. A nova revisão da imagem do Docker é implantada no Amazon EKS.

### Tip

Em vez de seguir este tutorial, você pode usar um esquema que faz uma configuração completa do Amazon EKS para você. Você precisará usar o esquema Implantação de aplicações EKS. Para obter mais informações, consulte [Criar um projeto com um esquema](#).

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: configurar sua máquina de desenvolvimento](#)
- [Etapa 2: criar um cluster do Amazon EKS](#)
- [Etapa 3: criar uma imagem de repositório do Amazon ECR](#)
- [Etapa 4: adicionar arquivos de origem](#)
- [Etapa 5: criar AWS funções](#)
- [Etapa 6: Adicionar AWS funções a CodeCatalyst](#)
- [Etapa 7: atualizar o ConfigMap](#)
- [Etapa 8: criar e executar um fluxo de trabalho](#)
- [Etapa 9: realizar uma alteração nos arquivos de origem](#)
- [Limpeza](#)

## Pré-requisitos

Antes de começar este tutorial:

- Você precisa de um CodeCatalyst espaço na Amazon com uma AWS conta conectada. Para obter mais informações, consulte [Criar um espaço](#).
- Em seu espaço, você precisa de um projeto vazio chamado:

```
codecatalyst-eks-project
```

Use a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- Em seu projeto, você precisa de um repositório de CodeCatalyst código-fonte vazio chamado:

```
codecatalyst-eks-source-repository
```

Para obter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).

- Em seu projeto, você precisa de um ambiente de CodeCatalyst CI/CD (não um ambiente de desenvolvimento) chamado:

```
codecatalyst-eks-environment
```

Configure esse ambiente da seguinte forma:

- Escolha qualquer tipo, como Não produção.
- Conecte sua AWS conta a ela.
- Para o Perfil do IAM padrão, escolha qualquer perfil. Você especificará um perfil diferente posteriormente.

Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).

### Etapa 1: configurar sua máquina de desenvolvimento

A primeira etapa deste tutorial é configurar uma máquina de desenvolvimento com algumas ferramentas que você usará ao longo deste tutorial. Essas ferramentas são:

- o utilitário `eksctl` — para criação de clusters
- o utilitário `kubectl` — um pré-requisito para `eksctl`
- o AWS CLI — também um pré-requisito para `eksctl`

Você pode instalar essas ferramentas em sua máquina de desenvolvimento existente, se tiver uma, ou pode usar um ambiente de CodeCatalyst desenvolvimento, que é baseado em nuvem. A vantagem de um ambiente de CodeCatalyst desenvolvimento é que ele é fácil de ativar e desativar e está integrado a outros CodeCatalyst serviços, permitindo que você conclua este tutorial em menos etapas.

Este tutorial pressupõe que você usará um ambiente de CodeCatalyst desenvolvimento.

As instruções a seguir descrevem uma maneira rápida de iniciar um ambiente de CodeCatalyst desenvolvimento e configurá-lo com as ferramentas necessárias, mas se você quiser instruções detalhadas, consulte:

- [Criar um Ambiente de Desenvolvimento](#) neste guia.
- [Instalação do kubectl](#) no Guia do usuário do Amazon EKS.
- [Instalação ou atualização do eksctl](#) no Guia do usuário do Amazon EKS.
- [Instalação ou atualização da versão mais recente da AWS CLI](#) no Guia do usuário do AWS Command Line Interface .

## Como iniciar um novo ambiente de desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto, `codecatalyst-eks-project`.
3. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
4. Escolha o nome para o seu repositório de origem, `codecatalyst-eks-source-repository`.
5. Na parte superior, escolha Criar ambiente de desenvolvimento e AWS Cloud9 (no navegador).
6. Verifique se Trabalhar na ramificação existente e principal estão selecionados e, depois, selecione Criar.

O Ambiente de Desenvolvimento é iniciado em uma nova guia do navegador e seu repositório (`codecatalyst-eks-source-repository`) é clonado nela.

## Para instalar e configurar o kubectl

1. No terminal do Ambiente de Desenvolvimento, insira:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Insira:

```
chmod +x ./kubectl
```

3. Insira:

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

4. Insira:

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

5. Insira:

```
kubectl version --short --client
```

6. Confira se uma versão é exibida.

Você agora tem o `kubectl` instalado.

Para instalar e configurar o `eksctl`

 Note

`eksctl` não é estritamente necessário porque você pode usar o `kubectl` em vez disso. No entanto, o `eksctl` tem a vantagem de automatizar grande parte da configuração do cluster e, portanto, é a ferramenta recomendada para este tutorial.

1. No terminal do Ambiente de Desenvolvimento, insira:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

2. Insira:

```
sudo cp /tmp/eksctl /usr/bin
```

3. Insira:

```
eksctl version
```

4. Confira se uma versão é exibida.

Você agora tem o `eksctl` instalado.

Para verificar se o AWS CLI está instalado

1. No terminal do Ambiente de Desenvolvimento, insira:

```
aws --version
```

2. Verifique se uma versão aparece para verificar se o AWS CLI está instalado.

Conclua os procedimentos restantes para configurar o AWS CLI com as permissões necessárias para acessar AWS.

## Para configurar o AWS CLI

Você deve configurar o AWS CLI com chaves de acesso e um token de sessão para dar acesso aos AWS serviços. As instruções a seguir fornecem uma maneira rápida de configurar as chaves e o token, mas se você quiser instruções detalhadas, consulte [Configuração da AWS CLI](#) no Guia do usuário do AWS Command Line Interface .

1. Crie um usuário do IAM Identity Center, da seguinte forma:
  - a. Faça login no Console de gerenciamento da AWS e abra o Centro de Identidade do AWS IAM console em <https://console.aws.amazon.com/singlesignon/>.

(Talvez seja necessário selecionar Ativar se você nunca tiver feito login no IAM Identity Center antes.)

### Note

Certifique-se de fazer login usando o Conta da AWS que está conectado ao seu CodeCatalyst espaço. Você pode verificar qual conta está conectada navegando até seu espaço e escolhendo a guia Contas da AWS. Para obter mais informações, consulte [Criar um espaço](#).

- b. No painel de navegação, escolha Usuários e depois Adicionar usuário.
    - c. Em Nome de usuário, digite:

**codecatalyst-eks-user**

- d. Em Senha, selecione Gerar uma senha de uso único que você possa compartilhar com esse usuário.
      - e. Em Endereço de e-mail e Confirmar endereço de e-mail, insira um endereço de e-mail que ainda não exista no IAM Identity Center.
      - f. Em Nome, insira:

**codecatalyst-eks-user**

- g. Em Sobrenome, insira:

**codecatalyst-eks-user**

- h. Em Nome de exibição, mantenha:

**codecatalyst-eks-user codecatalyst-eks-user**

- i. Escolha Próximo.
- j. Na página Adicionar usuário a grupos, selecione Avançar.
- k. Na página Revisar e adicionar usuário, revise as informações e selecione Adicionar usuário.

Uma caixa de diálogo Senha de uso único é exibida.

- l. Selecione Copiar e cole as informações de login em um arquivo de texto. As informações de login consistem na URL do portal de AWS acesso, um nome de usuário e uma senha de uso único.
- m. Escolha Fechar.

2. Crie um conjunto de permissões, da seguinte forma:

- a. No painel de navegação, escolha Conjuntos de permissões e, depois, escolha Criar conjunto de permissões.
- b. Escolha Conjunto de permissões predefinido e, em seguida, selecione AdministratorAccess. Esta política fornece permissões totais para todos os Serviços da AWS.
- c. Escolha Próximo.
- d. Em Nome do conjunto de permissões, remova AdministratorAccess e digite:

**codecatalyst-eks-permission-set**

- e. Escolha Próximo.
- f. Na página Revisar e criar, revise as informações e selecione Criar.
3. Atribua o conjunto de permissões para codecatalyst-eks-user, da seguinte forma:

- a. No painel de navegação Contas da AWS, escolha e marque a caixa de seleção ao lado da caixa de seleção na Conta da AWS qual você está conectado no momento.
- b. Escolha Atribuir usuários ou grupos.
- c. Escolha a guia Users.
- d. Marque a caixa de seleção ao lado dos logs do codecatalyst-eks-user.
- e. Escolha Próximo.

- f. Marque a caixa de seleção ao lado dos logs do codecatalyst-eks-permission-set



- g. Escolha Próximo.
- h. Revise suas informações e selecione Enviar.

Agora você designou `codecatalyst-eks-user` e `codecatalyst-eks-permission-set` para o seu Conta da AWS, unindo-os.

- 4. Obtenha as chaves de acesso de `codecatalyst-eks-user` e o token de sessão, da seguinte forma:

- a. Verifique se você tem a URL do portal de AWS acesso, o nome de usuário e a senha de uso único para `codecatalyst-eks-user`. Você deveria ter copiado essas informações para um editor de texto anteriormente.

 Note

Se você não tiver essas informações, acesse a página de detalhes do `codecatalyst-eks-user` no IAM Identity Center, selecione Redefinir senha, Gerar uma senha de uso único [...] e Redefinir senha novamente para exibir as informações na tela.

- b. Sair de AWS.
- c. Cole o URL do portal de AWS acesso na barra de endereço do seu navegador.
- d. Faça login com:

- Nome de usuário:

`codecatalyst-eks-user`

- Senha:

*one-time-password*

- e. Em Definir nova senha, insira uma nova senha e selecione Definir nova senha.

Uma caixa de Conta da AWS aparece na tela.

- f. Selecione Conta da AWS e, depois, escolha o nome da Conta da AWS a que você atribuiu o usuário `codecatalyst-eks-user` e o conjunto de permissões.
- g. Ao lado de `codecatalyst-eks-permission-set`, selecione Linha de comando ou acesso programático.

- h. Copie os comandos no meio da página. Eles são semelhantes a:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
export AWS_SESSION_TOKEN="session-token"
```

... onde *session-token* está uma longa sequência aleatória.

5. Adicione as chaves de acesso e o token de sessão ao AWS CLI, da seguinte forma:
- Retorne ao seu ambiente de CodeCatalyst desenvolvimento.
  - No prompt do terminal, cole os comandos que você copiou. Pressione Enter.

Agora você configurou o AWS CLI com chaves de acesso e um token de sessão. Agora você pode usar AWS CLI para concluir as tarefas exigidas por este tutorial.

 Important

Se, em algum momento durante este tutorial, você vir mensagens semelhantes a: `Unable to locate credentials. You can configure credentials by running "aws configure"`.

Ou:

`ExpiredToken: The security token included in the request is expired`

... é porque sua AWS CLI sessão expirou. Nesse caso, não execute o comando `aws configure`. Em vez disso, use as instruções na etapa 4 desse procedimento, que começa com `Obtain codecatalyst-eks-user's access key and session token`, para atualizar sua sessão.

## Etapa 2: criar um cluster do Amazon EKS

Nesta seção, você cria um cluster no Amazon EKS. As instruções abaixo descrevem uma maneira rápida de criar o cluster usando o `eksctl`, mas se você quiser instruções detalhadas, consulte:

- [Conceitos básicos do eksctl](#) no Guia do usuário do Amazon EKS

or

- [Introdução ao console e à AWS CLI](#) no Guia do usuário do Amazon EKS (este tópico fornece instruções do `kubectl` para criar o cluster)

 Note

Os [clusters privados](#) não são compatíveis com a CodeCatalyst integração com o Amazon EKS.

## Antes de começar

Verifique se você concluiu as seguintes tarefas na sua máquina de desenvolvimento:

- Instalou o utilitário `eksctl`.
- Instalou o utilitário `kubectl`.
- Instalei AWS CLI e configurei com chaves de acesso e um token de sessão.

Para ter informações sobre como concluir essas tarefas, consulte [Etapa 1: configurar sua máquina de desenvolvimento](#).

## Para criar um cluster

 Important

Não use a interface de usuário do serviço Amazon EKS para criar o cluster porque o cluster não será configurado corretamente. Use o utilitário `eksctl`, conforme descrito nas etapas a seguir.

1. Acesse o Ambiente de Desenvolvimento.
2. Crie um cluster e nós:

```
eksctl create cluster --name codecatalyst-eks-cluster --region us-west-2
```

Em que:

- *codecatalyst-eks-cluster* é substituído pelo nome que você deseja dar ao seu cluster.

- *us-west-2* é substituído pela sua região.

Depois de 10 a 20 minutos, uma mensagem semelhante à seguinte é exibida:

EKS cluster "codecatalyst-eks-cluster" in "us-west-2" region is ready

 Note

Você verá várias mensagens `waiting for CloudFormation stack` enquanto a AWS cria seu cluster. Isso é esperado.

3. Verifique se o cluster foi criado:

```
kubectl cluster-info
```

Você verá uma mensagem semelhante à seguinte, indicando uma criação de cluster bem-sucedida:

```
Kubernetes master is running at https://long-string.gr7.us-west-2.eks.amazonaws.com
CoreDNS is running at https://long-string.gr7.us-west-2.eks.amazonaws.com/api/v1/
namespaces/kube-system/services/kube-dns:dns/proxy
```

### Etapa 3: criar uma imagem de repositório do Amazon ECR

Nesta seção, você cria um repositório de imagens privadas no Amazon Elastic Container Registry (Amazon ECR). Esse repositório armazena a imagem do Docker do tutorial.

Para obter mais informações sobre o Amazon ECR, consulte o Guia do usuário do Amazon Elastic Container Registry.

Para criar um repositório de imagens no Amazon ECR

1. Acesse o Ambiente de Desenvolvimento.
2. Crie um repositório vazio no Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-eks-image-repo
```

*codecatalyst-eks-image-repo* Substitua pelo nome que você deseja dar ao repositório Amazon ECR.

Este tutorial pressupõe que você tem um repositório chamado `codecatalyst-eks-image-repo`.

3. Exiba os detalhes do repositório do Amazon ECR:

```
aws ecr describe-repositories \
 --repository-names codecatalyst-eks-image-repo
```

4. Observe o valor “`repositoryUri`”: , por exemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo`.

Você precisará dele mais tarde ao adicionar o repositório ao seu fluxo de trabalho.

#### Etapa 4: adicionar arquivos de origem

Nesta seção, você adiciona arquivos de origem da aplicação ao seu repositório de origem (`codecatalyst-eks-source-repository`). Eles consistem em:

- Um arquivo `index.html`: exibe uma mensagem “Hello, World!” no navegador.
- Um `Dockerfile`: descreve a imagem básica a ser usada para sua imagem do Docker e os comandos do Docker a serem aplicados a ela.
- Um arquivo `deployment.yaml`: o manifesto do Kubernetes que define o serviço e a implantação do Kubernetes.

A estrutura de pastas é esta:

```
|– codecatalyst-eks-source-repository
 |– Kubernetes
 |– deployment.yaml
 |– public-html
 | |– index.html
 |– Dockerfile
```

#### Tópicos

- [index.html](#)

- [Dockerfile](#)
- [deployment.yaml](#)

index.html

O arquivo `index.html` exibe uma mensagem “Hello, World!” no navegador.

Como adicionar o arquivo `index.html`

1. Acesse o Ambiente de Desenvolvimento.
2. Em `codecatalyst-eks-source-repository`, crie uma pasta chamada `public-html`.
3. Em `/public-html`, crie um arquivo chamado `index.html` com o conteúdo a seguir:

```
<html>
 <head>
 <title>Hello World</title>
 <style>
 body {
 background-color: black;
 text-align: center;
 color: white;
 font-family: Arial, Helvetica, sans-serif;
 }
 </style>
 </head>
 <body>
 <h1>Hello, World!</h1>
 </body>
</html>
```

4. No prompt do terminal, insira:

```
cd /projects/codecatalyst-eks-source-repository
```

5. Adicione, confirme e envie:

```
git add .
git commit -m "add public-html/index.html"
git push
```

O `index.html` é adicionado ao seu repositório em uma pasta `public-html`.

## Dockerfile

O Dockerfile descreve a imagem base do Docker a ser usada e os comandos do Docker a serem aplicados a ela. Para ter mais informações sobre o Dockerfile, consulte a [Referência de Dockerfile](#).

O Dockerfile especificado aqui indica o uso da imagem base do Apache 2.4 (`httpd`). Também inclui instruções para copiar um arquivo fonte chamado `index.html` para uma pasta no servidor Apache que serve páginas da Web. A instrução `EXPOSE` no Dockerfile diz ao Docker que o contêiner está escutando na porta 80.

Para adicionar o Dockerfile

1. Em `codecatalyst-eks-source-repository`, crie um arquivo chamado `Dockerfile` com o conteúdo a seguir:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

Não inclua uma extensão de arquivo.

### Important

O Dockerfile deve residir na pasta raiz do repositório. O comando `Docker build` do fluxo de trabalho espera que ele esteja lá.

2. Adicione, confirme e envie:

```
git add .
git commit -m "add Dockerfile"
git push
```

O Dockerfile é adicionado ao repositório.

## deployment.yaml

Nesta seção, você adiciona um arquivo `deployment.yaml` ao seu repositório. O arquivo `deployment.yaml` é um manifesto do Kubernetes que define dois tipos de recursos do Kubernetes a serem executados: um “serviço” e uma “implantação”.

- O 'serviço' implanta um balanceador de carga na Amazon. EC2 O balanceador de carga fornece um URL público voltado para a Internet e uma porta padrão (porta 80) que você pode usar para navegar até a aplicação “Hello, World!” aplicativo.
- A “implantação” implanta três pods, e cada pod terá um contêiner do Docker com a aplicação “Hello, World!” aplicativo. Os três pods são implantados nos nós que foram criados quando você criou o cluster.

O manifesto neste tutorial é curto. No entanto, um manifesto pode incluir vários tipos de recursos do Kubernetes, como pods, trabalhos, entradas e políticas de rede. Além disso, você pode usar vários arquivos de manifesto se sua implantação for complexa.

Para adicionar um arquivo `deployment.yaml`

1. Em `codecatalyst-eks-source-repository`, crie uma pasta chamada Kubernetes.
2. Em `/Kubernetes`, crie um arquivo chamado `deployment.yaml` com o conteúdo a seguir:

```
apiVersion: v1
kind: Service
metadata:
 name: my-service
 labels:
 app: my-app
spec:
 type: LoadBalancer
 selector:
 app: my-app
 ports:
 - protocol: TCP
 port: 80
 targetPort: 80

apiVersion: apps/v1
kind: Deployment
metadata:
```



```
name: my-deployment
labels:
 app: my-app
spec:
 replicas: 3
 selector:
 matchLabels:
 app: my-app
 template:
 metadata:
 labels:
 app: my-app
 spec:
 containers:
 - name: codecatalyst-eks-container
 # The $REPOSITORY_URI and $IMAGE_TAG placeholders will be replaced by
 actual values supplied by the build action in your workflow
 image: $REPOSITORY_URI:$IMAGE_TAG
 ports:
 - containerPort: 80
```

### 3. Adicione, confirme e envie:

```
git add .
git commit -m "add Kubernetes/deployment.yaml"
git push
```

O arquivo `deployment.yaml` é adicionado ao seu repositório em uma pasta chamada `Kubernetes`.

Agora você adicionou todos os seus arquivos de origem.

Reserve um momento para verificar seu trabalho e garantir que colocou todos os arquivos nas pastas corretas. A estrutura de pastas é esta:

```
|— codecatalyst-eks-source-repository
 |— Kubernetes
 |— deployment.yaml
 |— public-html
 | |— index.html
 |— Dockerfile
```

## Etapa 5: criar AWS funções

Nesta seção, você cria funções AWS do IAM que seu CodeCatalyst fluxo de trabalho precisará para funcionar. Esses perfis são:

- Criar função — Concede permissão à ação de CodeCatalyst criação (no fluxo de trabalho) para acessar sua AWS conta e gravar no Amazon ECR e na Amazon EC2.
- Função de implantação — concede à ação de cluster CodeCatalyst Deploy to Kubernetes (no fluxo de trabalho) permissão para acessar sua conta AWS e o Amazon EKS.

Para ter informações sobre perfis do IAM, consulte [Perfis do IAM](#) no Guia do usuário do AWS Identity and Access Management .

### Note

Para economizar tempo, você pode criar um único perfil, chamado `CodeCatalystWorkflowDevelopmentRole-spaceName`, em vez dos dois perfis listados anteriormente. Para obter mais informações, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões bem amplas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando os dois perfis listados anteriormente.

Para criar os perfis de criação e implantação, conclua a série de procedimentos a seguir.

### 1. Para criar uma política de confiança para os dois perfis

1. Acesse o Ambiente de Desenvolvimento.
2. No diretório Cloud9-*long-string*, crie um arquivo chamado `codecatalyst-eks-trust-policy.json` com o seguinte conteúdo:

### 2. Para criar a política de criação do perfil de criação

- No diretório Cloud9-*long-string*, crie um arquivo chamado `codecatalyst-eks-build-policy.json` com o seguinte conteúdo:

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecr:*",
 "ec2:*"
],
 "Resource": "*"
 }
]
}
```

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

### 3. Para criar a política do perfil de implantação

- No diretório Cloud9-*long-string*, crie um arquivo chamado codecatalyst-eks-deploy-policy.json com o seguinte conteúdo:

## JSON

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

```

 "Action": [
 "eks:DescribeCluster",
 "eks:ListClusters"
],
 "Resource": "*"
 }
}

```

#### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

Agora você adicionou três documentos de política ao seu Ambiente de Desenvolvimento. A estrutura do seu diretório agora deve ficar assim:

```

|- Cloud9-long-string
 |- .c9
 |- codecatalyst-eks-source-repository
 |- Kubernetes
 |- public-html
 |- Dockerfile
 codecatalyst-eks-build-policy.json
 codecatalyst-eks-deploy-policy.json
 codecatalyst-eks-trust-policy.json

```

## 4. Para adicionar a política de compilação ao AWS

1. No terminal do Ambiente de Desenvolvimento, insira:

```
cd /projects
```

2. Insira:

```
aws iam create-policy \
 --policy-name codecatalyst-eks-build-policy \
 --policy-document file://codecatalyst-eks-build-policy.json
```

3. Pressione Enter.
4. Na saída do comando, observe o valor "arn":, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Esse ARN será necessário posteriormente.

## 5. Para adicionar a política de implantação ao AWS

1. Insira:

```
aws iam create-policy \
 --policy-name codecatalyst-eks-deploy-policy \
 --policy-document file://codecatalyst-eks-deploy-policy.json
```

2. Pressione Enter.
3. Na saída do comando, observe o valor "arn": da política de implantação, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy`. Esse ARN será necessário posteriormente.

## 6. Para criar o perfil de criação

1. Insira:

```
aws iam create-role \
 --role-name codecatalyst-eks-build-role \
 --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Pressione Enter.
3. Insira:

```
aws iam attach-role-policy \
 --role-name codecatalyst-eks-build-role \
 --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

Onde *arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy* é substituído pelo ARN da política de construção que você observou anteriormente.

4. Pressione Enter.
5. No prompt do terminal, insira:

```
aws iam get-role \
 --role-name codecatalyst-eks-build-role
```

6. Pressione Enter.
7. Observe o valor "Arn": do perfil, por exemplo, *arn:aws:iam::111122223333:role/codecatalyst-eks-build-role*. Esse ARN será necessário posteriormente.

## 7. Para criar o perfil de implantação

1. Insira:

```
aws iam create-role \
 --role-name codecatalyst-eks-deploy-role \
 --assume-role-policy-document file:///codecatalyst-eks-trust-policy.json
```

2. Pressione Enter.
3. Insira:

```
aws iam attach-role-policy \
 --role-name codecatalyst-eks-deploy-role \
 --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy
```

Onde *arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy* é substituído pelo ARN da política de implantação que você anotou anteriormente.

4. Pressione Enter.
5. Insira:

```
aws iam get-role \
 --role-name codecatalyst-eks-deploy-role
```

6. Pressione Enter.

7. Observe o valor "Arn": do perfil, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Esse ARN será necessário posteriormente.

Agora você criou funções de criação e implantação e anotou suas ARNs.

#### Etapa 6: Adicionar AWS funções a CodeCatalyst

Nesta etapa, você adiciona a função de criação (`codecatalyst-eks-build-role`) e a função de implantação (`codecatalyst-eks-deploy-role`) à Conta da AWS função conectada ao seu espaço. Isso torna os perfis disponíveis para uso em seu fluxo de trabalho.

Para adicionar funções de criação e implantação ao seu Conta da AWS

1. No CodeCatalyst console, navegue até seu espaço.
2. Na parte superior, selecione Configurações.
3. No painel de navegação, selecione Contas da AWS . Uma lista de contas é exibida.
4. Na coluna Nome de CodeCatalyst exibição da Amazon, copie o nome de exibição de Conta da AWS onde você criou suas funções de criação e implantação. (Pode ser um número.) Esse valor será necessário posteriormente, ao criar seu fluxo de trabalho.
5. Escolha o nome de exibição.
6. Escolha Gerenciar funções no console de AWS gerenciamento.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é exibida. Talvez seja necessário fazer login para acessá-la.

7. Selecione Adicionar um perfil existente que você criou no IAM.

Uma lista suspensa aparece. A lista exibe os perfis de criação e implantação e outros perfis do IAM com uma política de confiança que inclui as entidades principais de serviço `codecatalyst-runner.amazonaws.com` e `codecatalyst.amazonaws.com`.

8. Na lista suspensa, adicione:
  - `codecatalyst-eks-build-role`
  - `codecatalyst-eks-deploy-role`

 Note

Se você vir `The security token included in the request is invalid`, pode ser porque você não tem as permissões corretas. Para corrigir esse problema, saia e faça login novamente com a AWS conta que você usou ao criar seu CodeCatalyst espaço. AWS

9. Volte ao CodeCatalyst console e atualize a página.

Os perfis de criação e implantação agora devem aparecer em Perfis do IAM.

Essas funções agora estão disponíveis para uso em CodeCatalyst fluxos de trabalho.

### Etapa 7: atualizar o ConfigMap

Você deve adicionar o perfil de implantação que você criou em [Etapa 5: criar AWS funções](#) ao arquivo ConfigMap do Kubernetes para dar à ação Implantar no cluster do Kubernetes (no fluxo de trabalho) a capacidade de acessar e interagir com seu cluster. Você pode usar o `eksctl` ou `kubectl` para executar essa tarefa.

Para configurar o arquivo Kubernetes ConfigMap usando `eksctl`

- No terminal do Ambiente de Desenvolvimento, insira:

```
eksctl create iamidentitymapping --cluster codecatalyst-eks-cluster --
arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role --group
system:masters --username codecatalyst-eks-deploy-role --region us-west-2
```

Em que:

- codecatalyst-eks-cluster* é substituído pelo nome do cluster Amazon EKS.
- arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role* é substituído pelo ARN da função de implantação que você criou em [Etapa 5: criar AWS funções](#)
- codecatalyst-eks-deploy-role* (ao lado de `--username`) é substituído pelo nome da função de implantação que você criou em [Etapa 5: criar AWS funções](#).



 Note

Se você decidiu não criar uma função de implantação, *codecatalyst-eks-deploy-role* substitua pelo nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função. Para obter mais informações sobre essa função, consulte [Etapa 5: criar AWS funções](#).

- *us-west-2* é substituído pela sua região.

Para saber detalhes sobre esse comando, consulte [Gerenciar usuários e perfis do IAM](#).

Uma mensagem semelhante à seguinte é exibida:

```
2023-06-09 00:58:29 [#] checking arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role against entries in the auth ConfigMap
2023-06-09 00:58:29 [#] adding identity "arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role" to auth ConfigMap
```

Para configurar o arquivo Kubernetes ConfigMap usando kubectl

1. No terminal do Ambiente de Desenvolvimento, insira:

```
kubectl edit configmap -n kube-system aws-auth
```

O ConfigMap arquivo aparece na tela.

2. Adicione o texto em itálico vermelho:

```
Please edit the object below. Lines beginning with a '#' will be ignored,
and an empty file will abort the edit. If an error occurs while saving this file
will be
reopened with the relevant failures.
#
apiVersion: v1
data:
 mapRoles: |
 - groups:
 - system:bootstrappers
 - system:nodes
```

```

 rolearn: arn:aws:iam::111122223333:role/eksctl-codecatalyst-eks-cluster-n-
NodeInstanceRole-16BC456ME6YR5
 username: system:node:{{EC2PrivateDNSName}}
 - groups:
 - system:masters
 rolearn: arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role
 username: codecatalyst-eks-deploy-role
 mapUsers: |
 []
kind: ConfigMap
metadata:
 creationTimestamp: "2023-06-08T19:04:39Z"
 managedFields:
 ...

```

Em que:

- *arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role* é substituído pelo ARN da função de implantação que você criou em [Etapa 5: criar AWS funções](#)
- *codecatalyst-eks-deploy-role* (ao lado de `username:`) é substituído pelo nome da função de implantação que você criou em [Etapa 5: criar AWS funções](#).

#### Note

Se você decidiu não criar uma função de implantação, *codecatalyst-eks-deploy-role* substitua pelo nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função. Para obter mais informações sobre essa função, consulte [Etapa 5: criar AWS funções](#).

Para saber detalhes, consulte [Habilitar o acesso da entidade principal do IAM ao seu cluster](#) no Guia do usuário do Amazon EKS.

Agora você concedeu ao perfil de implantação e, por extensão, à ação Implantar no Amazon EKS, permissões `system:masters` para seu cluster do Kubernetes.

## Etapa 8: criar e executar um fluxo de trabalho

Nesta etapa, você cria um fluxo de trabalho que cria os arquivos de origem em uma imagem do Docker e, depois, implanta a imagem em três pods no cluster do Amazon EKS.

O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de criação (BuildBackend) – No gatilho, a ação cria a imagem do Docker usando o Dockerfile e envia a imagem para o Amazon ECR. A ação de criação também atualiza as variáveis \$REPOSITORY\_URI e \$IMAGE\_TAG no arquivo deployment.yaml com os valores corretos e, depois, cria um artefato de saída desse arquivo e de qualquer outro na pasta Kubernetes. Neste tutorial, o único arquivo na pasta Kubernetes é deployment.yaml, mas você pode incluir mais arquivos. O artefato é usado como entrada para a ação de implantação, que será a seguir.

Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).

- Uma ação de implantação (DeployToEKS) – Ao concluir a ação de criação, a ação de implantação procura o artefato de saída gerado pela ação de criação (Manifests), encontra o deployment.yaml dentro dele. Em seguida, a ação segue as instruções no arquivo deployment.yaml para executar três pods, cada um contendo um único “Hello, World!” Contêiner do Docker, dentro do cluster do Amazon EKS.

Para criar um fluxo de trabalho

1. Vá para o CodeCatalyst console.
2. Navegue até o projeto (codecatalyst-eks-project).
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione Criar fluxo de trabalho.
5. Em Repositório de origem, selecione codecatalyst-eks-source-repository.
6. Em Ramificação, selecione main.
7. Escolha Criar.
8. Exclua o código de amostra YAML.
9. Adicione o código YAML a seguir para criar um arquivo de definição de fluxo de trabalho:

**Note**

Para ter mais informações sobre o arquivo de definição de fluxo de trabalho, consulte [Definição do YAML do fluxo de trabalho](#).

**Note**

No código YAML a seguir, você pode omitir as seções `Connections:` se quiser. Se você omitir essas seções, deverá garantir que o perfil especificado no campo Perfil do IAM padrão em seu ambiente inclua as permissões e as políticas de confiança dos dois perfis descritas em [Etapa 6: Adicionar AWS funções a CodeCatalyst](#). Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#).

```
Name: codecatalyst-eks-workflow
```

```
SchemaVersion: 1.0
```

```
Triggers:
```

```
- Type: PUSH
```

```
 Branches:
```

```
 - main
```

```
Actions:
```

```
 BuildBackend:
```

```
 Identifier: aws/build@v1
```

```
 Environment:
```

```
 Name: codecatalyst-eks-environment
```

```
 Connections:
```

```
 - Name: codecatalyst-account-connection
```

```
 Role: codecatalyst-eks-build-role
```

```
 Inputs:
```

```
 Sources:
```

```
 - WorkflowSource
```

```
 Variables:
```

```
 - Name: REPOSITORY_URI
```

```
 Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-
image-repo
```

```
 - Name: IMAGE_TAG
```

```

 Value: ${WorkflowSource.CommitId}
Configuration:
Steps:
 #pre_build:
 - Run: echo Logging in to Amazon ECR...
 - Run: aws --version
 - Run: aws ecr get-login-password --region us-west-2 | docker login --
username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com
 #build:
 - Run: echo Build started on `date`
 - Run: echo Building the Docker image...
 - Run: docker build -t $REPOSITORY_URI:latest .
 - Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
 #post_build:
 - Run: echo Build completed on `date`
 - Run: echo Pushing the Docker images...
 - Run: docker push $REPOSITORY_URI:latest
 - Run: docker push $REPOSITORY_URI:$IMAGE_TAG
 # Replace the variables in deployment.yaml
 - Run: find Kubernetes/ -type f | xargs sed -i "s|\\$REPOSITORY_URI|
$REPOSITORY_URI|g"
 - Run: find Kubernetes/ -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|g"
 - Run: cat Kubernetes/*
 # The output artifact will be a zip file that contains Kubernetes manifest
files.
Outputs:
 Artifacts:
 - Name: Manifests
 Files:
 - "Kubernetes/*"
DeployToEKS:
 DependsOn:
 - BuildBackend
 Identifier: aws/kubernetes-deploy@v1
 Environment:
 Name: codecatalyst-eks-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-eks-deploy-role
 Inputs:
 Artifacts:
 - Manifests
 Configuration:
 Namespace: default

```

```
Region: us-west-2
Cluster: codecatalyst-eks-cluster
Manifests: Kubernetes/
```

No código anterior, substitua:

- Ambas as instâncias de *codecatalyst-eks-environment* com o nome do ambiente em que você criou [Pré-requisitos](#).
- Ambas as instâncias *codecatalyst-account-connection* com o nome de exibição da conexão da sua conta. O nome de exibição pode ser um número. Para obter mais informações, consulte [Etapa 6: Adicionar AWS funções a CodeCatalyst](#).
- *codecatalyst-eks-build-role* com o nome da função de construção em que você criou [Etapa 5: criar AWS funções](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo* (na Value: propriedade) com o URI do repositório Amazon ECR em que você criou. [Etapa 3: criar uma imagem de repositório do Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (no Run: aws ecr comando) com o URI do repositório Amazon ECR sem o sufixo da imagem (). */codecatalyst-eks-image-repo*
- *codecatalyst-eks-deploy-role* com o nome da função de implantação na qual você criou [Etapa 5: criar AWS funções](#).
- Ambas as instâncias *us-west-2* com seu código de AWS região. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.

 Note

Se você decidiu não criar e implantar funções, *codecatalyst-eks-deploy-role* substitua *codecatalyst-eks-build-role* e pelo nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função. Para obter mais informações sobre essa função, consulte [Etapa 5: criar AWS funções](#).

10. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
11. Selecione Confirmar.
12. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
  - a. Em Mensagem de confirmação, remova o texto e digite:

**Add first workflow**

- b. Em Repositório, selecione `codecatalyst-eks-source-repository`.
- c. Em Nome da ramificação, selecione a principal.
- d. Selecione Confirmar.

Agora você criou um fluxo de trabalho. A execução de um fluxo de trabalho é iniciada automaticamente devido ao gatilho definido na parte superior do fluxo de trabalho. Especificamente, quando você confirmou (e enviou) o arquivo `workflow.yaml` ao repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para visualizar o andamento da execução do fluxo de trabalho

1. No painel de navegação do CodeCatalyst console, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar, `codecatalyst-eks-workflow`.
3. Escolha BuildBackendver o progresso da construção.
4. Escolha DeployToEKS para ver o progresso da implantação.

Para ter mais informações sobre como visualizar os detalhes da execução, consulte o [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Como verificar a implantação

1. Abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
2. À esquerda, na parte inferior, escolha Balanceadores de carga.
3. Selecione o balanceador de carga que foi criado como parte da sua implantação do Kubernetes. Se você não tiver certeza de qual balanceador de carga escolher, procure as seguintes tags na guia Tags:
  - `kubernetes.io/service-name`
  - `kubernetes.io/cluster/ekstutorialcluster`
4. Com o balanceador de carga correto selecionado, escolha a guia Descrição.
5. Copie e cole o valor do Nome do DNS na barra de endereço do navegador.

A página da Web “Hello, World!” aparece no navegador, indicando que você implantou a aplicação com êxito.

### Etapa 9: realizar uma alteração nos arquivos de origem

Nesta seção, você faz uma alteração no arquivo `index.html` no repositório de origem. Essa alteração faz com que o fluxo de trabalho crie uma nova imagem do Docker, a marque com um ID de confirmação, a envie por push ao Amazon ECR e a implante no Amazon ECS.

Para alterar o `index.html`

1. Acesse o Ambiente de Desenvolvimento.
2. No prompt do terminal, mude para seu repositório de origem:

```
cd /projects/codecatalyst-eks-source-repository
```

3. Obtenha as alterações mais recentes do fluxo de trabalho:

```
git pull
```

4. Abra o `codecatalyst-eks-source-repository/public-html/index.html`.
5. Na linha 14, altere o texto `Hello, World!` para `Tutorial complete!`.
6. Adicione, confirme e envie:

```
git add .
git commit -m "update index.html title"
git push
```

A execução de um fluxo de trabalho é iniciada automaticamente.

7. (Opcional) Insira:

```
git show HEAD
```

Anote o ID de confirmação da alteração `index.html`. Esse ID de confirmação será marcado na imagem do Docker que será implantada pela execução do fluxo de trabalho que você acabou de iniciar.

8. Assista ao andamento da implantação:



- a. No CodeCatalyst console, no painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
  - b. Escolha `codecatalyst-eks-workflow` para ver a última execução.
  - c. Escolha BuildBackende DeployToEKS para ver o progresso da execução do fluxo de trabalho.
9. Verifique se a aplicação foi atualizada, da seguinte forma:
- a. Abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
  - b. À esquerda, na parte inferior, escolha Balanceadores de carga.
  - c. Selecione o balanceador de carga que foi criado como parte da sua implantação do Kubernetes.
  - d. Copie e cole o valor do Nome do DNS na barra de endereço do navegador.
- A página da Web “Tutorial concluído!” aparece no navegador, indicando que você implantou uma nova revisão da aplicação com êxito.
10. (Opcional) Em AWS, alterne para o console do Amazon ECR e verifique se a nova imagem do Docker foi marcada com o ID de confirmação da etapa 7 deste procedimento.

## Limpeza

Você deve limpar seu ambiente para não receber cobranças desnecessárias pelos recursos de armazenamento e computação usados neste tutorial.

### Para limpar

1. Exclua seu cluster:

- No terminal do Ambiente de Desenvolvimento, insira:

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Em que:

- *us-west-2* é substituído pela sua região.
- *codecatalyst-eks-cluster* é substituído pelo nome do cluster que você criou.

Depois de 5 a 10 minutos, o cluster e os recursos associados são excluídos, incluindo, mas não se limitando a, CloudFormation pilhas, grupos de nós (na Amazon EC2) e balanceadores de carga.

 Important

Se o `eksctl delete cluster` comando não funcionar, talvez seja necessário atualizar suas AWS credenciais ou suas `kubectl` credenciais. Se você não tiver certeza de quais credenciais atualizar, atualize as AWS credenciais primeiro. Para atualizar suas credenciais da AWS, consulte [Como faço para corrigir os erros “Não é possível localizar credenciais” e “ExpiredToken”?](#). Para atualizar suas credenciais do `kubectl`, consulte [Como faço para corrigir os erros “Não foi possível conectar-se ao servidor”?](#).

2. No AWS console, limpe da seguinte forma:
  1. No Amazon ECR, exclua `codecatalyst-eks-image-repo`.
  2. No IAM Identity Center, exclua:
    - a. `codecatalyst-eks-user`
    - b. `codecatalyst-eks-permission-set`
  3. No IAM, exclua:
    - `codecatalyst-eks-build-role`
    - `codecatalyst-eks-deploy-role`
    - `codecatalyst-eks-build-policy`
    - `codecatalyst-eks-deploy-policy`
3. No CodeCatalyst console, limpe da seguinte forma:
  1. Exclua `codecatalyst-eks-workflow`.
  2. Exclua `codecatalyst-eks-environment`.
  3. Exclua `codecatalyst-eks-source-repository`.
  4. Exclua o Ambiente de Desenvolvimento.
  5. Exclua `codecatalyst-eks-project`.

Neste tutorial, você aprendeu a implantar um aplicativo em um serviço Amazon EKS usando um CodeCatalyst fluxo de trabalho e uma ação de cluster Deploy to Kubernetes.

## Adição da ação “Implantar no cluster do Kubernetes”

Use as instruções a seguir para adicionar a ação Implantar no cluster do Kubernetes ao seu fluxo de trabalho.

Antes de começar

Antes de adicionar a ação Implantar no cluster do Kubernetes ao seu fluxo de trabalho, você deve preparar o seguinte:

### Tip

Para configurar esses pré-requisitos rapidamente, siga as instruções em [Tutorial: Implantar uma aplicação no Amazon EKS](#).

- Um cluster do Kubernetes no Amazon EKS. Para ter informações sobre clusters, consulte o [Clusters do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- Pelo menos um Dockerfile que descreva como montar a aplicação em uma imagem do Docker. Para ter mais informações sobre Dockerfiles, consulte a [Referência de Dockerfile](#).
- Pelo menos um arquivo de manifesto do Kubernetes, chamado de arquivo de configuração ou configuração na documentação do Kubernetes. Para ter mais informações, consulte [Gerenciamento de recursos](#) na documentação do Kubernetes.
- Um perfil do IAM que dá à ação Implantar no cluster do Kubernetes a capacidade de acessar e interagir com seu cluster do Amazon EKS. Para obter mais informações, consulte o tópico [Role no YAML da ação “Implantar no cluster do Kubernetes”](#).

Depois de criar esse perfil, você deve adicioná-lo a:

- Seu arquivo Kubernetes ConfigMap . Para saber como adicionar uma função a um ConfigMap arquivo, consulte [Habilitar o acesso principal do IAM ao seu cluster](#) no Guia do usuário do Amazon EKS.
- CodeCatalyst. Para saber como adicionar uma função do IAM ao CodeCatalyst, consulte [Adicionar perfis do IAM às conexões da conta](#).
- Um CodeCatalyst espaço, projeto e ambiente. O espaço e o ambiente devem estar conectados à AWS conta na qual você implantará seu aplicativo. Para obter mais informações, consulte [Criar um](#)

[espaço](#), [Criando um projeto vazio na Amazon CodeCatalyst](#) e [Implantação em e Contas da AWS VPCs](#).

- Um repositório de origem suportado pelo CodeCatalyst. O repositório armazena os arquivos de origem da aplicação, Dockerfiles e manifestos do Kubernetes. Para obter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).

## Visual

Para adicionar a ação “Implantar no cluster do Kubernetes” usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Implantar no cluster do Kubernetes e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Selecione Implantar no cluster do Kubernetes. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [YAML da ação “Implantar no cluster do Kubernetes”](#). Essa referência fornece informações detalhadas sobre cada

campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.

11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação “Implantar no cluster do Kubernetes” usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Implantar no cluster do Kubernetes e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Selecione Implantar no cluster do Kubernetes. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [YAML da ação “Implantar no cluster do Kubernetes”](#).

11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Variáveis de “Implantar no cluster do Kubernetes”

A ação Implantar no cluster do Kubernetes produz e define as seguintes variáveis em runtime. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave               | Valor                                                                                                                                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cluster             | <p>O nome do recurso da Amazon.com (ARN) do cluster do Amazon EKS no qual foi implantado durante a execução do fluxo de trabalho.</p> <p>Exemplo: <code>arn:aws:eks:us-west-2:111122223333:cluster/codacatalyst-eks-cluster</code></p> |
| deployment-platform | <p>O nome da plataforma de implantação.</p> <p>Codificado para AWS : EKS.</p>                                                                                                                                                          |
| metadata            | <p>Reservado. Metadados com formatação JSON relacionados ao cluster implantado durante a execução do fluxo de trabalho.</p>                                                                                                            |
| namespace           | <p>O namespace do Kubernetes no qual o cluster foi implantado.</p> <p>Exemplo: <code>default</code></p>                                                                                                                                |
| recursos            | <p>Reservado. Metadados com formatação JSON relacionados aos recursos implantados durante a execução do fluxo de trabalho.</p>                                                                                                         |

| Chave    | Valor                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| servidor | <p>O nome do endpoint do servidor de API que você pode usar para se comunicar com seu cluster usando ferramentas de gerenciamento, como <code>kubectl</code>.</p> <p>Para ter mais informações sobre o endpoint de serviço da API, consulte o <a href="#">Controle de acesso ao endpoint do cluster do Amazon EKS</a> no Guia do usuário do Amazon EKS.</p> <p>Exemplo: <code>https://random-string.gr7.us-west-2.eks.amazonaws.com</code></p> |

## YAML da ação “Implantar no cluster do Kubernetes”

Confira a seguir a definição de YAML da ação Implantar no cluster do Kubernetes. Para saber como usar essa ação, consulte [Implantar no Amazon EKS com um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:
```

```
The action definition starts here.
DeployToKubernetesCluster_nn:
 Identifier: aws/kubernetes-deploy@v1
 DependsOn:
 - build-action
 Compute:
 - Type: EC2 | Lambda
 - Fleet: fleet-name
 Timeout: timeout-minutes
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: DeployToEKS
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - manifest-artifact
 Configuration:
 Namespace: namespace
 Region: us-east-1
 Cluster: eks-cluster
 Manifests: manifest-path
```

## DeployToKubernetesCluster

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: DeployToKubernetesCluster\_nn.

Interface de usuário correspondente: guia Configuração/Nome de exibição da ação

Identifier

(*DeployToKubernetesCluster*/Identifier)



## (Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/kubernetes-deploy@v1.

Interface de usuário correspondente: Diagrama de fluxo de trabalho/DeployToKubernetesCluster\_nn/  
rótulo aws/kubernetes-deploy@v1

DependsOn

(*DeployToKubernetesCluster*/DependsOn)

## (Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*DeployToKubernetesCluster*/Compute)

## (Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Type

(*DeployToKubernetesCluster*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: Configuração tab/Advanced - opcional/Tipo de computação

Fleet

(*DeployToKubernetesCluster*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

UI correspondente: configuração tab/Advanced - opcional/frota de computação

Timeout

(*DeployToKubernetesCluster*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

Environment

(*DeployToKubernetesCluster*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs e Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*DeployToKubernetesCluster*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*DeployToKubernetesCluster*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Name

(*DeployToKubernetesCluster*/Environment/Connections/Name)

(Optional)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome do perfil do IAM que a ação Implantar no cluster do Kubernetes usa para acessar a AWS. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

 Warning

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "eks:DescribeCluster",
 "eks:ListClusters"
],
 "Resource": "*"
 }
]
}
```

 Note

Na primeira vez em que o perfil for usado, use o caractere curinga a seguir na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A política de confiança personalizada a seguir:

Esse perfil deve ser adicionado a:

- Sua conexão de conta. Para saber mais sobre como adicionar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#).
- Seu Kubernetes ConfigMap. Para saber mais sobre como adicionar uma função do IAM a uma ConfigMap, consulte [Gerenciar usuários e funções do IAM](#) na eksctl documentação.

#### Tip

Consulte também [Tutorial: Implantar uma aplicação no Amazon EKS](#) para obter instruções sobre como adicionar uma função do IAM a uma conexão de conta ConfigMap e.

#### Note

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/ ' / Função Environment/account/role

## Inputs

(*DeployToKubernetesCluster*/Inputs)

(Obrigatório se [Connections](#) for incluído)

A seção Inputs define os dados que DeployToKubernetesCluster precisa durante a execução de um fluxo de trabalho.

### Note

Somente uma entrada (uma origem ou um artefato) é permitida por ação Implantar no Amazon EKS.

Interface de usuário correspondente: guia Entradas

## Sources

(*DeployToKubernetesCluster*/Inputs/Sources)

(Obrigatório se o arquivo de manifesto estiver armazenado em um repositório de origem)

Se os arquivos de manifesto do Kubernetes estiverem armazenados em um repositório de origem, especifique o rótulo do repositório de origem. Atualmente, o único rótulo compatível é WorkflowSource.

Se os arquivos de manifesto não estiverem em um repositório de origem, eles deverão residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*DeployToKubernetesCluster*/Inputs/Artifacts)

(Obrigatório se o arquivo de manifesto estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se os arquivos de manifesto do Kubernetes estiverem contidos em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se os arquivos de manifesto não estiverem em um artefato, eles deverão residir em seu repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

Configuration

(*DeployToKubernetesCluster*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

Namespace

(*DeployToKubernetesCluster*/Configuration/Namespace)

(Optional)

Especifique o namespace do Kubernetes no qual a aplicação Kubernetes será implantada. Use default se você não estiver usando namespaces com seu cluster. Para ter mais informações sobre namespaces, consulte [Subdivisão do cluster usando namespaces do Kubernetes](#) na documentação do Kubernetes.

Se você omitir o namespace, um valor de default será usado.

Interface de usuário correspondente: guia Configuração/Namespace

Region

(*DeployToKubernetesCluster*/Configuration/Region)

(Obrigatório)

Especifique a AWS região em que seu cluster e serviço do Amazon EKS residem. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.

Interface de usuário correspondente: guia Configuração/Região



## Cluster

(*DeployToKubernetesCluster*/Configuration/Cluster)

(Obrigatório)

Especifique o nome de um cluster existente do Amazon EKS. A ação Implantar no cluster do Kubernetes implantará a aplicação em contêineres nesse cluster. Para ter informações sobre clusters do Amazon EKS, consulte [Clusters](#) no Guia do usuário do Amazon EKS.

Interface de usuário correspondente: guia Configuração/Cluster

## Manifests

(*DeployToKubernetesCluster*/Configuration/Manifests)

(Obrigatório)

Especifique o caminho para seus arquivos de manifesto do Kubernetes formatados em YAML, que são chamados de arquivos de configuração ou, simplesmente, configurações na documentação do Kubernetes.

Se você estiver usando vários arquivos de manifesto, coloque-os em uma única pasta e faça referência a essa pasta. Os arquivos de manifesto são processados alfanumericamente pelo Kubernetes, portanto, prefixe os nomes dos arquivos com números ou letras crescentes para controlar a ordem de processamento. Por exemplo:

00-namespace.yaml

01-deployment.yaml

Se os arquivos de manifesto residirem em seu repositório de origem, o caminho é relativo à pasta raiz do repositório de origem. Se os arquivos residirem em um artefato de uma ação anterior do fluxo de trabalho, o caminho é relativo à pasta raiz do artefato.

Exemplos:

Manifests/

deployment.yaml

my-deployment.yaml

Não use curingas (\*).

**Note**

Os [charts do Helm](#) e os [arquivos de personalização](#) não são aceitos.

Para ter mais informações sobre os arquivos de manifesto, consulte [Como organizar as configurações de recursos](#) na documentação do Kubernetes.

Interface de usuário correspondente: guia Configuração/Manifestos

## Implantação de uma pilha CloudFormation

Esta seção descreve como implantar uma AWS CloudFormation pilha usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar a ação Deploy CloudFormation stack ao seu fluxo de trabalho. A ação implanta uma CloudFormation pilha de recursos AWS com base em um modelo fornecido por você. O modelo pode ser um:

- CloudFormation modelo — Para obter mais informações, consulte Como [trabalhar com CloudFormation modelos](#).
- AWS SAM modelo — Para obter mais informações, consulte a [especificação AWS Serverless Application Model \(AWS SAM\)](#).

**Note**

Para usar um AWS SAM modelo, primeiro você deve empacotar seu AWS SAM aplicativo usando a [sam package](#) operação. Para ver um tutorial que mostra como fazer esse empacotamento automaticamente como parte de um CodeCatalyst fluxo de trabalho da Amazon, consulte [Tutorial: Implantar uma aplicação sem servidor](#).

Se a pilha já existir, a ação executará a CloudFormation [CreateChangeSet](#) operação e, em seguida, a [ExecuteChangeSet](#) operação. A ação espera que as alterações sejam implantadas e se marca como bem-sucedida ou com falha, dependendo dos resultados.

Use a ação Deploy CloudFormation stack se você já tiver um AWS SAM modelo CloudFormation ou que contenha recursos que gostaria de implantar, ou se planeja gerar um automaticamente como parte de uma [ação de criação](#) de fluxo de trabalho usando ferramentas como AWS SAM e. [AWS Cloud Development Kit \(AWS CDK\)](#)

Não há restrições sobre o modelo que você pode usar, seja o que for que você possa criar CloudFormation ou AWS SAM usar com a ação Deploy CloudFormation stack.

 Tip

Para ver um tutorial que mostra como implantar um aplicativo sem servidor usando a ação Deploy CloudFormation stack, consulte. [Tutorial: Implantar uma aplicação sem servidor](#)

## Tópicos

- [Imagem de tempo de execução usada pela ação 'Deploy CloudFormation stack'](#)
- [Tutorial: Implantar uma aplicação sem servidor](#)
- [Adicionando a ação “Deploy CloudFormation stack”](#)
- [Configurar reversões](#)
- [Variáveis de “ CloudFormation pilha de implantação”](#)
- [Ação “Implantar CloudFormation pilha” YAML](#)

## Imagem de tempo de execução usada pela ação 'Deploy CloudFormation stack'

A ação Deploy CloudFormation stack é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Tutorial: Implantar uma aplicação sem servidor

Neste tutorial, você aprende a criar, testar e implantar um aplicativo sem servidor como uma CloudFormation pilha usando um fluxo de trabalho.

A aplicação neste tutorial é uma aplicação web simples que gera uma mensagem “Hello World”. Ele consiste em uma AWS Lambda função e um Amazon API Gateway, e você o cria usando o [AWS Serverless Application Model \(AWS SAM\)](#), que é uma extensão do [CloudFormation](#).

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um repositório de origem](#)
- [Etapa 2: criar AWS funções](#)

- [Etapa 3: adicionar AWS funções a CodeCatalyst](#)
- [Etapa 4: criar um bucket do Amazon S3](#)
- [Etapa 5: adicionar arquivos de origem](#)
- [Etapa 6: criar e executar um fluxo de trabalho](#)
- [Etapa 7: fazer uma alteração](#)
- [Limpeza](#)

## Pré-requisitos

### Antes de começar

- Você precisa de um CodeCatalyst espaço com uma AWS conta conectada. Para obter mais informações, consulte [Criar um espaço](#).
- Em seu espaço, você precisa de um projeto vazio chamado:

```
codecatalyst-cfn-project
```

Use a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- Em seu projeto, você precisa de um CodeCatalyst ambiente chamado:

```
codecatalyst-cfn-environment
```

Configure esse ambiente da seguinte forma:

- Escolha qualquer tipo, como Não produção.
- Conecte sua AWS conta a ela.
- Para o Perfil do IAM padrão, escolha qualquer perfil. Você especificará um perfil diferente posteriormente.

Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).

## Etapa 1: criar um repositório de origem

Nesta etapa, você cria um repositório de origem no CodeCatalyst. Esse repositório é usado para armazenar os arquivos de origem do tutorial, como o arquivo da função do Lambda.

Para ter mais informações sobre repositórios de origem, consulte [Criar um repositório de origem](#).

### Como criar um repositório de origem

1. Em CodeCatalyst, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.
2. Escolha Adicionar repositório e selecione Criar repositório.
3. Em Nome do repositório, insira:

```
codecatalyst-cfn-source-repository
```

4. Escolha Criar.

Agora você criou um repositório chamado `codecatalyst-cfn-source-repository`.

### Etapa 2: criar AWS funções

Nesta etapa, você cria as seguintes funções AWS do IAM:

- Função de implantação — concede permissão à ação CodeCatalyst Deploy CloudFormation stack para acessar sua AWS conta e CloudFormation serviço onde você implantará seu aplicativo sem servidor. A ação Deploy CloudFormation stack faz parte do seu fluxo de trabalho.
- Função de criação — concede à ação de CodeCatalyst criação permissão para acessar sua AWS conta e gravar no Amazon S3, onde seu pacote de aplicativos sem servidor será armazenado. A ação de criação faz parte do seu fluxo de trabalho.
- Função do Stack — concede CloudFormation permissão para ler e modificar os recursos especificados no AWS SAM modelo que você fornecerá posteriormente. Também concede permissão para CloudWatch.

Para ter informações sobre perfis do IAM, consulte [Perfis do IAM](#) no Guia do usuário do AWS Identity and Access Management .

#### Note

Para economizar tempo, você pode criar um único perfil, chamado `CodeCatalystWorkflowDevelopmentRole-spaceName`, em vez dos três perfis listados anteriormente. Para obter mais informações, consulte [Criar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` para a conta e o espaço](#). Entenda que

o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões bem amplas que podem representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando os três perfis listados anteriormente.

 Note

Um [perfil de execução do Lambda](#) também é necessário, mas você não precisa criá-lo agora porque o arquivo `sam-template.yml` o cria quando você executa o fluxo de trabalho na etapa 5.

Para criar um perfil de implantação

1. Crie uma política para o perfil da seguinte maneira:

- a. Faça login em AWS.
- b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, selecione Políticas.
- d. Escolha Criar política.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.

- j. Em Nome, insira:

**codecatalyst-deploy-policy**

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de implantação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure **codecatalyst-deploy-policy** e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

**codecatalyst-deploy-role**

- i. Em Descrição do perfil, insira:

**CodeCatalyst deploy role**

- j. Selecione Criar perfil.

Agora você criou um perfil de implantação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN do perfil de implantação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (**codecatalyst-deploy-role**).
- c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de implantação com as permissões apropriadas e obteve o ARN.

## Como criar um perfil de criação

1. Crie uma política para o perfil da seguinte maneira:

- a. Faça login em AWS.
- b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, selecione Políticas.
- d. Escolha Criar política.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-build-policy
```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.



- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure `codecatalyst-build-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

`codecatalyst-build-role`

- i. Em Descrição do perfil, insira:

`CodeCatalyst build role`

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN do perfil de criação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (`codecatalyst-build-role`).
- c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de criação com as permissões apropriadas e obteve o ARN.

Para criar um perfil de pilha

1. Faça login AWS usando a conta na qual você deseja implantar sua pilha.

2. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

### 3. Crie o perfil de pilha, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Escolha Criar Perfil.
- c. Selecione Serviço da AWS .
- d. Na seção Caso de uso, escolha na CloudFormation lista suspensa.
- e. Selecione o botão CloudFormation de rádio.
- f. Selecione Avançar na parte inferior.
- g. Usando a caixa de pesquisa, encontre as políticas de permissões a seguir e marque as respectivas caixas de seleção.

#### Note

Se você procurar uma política e ela não aparecer, selecione Limpar filtros e tente novamente.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- IAMFullAcesso
- AWS Lambda\_ FullAccess
- APIGatewayAdministrador da Amazon
- Amazon S3 FullAccess
- AmazonEC2ContainerRegistryFullAccess

A primeira política permite o acesso para CloudWatch habilitar reversões de pilha quando ocorre um alarme.

As políticas restantes permitem AWS SAM acessar os serviços e recursos na pilha que serão implantados neste tutorial. Para ter mais informações, consulte [Permissões](#) no Guia do desenvolvedor do AWS Serverless Application Model .

- h. Escolha Próximo.
- i. Em Nome do perfil, insira:

**codecatalyst-stack-role**

- j. Selecione Criar perfil.
4. Obtenha o ARN do perfil de pilha, da seguinte forma:
  - a. No painel de navegação, selecione Perfis.
  - b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (codecatalyst-stack-role).
  - c. Escolha o perfil na lista.
  - d. Na seção Resumo, copie o valor de ARN. Você precisará disso mais tarde.

Agora você criou o perfil de pilha com as permissões apropriadas e obteve o ARN.

### Etapa 3: adicionar AWS funções a CodeCatalyst

Nesta etapa, você adiciona a função de criação (codecatalyst-build-role) e a função de implantação (codecatalyst-deploy-role) à conexão da CodeCatalyst conta em seu espaço.

#### Note

Não é necessário adicionar o perfil de pilha (codecatalyst-stack-role) à conexão. Isso ocorre porque a função de pilha é usada por CloudFormation (não CodeCatalyst), depois que uma conexão já foi estabelecida entre CodeCatalyst e AWS usando a função de implantação. Como a função de pilha não é usada por CodeCatalyst para obter acesso AWS, ela não precisa ser associada a uma conexão de conta.

Para adicionar perfis de criação e implantação à conexão da sua conta

1. Dentro CodeCatalyst, navegue até seu espaço.
2. Selecione Contas da AWS . Uma lista de conexões de conta é exibida.
3. Escolha a conexão de conta que representa a AWS conta em que você criou suas funções de criação e implantação.
4. Escolha Gerenciar funções no console de AWS gerenciamento.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é exibida. Talvez seja necessário fazer login para acessá-la.

5. Selecione Adicionar um perfil existente que você criou no IAM.

Uma lista suspensa aparece. A lista exibe todos os perfis do IAM com uma política de confiança que inclui as entidades principais de serviço `codecatalyst-runner.amazonaws.com` e `codecatalyst.amazonaws.com`.

6. Na lista suspensa, selecione `codecatalyst-build-role` e Adicionar perfil.
7. Escolha Adicionar perfil do IAM, selecione Adicionar um perfil existente que você criou no IAM e, na lista suspensa, escolha `codecatalyst-deploy-role`. Escolha Add role (adicionar função).

Agora você adicionou os perfis de criação e implantação ao seu espaço.

8. Copie o valor do nome de CodeCatalyst exibição da Amazon. Esse valor será necessário posteriormente, ao criar seu fluxo de trabalho.

#### Etapa 4: criar um bucket do Amazon S3

Nesta etapa, você cria um bucket do Amazon S3 no qual armazena o arquivo .zip do pacote de implantação da aplicação sem servidor.

##### Como criar um bucket do Amazon S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel principal, selecione Criar bucket.
3. Em Nome do bucket, insira:

**`codecatalyst-cfn-s3-bucket`**

4. Na região da AWS, escolha uma região. Este tutorial pressupõe que você tenha escolhido o Oeste dos EUA (Oregon) `us-west-2`. Para ter informações sobre regiões com suporte do Amazon S3, consulte [Endpoints e cotas do Amazon Simple Storage Service](#) na Referência geral da AWS.
5. Selecione Criar bucket na parte inferior da página.

Agora você criou um bucket chamado **codecatalyst-cfn-s3-bucket** na região Oeste dos EUA (Oregon), us-west-2.

### Etapa 5: adicionar arquivos de origem

Nesta etapa, você adiciona vários arquivos de origem do aplicativo ao seu repositório CodeCatalyst de origem. A pasta `hello-world` contém os arquivos da aplicação que você implantará. A pasta `tests` contém testes de unidade. A estrutura de pastas é esta:

```
.
|- hello-world
| |- tests
| |- unit
| |- test-handler.js
| |- app.js
|- .npmignore
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

### Arquivo .npmignore

O arquivo `.npmignore` indica quais arquivos e pastas o npm deve excluir do pacote da aplicação. Neste tutorial, o npm exclui a pasta `tests` porque ela não faz parte da aplicação.

### Como adicionar o arquivo .npmignore

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto, `codecatalyst-cfn-project`
3. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
4. Na lista de repositórios de origem, escolha o seu repositório, `codecatalyst-cfn-source-repository`.
5. Em Arquivos, selecione Criar arquivo.
6. Em Nome do arquivo, insira:

```
.npmignore
```

7. Na caixa de texto, insira o código a seguir:

```
tests/*
```

8. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você criou um arquivo chamado `.npmignore` na raiz do repositório.

## Arquivo package.json

O arquivo `package.json` contém metadados importantes sobre seu projeto Node, como nome do projeto, número da versão, descrição, dependências e outros detalhes que descrevem como interagir e executar a aplicação.

O `package.json` neste tutorial inclui uma lista de dependências e um script `test`. O script de teste faz o seguinte:

- Usando [mocha](#), o script de teste executa os testes de unidade especificados em `hello-world/tests/unit/` e grava os resultados em um arquivo `junit.xml` usando o repórter [xunit](#).
- Usando [Istanbul \(nyc\)](#), o script de teste gera um relatório de cobertura de código (`clover.xml`) usando o repórter [clover](#). Para ter mais informações, consulte [Uso de repórteres alternativos](#) na documentação do Istanbul.

## Como adicionar o arquivo package.json

1. No repositório, em Arquivos, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

```
package.json
```

3. Na caixa de texto, insira o código a seguir:

```
{
 "name": "hello_world",
 "version": "1.0.0",
 "description": "hello world sample for NodeJS",
 "main": "app.js",
 "repository": "https://github.com/awslabs/aws-sam-cli/tree/develop/samcli/local/init/templates/cookiecutter-aws-sam-hello-nodejs",
 "author": "SAM CLI",
 "license": "MIT",
```

```
"dependencies": {
 "axios": "^0.21.1",
 "nyc": "^15.1.0"
},
"scripts": {
 "test": "nyc --reporter=clover mocha hello-world/tests/unit/ --reporter xunit
--reporter-option output=junit.xml"
},
"devDependencies": {
 "aws-sdk": "^2.815.0",
 "chai": "^4.2.0",
 "mocha": "^8.2.1"
}
}
```

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `package.json` à raiz do repositório.

#### Arquivo `sam-template.yml`

O arquivo `sam-template.yml` contém as instruções para implantar a função do Lambda e o gateway de API e configurá-los juntos. Ele segue a [especificação do AWS Serverless Application Model modelo](#), que estende a especificação do CloudFormation modelo.

Você usa um AWS SAM modelo neste tutorial em vez de um CloudFormation modelo normal porque AWS SAM oferece um tipo de recurso útil [AWS::Serverless::Function](#). Esse tipo executa muitas behind-the-scenes configurações que você normalmente precisa escrever para usar a CloudFormation sintaxe básica. Por exemplo, o `AWS::Serverless::Function` cria uma função do Lambda, uma função de execução do Lambda e mapeamentos da origem do evento que iniciam a função. Você precisa codificar tudo isso se quiser escrevê-lo usando o básico CloudFormation.

Embora este tutorial use um modelo pré-escrito, você pode gerar um como parte do seu fluxo de trabalho usando uma ação de criação. Para obter mais informações, consulte [Implantação de uma pilha CloudFormation](#).

Para adicionar o arquivo `sam-template.yml`

1. No repositório, em Arquivos, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

**sam-template.yml**

## 3. Na caixa de texto, insira o código a seguir:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
 serverless-api

 Sample SAM Template for serverless-api

More info about Globals: https://github.com/awslabs/serverless-application-model/
blob/master/docs/globals.rst
Globals:
 Function:
 Timeout: 3

Resources:
 HelloWorldFunction:
 Type: AWS::Serverless::Function # For details on this resource type,
 see https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#awsserverlessfunction
 Properties:
 CodeUri: hello-world/
 Handler: app.lambdaHandler
 Runtime: nodejs12.x
 Events:
 HelloWorld:
 Type: Api # For details on this event source type, see
 https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#api
 Properties:
 Path: /hello
 Method: get

Outputs:
 # ServerlessRestApi is an implicit API created out of the events key under
 Serverless::Function
 # Find out about other implicit resources you can reference within AWS SAM at
 # https://github.com/awslabs/serverless-application-model/blob/master/docs/
internals/generated_resources.rst#api
 HelloWorldApi:
```



```
Description: "API Gateway endpoint URL for the Hello World function"
Value: !Sub "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/hello/"
HelloWorldFunction:
 Description: "Hello World Lambda function ARN"
 Value: !GetAtt HelloWorldFunction.Arn
HelloWorldFunctionIamRole:
 Description: "Implicit Lambda execution role created for the Hello World
function"
 Value: !GetAtt HelloWorldFunctionRole.Arn
```

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `sam-template.yml` à pasta raiz do repositório.

#### Arquivo `setup-sam.sh`

O `setup-sam.sh` arquivo contém as instruções para baixar e instalar o utilitário AWS SAM CLI. O fluxo de trabalho usa esse utilitário para empacotar a origem de `hello-world`.

Para adicionar o arquivo `setup-sam.sh`

1. No repositório, em Arquivos, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

```
setup-sam.sh
```

3. Na caixa de texto, insira o código a seguir:

```
#!/usr/bin/env bash
echo "Setting up sam"

yum install unzip -y

curl -LO https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-
linux-x86_64.zip
unzip -qq aws-sam-cli-linux-x86_64.zip -d sam-installation-directory

./sam-installation-directory/install; export AWS_DEFAULT_REGION=us-west-2
```

No código anterior, *us-west-2* substitua por sua AWS região.

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `setup-sam.sh` à raiz do repositório.

### Arquivo `app.js`

O `app.js` contém o código da função do Lambda. Neste tutorial, o código retorna o texto `hello world`.

Para adicionar o arquivo `app.js`

1. No repositório, em Arquivos, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

**hello-world/app.js**

3. Na caixa de texto, insira o código a seguir:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;

/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 */
exports.lambdaHandler = async (event, context) => {
 try {
 // const ret = await axios(url);
 response = {
 'statusCode': 200,
```

```
 'body': JSON.stringify({
 message: 'hello world',
 // location: ret.data.trim()
 })
 }
} catch (err) {
 console.log(err);
 return err;
}

return response
};
```

4. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você criou uma pasta chamada `hello-world` e um arquivo chamado `app.js`.

#### Arquivo `test-handler.js`

O arquivo `test-handler.js` contém testes de unidade para a função do Lambda.

Para adicionar o arquivo `test-handler.js`

1. No repositório, em Arquivos, selecione Criar arquivo.
2. Em Nome do arquivo, insira:

**`hello-world/tests/unit/test-handler.js`**

3. Na caixa de texto, insira o código a seguir:

```
'use strict';

const app = require('../../app.js');
const chai = require('chai');
const expect = chai.expect;
var event, context;

describe('Tests index', function () {
 it('verifies successful response', async () => {
 const result = await app.lambdaHandler(event, context)

 expect(result).to.be.an('object');
```

```
 expect(result.statusCode).to.equal(200);
 expect(result.body).to.be.an('string');

 let response = JSON.parse(result.body);

 expect(response).to.be.an('object');
 expect(response.message).to.be.equal("hello world");
 // expect(response.location).to.be.an("string");
 });
});
```

#### 4. Selecione Confirmar e, depois, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `test-handler.js` abaixo da pasta `hello-world/tests/unit`.

Agora você adicionou todos os seus arquivos de origem.

Reserve um momento para verificar seu trabalho e garantir que colocou todos os arquivos nas pastas corretas. A estrutura de pastas é esta:

```
.
|- hello-world
| |- tests
| |- unit
| |- test-handler.js
| |- app.js
|- .npmignore
|- README.md
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

#### Etapa 6: criar e executar um fluxo de trabalho

Nesta etapa, você cria um fluxo de trabalho que empacota o código-fonte do Lambda e o implanta. O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

- Uma ação de teste (Test) — No gatilho, essa ação instala o [gerenciador de pacotes Node \(npm\)](#) e, depois, executa o comando `npm run test`. Esse comando diz ao npm para executar o script `test` definido no arquivo `package.json`. O script `test`, por sua vez, executa os testes de unidade e gera dois relatórios: um relatório de teste (`junit.xml`) e um relatório de cobertura de código (`clover.xml`). Para obter mais informações, consulte [Arquivo package.json](#).

Em seguida, a ação de teste transforma os relatórios XML em CodeCatalyst relatórios e os exibe no CodeCatalyst console, na guia Relatórios da ação de teste.

Para ter mais informações sobre a ação de teste, consulte [Teste com fluxos de trabalho](#).

- Uma ação de construção (BuildBackend) — Ao concluir a ação de teste, a ação de compilação baixa e instala a AWS SAM CLI, empacota `hello-world` a fonte e copia o pacote em seu bucket do Amazon S3, onde o serviço Lambda espera que esteja. A ação também gera um novo arquivo de AWS SAM modelo chamado `sam-template-packaged.yml` e o coloca em um artefato de saída chamado `buildArtifact`.

Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).

- Uma ação de implantação (DeployCloudFormationStack) — Ao concluir a ação de criação, a ação de implantação procura o artefato de saída gerado pela ação de construção (`buildArtifact`), encontra o AWS SAM modelo dentro dela e, em seguida, executa o modelo. O AWS SAM modelo cria uma pilha que implanta o aplicativo sem servidor.

Como criar um fluxo de trabalho

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Selecione Criar fluxo de trabalho.
3. Em Repositório de origem, selecione `codecatalyst-cfn-source-repository`.
4. Em Ramificação, selecione `main`.
5. Escolha Criar.
6. Exclua o código de amostra YAML.
7. Adicione o seguinte código YAML:

 Note

No código YAML a seguir, você pode omitir as seções `Connections`: se quiser. Se você omitir essas seções, deverá garantir que o perfil especificado no campo Perfil do

IAM padrão em seu ambiente inclua as permissões e as políticas de confiança dos dois perfis descritas em [Etapa 2: criar AWS funções](#). Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#).

```
Name: codecatalyst-cfn-workflow
SchemaVersion: 1.0

Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 Test:
 Identifier: aws/managed-test@v1
 Inputs:
 Sources:
 - WorkflowSource
 Outputs:
 Reports:
 CoverageReport:
 Format: CLOVERXML
 IncludePaths:
 - "coverage/*"
 TestReport:
 Format: JUNITXML
 IncludePaths:
 - junit.xml
 Configuration:
 Steps:
 - Run: npm install
 - Run: npm run test
 BuildBackend:
 Identifier: aws/build@v1
 DependsOn:
 - Test
 Environment:
 Name: codecatalyst-cfn-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-build-role
 Inputs:
```

```

Sources:
 - WorkflowSource
Configuration:
 Steps:
 - Run: . ./setup-sam.sh
 - Run: sam package --template-file sam-template.yml --s3-
bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml
--region us-west-2
 Outputs:
 Artifacts:
 - Name: buildArtifact
 Files:
 - "**/*"
DeployCloudFormationStack:
 Identifier: aws/cfn-deploy@v1
 DependsOn:
 - BuildBackend
 Environment:
 Name: codecatalyst-cfn-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-deploy-role
 Inputs:
 Artifacts:
 - buildArtifact
 Sources: []
 Configuration:
 name: codecatalyst-cfn-stack
 region: us-west-2
 role-arn: arn:aws:iam::111122223333:role/StackRole
 template: ./sam-template-packaged.yml
 capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND

```

No código anterior, substitua:

- Ambas as instâncias de *codecatalyst-cfn-environment* com o nome do seu ambiente.
- Ambas as instâncias *codecatalyst-account-connection* com o nome de exibição da conexão da sua conta. O nome de exibição pode ser um número. Para obter mais informações, consulte [Etapa 3: adicionar AWS funções a CodeCatalyst](#).
- *codecatalyst-build-role* pelo nome do perfil de criação criado em [Etapa 2: criar AWS funções](#).

- `codecatalyst-cfn-s3-bucket` com o nome do bucket do Amazon S3 em que você criou. [Etapa 4: criar um bucket do Amazon S3](#)
- Ambas as instâncias são `us-west-2` com a região em que seu bucket do Amazon S3 reside (primeira instância) e onde sua pilha será implantada (segunda instância). Essas regiões podem ser diferentes. Este tutorial pressupõe que as duas regiões estejam definidas como `us-west-2`. Para obter detalhes sobre regiões suportadas pelo Amazon S3 CloudFormation, consulte [Pontos de extremidade e cotas de serviço](#) no. Referência geral da AWS
- `codecatalyst-deploy-role` pelo nome do perfil de implantação criado em [Etapa 2: criar AWS funções](#).
- `codecatalyst-cfn-environment` com o nome do ambiente em que você criou [Pré-requisitos](#).
- `arn:aws:iam::111122223333:role/StackRole` com o Amazon Resource Name (ARN) da função de pilha que você criou. [Etapa 2: criar AWS funções](#)

 Note

Se você decidiu não criar, implantar e empilhar funções `codecatalyst-build-role` `codecatalyst-deploy-role`, substitua e `arn:aws:iam::111122223333:role/StackRole` pelo nome ou ARN da CodeCatalystWorkflowDevelopmentRole-`spaceName` função. Para obter mais informações sobre essa função, consulte [Etapa 2: criar AWS funções](#).

Para ter informações sobre as propriedades no código mostrado anteriormente, consulte a [Ação “Implantar CloudFormation pilha” YAML](#).

8. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
9. Selecione Confirmar.
10. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
  - a. Em Nome do arquivo do fluxo de trabalho, mantenha o padrão, `codecatalyst-cfn-workflow`.
  - b. Em Confirmar mensagem, insira:

`add initial workflow file`

- c. Em Repositório, selecione `codecatalyst-cfn-source-repository`.



- d. Em Nome da ramificação, selecione principal.
- e. Selecione Confirmar.

Agora você criou um fluxo de trabalho. A execução de um fluxo de trabalho é iniciada automaticamente devido ao gatilho definido na parte superior do fluxo de trabalho. Especificamente, quando você confirmou (e enviou) o arquivo `codecatalyst-cfn-workflow.yaml` ao repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para visualizar a execução do fluxo de trabalho em andamento

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar: `codecatalyst-cfn-workflow`.
3. Escolha a guia Execuções.
4. Na coluna ID da execução, escolha o ID da execução.
5. Selecione Teste para ver o andamento dos testes.
6. Escolha BuildBackendver o progresso da compilação.
7. Escolha DeployCloudFormationStackver o progresso da implantação.

Para ter mais informações sobre como visualizar os detalhes da execução, consulte o [Visualização do status e detalhes de execução do fluxo de trabalho](#).

8. Quando a DeployCloudFormationStackação terminar, faça o seguinte:
  - Se a execução do fluxo de trabalho for bem-sucedida, vá para o próximo procedimento.
  - Se a execução do fluxo de trabalho falhar no teste ou na BuildBackendação, escolha Logs para solucionar o problema.
  - Se a execução do fluxo de trabalho falhar na DeployCloudFormationStackação, escolha a ação de implantação e, em seguida, escolha a guia Resumo. Role até a seção de CloudFormation eventos para ver a mensagem de erro detalhada. Se ocorrer uma reversão, exclua a `codecatalyst-cfn-stack` pilha por meio do CloudFormation console AWS antes de executar novamente o fluxo de trabalho.

Como verificar a implantação

1. Depois de uma implantação bem-sucedida, selecione Variáveis (7) na barra de menu horizontal próxima à parte superior. (Não selecione Variáveis no painel à direita.)

2. Em seguida HelloWorldApi, cole o `https://` URL em um navegador.

Uma mensagem JSON hello world da função do Lambda é exibida, indicando que o fluxo de trabalho implantou e configurou a função do Lambda e o gateway de API.

 Tip

Você pode CodeCatalyst exibir esse URL no diagrama do fluxo de trabalho com algumas configurações pequenas. Para obter mais informações, consulte [Exibir o URL da aplicação no diagrama do fluxo de trabalho](#).

Para verificar os resultados do teste de unidade e a cobertura do código

1. No diagrama do fluxo de trabalho, selecione Teste e, depois, escolha Relatórios.
2. Escolha TestReportvisualizar os resultados do teste de unidade ou opte por CoverageReportvisualizar os detalhes da cobertura de código dos arquivos que estão sendo testados, neste caso `app.js` `test-handler.js` e.

Para verificar os recursos implantados

1. Faça login no Console de gerenciamento da AWS e abra o console do API Gateway em <https://console.aws.amazon.com/apigateway/>.
2. Observe a `codecatalyst-cfn-stackAPI` que o AWS SAM modelo criou. O nome da API vem do valor `Configuration/name` no arquivo de definição do fluxo de trabalho (`codecatalyst-cfn-workflow.yaml`).
3. Abra o AWS Lambda console em <https://console.aws.amazon.com/lambda/>.
4. Selecione Funções no painel de navegação.
5. Selecione a função do Lambda, `codecatalyst-cfn-stack-HelloWorldFunction-string`.
6. Você pode ver como o gateway de API é um gatilho para a função. Essa integração foi configurada automaticamente pelo tipo `AWS SAM AWS::Serverless::Function` de recurso.

## Etapa 7: fazer uma alteração

Nesta etapa, você faz uma alteração no código-fonte do Lambda e o confirma. Essa confirmação inicia a execução de um novo fluxo de trabalho. Essa execução implanta a nova função do Lambda em um esquema azul esverdeado que usa a configuração padrão de mudança de tráfego especificada no console do Lambda.

Para fazer uma alteração na origem do Lambda

1. Em CodeCatalyst, navegue até seu projeto.
2. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
3. Escolha seu repositório de origem `codecatalyst-cfn-source-repository`.
4. Altere o arquivo da aplicação:
  - a. Escolha a pasta `hello-world`.
  - b. Escolha o arquivo `app.js`.
  - c. Escolha Editar.
  - d. Na linha 23, altere `hello world` para **Tutorial complete!**.
  - e. Selecione Confirmar e, depois, escolha Confirmar novamente.

A confirmação faz com que uma execução do fluxo de trabalho seja iniciada. Essa execução falhará porque você não atualizou os testes de unidade para refletir a alteração do nome.

5. Atualize os testes de unidade:
  - a. Selecione `hello-world\tests\unit\test-handler.js`.
  - b. Escolha Editar.
  - c. Na linha 19, altere `hello world` para **Tutorial complete!**.
  - d. Selecione Confirmar e, depois, escolha Confirmar novamente.

A confirmação faz com que outra execução do fluxo de trabalho seja iniciada. Essa execução será bem-sucedida.

6. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
7. Selecione `codecatalyst-cfn-workflow` e Execuções.
8. Escolha o ID da última execução. O status ainda deve ser em andamento.

9. Escolha Testar BuildBackend, e DeployCloudFormationStack para ver o progresso da execução do fluxo de trabalho.
10. Quando o fluxo de trabalho terminar, selecione Variáveis (7) na parte superior.
11. Em seguida HelloWorldApi, cole o `https://` URL em um navegador.

Uma mensagem Tutorial complete! aparece no navegador, indicando que a nova aplicação foi implantada.

## Limpeza

Limpe os arquivos e serviços usados neste tutorial para evitar cobranças por eles.

Para limpar no CodeCatalyst console

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Exclua `codecatalyst-cfn-workflow`.
3. Exclua `codecatalyst-cfn-environment`.
4. Exclua `codecatalyst-cfn-source-repository`.
5. Exclua `codecatalyst-cfn-project`.

Para limpar no Console de gerenciamento da AWS

1. Limpe da CloudFormation seguinte forma:
  - a. Abra o CloudFormation console em <https://console.aws.amazon.com/cloudformation>.
  - b. Exclua o `codecatalyst-cfn-stack`.

A exclusão da pilha remove todos os recursos do tutorial dos serviços do gateway de API e Lambda.

2. Limpe no Amazon S3, da seguinte forma:
  - a. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
  - b. Selecione a `codecatalyst-cfn-s3-bucket`.
  - c. Exclua os conteúdos do bucket.
  - d. Exclua o bucket.
3. Limpe no IAM, da seguinte forma:

- a. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- b. Exclua o `codecatalyst-deploy-policy`.
- c. Exclua o `codecatalyst-build-policy`.
- d. Exclua o `codecatalyst-stack-policy`.
- e. Exclua o `codecatalyst-deploy-role`.
- f. Exclua o `codecatalyst-build-role`.
- g. Exclua o `codecatalyst-stack-role`.

Neste tutorial, você aprendeu a implantar um aplicativo sem servidor como uma CloudFormation pilha usando um CodeCatalyst fluxo de trabalho e uma ação Deploy CloudFormation stack.

## Adicionando a ação “Deploy CloudFormation stack”

Use as instruções a seguir para adicionar a ação Deploy CloudFormation stack ao seu fluxo de trabalho.

### Visual

Para adicionar a ação “Implantar CloudFormation pilha” usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy CloudFormation stack e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Deploy CloudFormation stack. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
  - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
  - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- 10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [Ação “Implantar CloudFormation pilha” YAML](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
- 11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação “Implantar CloudFormation pilha” usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy CloudFormation stack e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Deploy CloudFormation stack. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
  - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
  - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- 10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [Ação “Implantar CloudFormation pilha” YAML](#).
- 11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Configurar reversões

Por padrão, se a ação Implantar CloudFormation pilha falhar, ela fará CloudFormation com que a pilha volte para o último estado estável conhecido. Você pode alterar o comportamento para que as reversões ocorram não apenas quando a ação falhar, mas também quando ocorrer um CloudWatch alarme específico da Amazon. Para obter mais informações sobre CloudWatch alarmes, consulte [Usando CloudWatch alarmes da Amazon no Guia CloudWatch](#) do usuário da Amazon.

Você também pode alterar o comportamento padrão para que CloudFormation não reverta a pilha quando a ação falhar.

Use as instruções a seguir para configurar reversões.

### Note

Você não pode iniciar uma reversão manualmente.

## Visual

### Antes de começar

1. Certifique-se de ter um [fluxo de trabalho](#) que inclua uma ação funcional do Deploy CloudFormation stack. Para obter mais informações, consulte [Implantação de uma pilha CloudFormation](#).
2. Na função especificada no campo opcional Função da pilha - da ação Implantar CloudFormation pilha, certifique-se de incluir a CloudWatchFullAccesspermissão. Para ter informações sobre como criar esse perfil com as permissões apropriadas, consulte [Etapa 2: criar AWS funções](#).

Para configurar alarmes de reversão para a ação “Implantar pilha” CloudFormation

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. Escolha sua ação Deploy CloudFormation stack.
8. No painel de detalhes, selecione Configuração.
9. Na parte inferior, expanda Avançado.
10. Em Monitorar alarme ARNs, escolha Adicionar alarme.
11. Insira informações nos seguintes campos.

- ARN do alarme

Especifique o Amazon Resource Name (ARN) de um CloudWatch alarme da Amazon para usar como gatilho de reversão. Por exemplo, `.arn:aws:cloudwatch::123456789012:alarm/MyAlarm` Você pode ter no máximo cinco gatilhos de reversão.



 Note

Se você especificar um ARN de CloudWatch alarme, também precisará configurar permissões adicionais para permitir o acesso à ação. CloudWatch Para obter mais informações, consulte [Configurar reversões](#).

- Tempo de monitoramento

Especifique um período de tempo, de 0 a 180 minutos, durante o qual CloudFormation monitora os alarmes especificados. O monitoramento começa após a implantação de todos os recursos da pilha. Se o alarme ocorrer dentro do tempo de monitoramento especificado, a implantação falhará e CloudFormation reverterá toda a operação da pilha.

Padrão: 0. CloudFormation monitora apenas os alarmes enquanto os recursos da pilha estão sendo implantados, não depois.

## YAML

Para configurar acionadores de reversão para a ação “Implantar pilha” CloudFormation

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Implantar pilha do CloudFormation . É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Adicione as propriedades `monitor-alarm-arns` e `monitor-timeout-in-minutes` no código YAML para adicionar gatilhos de reversão. Para ver uma explicação de cada propriedade, consulte [Ação “Implantar CloudFormation pilha” YAML](#).
8. Na função especificada na `role-arn` propriedade da ação Deploy CloudFormation stack, certifique-se de incluir a `CloudWatchFullAccess` permissão. Para ter informações sobre como criar esse perfil com as permissões apropriadas, consulte [Etapa 2: criar AWS funções](#).

## Visual

Para desativar as reversões da ação “Implantar pilha” CloudFormation

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Implantar pilha do CloudFormation . É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. Escolha sua ação Deploy CloudFormation stack.
8. No painel de detalhes, selecione Configuração.
9. Na parte inferior, expanda Avançado.
10. Ative a opção Desabilitar reversão.

## YAML

Para desativar as reversões da ação “Implantar pilha” CloudFormation

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Implantar pilha do CloudFormation . É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Adicione a propriedade `disable-rollback: 1` no código YAML para interromper as reversões. Para ver uma explicação dessa propriedade, consulte [Ação “Implantar CloudFormation pilha” YAML](#).

## Variáveis de “ CloudFormation pilha de implantação”

A ação Deploy CloudFormation stack produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave               | Valor                                                                                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| deployment-platform | O nome da plataforma de implantação.<br><br>Codificado para AWS:CloudFormation .                                                                                                       |
| region              | O código da região em Região da AWS que foi implantado durante a execução do fluxo de trabalho.<br><br>Exemplo: us-west-2                                                              |
| stack-id            | O nome do recurso da Amazon (ARN) da pilha implantada.<br><br>Exemplo: arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cfn-stack/6aad4380-100a-11ec-a10a-03b8a84d40df |

## Ação “Implantar CloudFormation pilha” YAML

A seguir está a definição YAML da ação Deploy CloudFormation stack. Para saber como usar essa ação, consulte [Implantação de uma pilha CloudFormation](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

**Note**

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
DeployCloudFormationStack:
 Identifier: aws/cfn-deploy@v1
 DependsOn:
 - build-action
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: DeployRole
 Inputs:
 Sources:
 - source-name-1
 Artifacts:
 - CloudFormation-artifact
 Configuration:
 name: stack-name
 region: us-west-2
 template: template-path
 role-arn: arn:aws:iam::123456789012:role/StackRole
 capabilities: CAPABILITY_IAM,CAPABILITY_NAMED_IAM,CAPABILITY_AUTO_EXPAND
 parameter-overrides: KeyOne=ValueOne,KeyTwo=ValueTwo | path-to-JSON-file
 no-execute-changeset: 1/0
 fail-on-empty-changeset: 1/0
```

```

 disable-rollback: 1/0
 termination-protection: 1/0
 timeout-in-minutes: minutes
 notification-arns: arn:aws:sns:us-east-1:123456789012:MyTopic,arn:aws:sns:us-east-1:123456789012:MyOtherTopic
 monitor-alarm-arns: arn:aws:cloudwatch::123456789012:alarm/MyAlarm,arn:aws:cloudwatch::123456789012:alarm/MyOtherAlarm
 monitor-timeout-in-minutes: minutes
 tags: '[{"Key": "MyKey1", "Value": "MyValue1"}, {"Key": "MyKey2", "Value": "MyValue2"}]'

```

## DeployCloudFormationStack

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: DeployCloudFormationStack\_nn.

Interface de usuário correspondente: guia Configuração/Nome de exibição da ação

Identifier

*(DeployCloudFormationStack/Identifier)*

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/cfn-deploy@v1.

Interface de usuário correspondente: rótulo do diagrama de fluxo de trabalho/  
DeployCloudFormationStack\_nn/aws/cfn-deploy@v1

DependsOn

*(DeployCloudFormationStack/DependsOn)*

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*DeployCloudFormationStack*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Type

(*DeployCloudFormationStack*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: Configuração tab/Advanced - opcional/Tipo de computação

## Fleet

(*DeployCloudFormationStack*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

UI correspondente: configuração tab/Advanced - opcional/frota de computação

## Timeout

(*DeployCloudFormationStack*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite em minutos - opcional

## Environment

(*DeployCloudFormationStack*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM

especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*DeployCloudFormationStack*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*DeployCloudFormationStack*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.



Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Name

(*DeployCloudFormationStack*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Role

(*DeployCloudFormationStack*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da função do IAM que a ação Deploy CloudFormation stack usa para acessar AWS e o CloudFormation serviço. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

**⚠ Warning**

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

**ℹ Note**

Na primeira vez em que o perfil for usado, use o caractere curinga a seguir na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A política de confiança personalizada a seguir:

**ℹ Note**

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração `tab/Environment/What está pronta?` *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' / Função Environment/account/role

**Inputs**

(*DeployCloudFormationStack*/Inputs)

## (Optional)

A seção `Inputs` define os dados que `DeployCloudFormationStack` precisa durante a execução de um fluxo de trabalho.

### Note

São permitidas no máximo quatro entradas (uma fonte e três artefatos) por ação de implantação da CloudFormation pilha.

Se você precisar consultar arquivos que residem em entradas diferentes (digamos, uma origem e um artefato), a entrada de origem será a entrada primária e o artefato será a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferirem das primárias. Para obter detalhes, consulte [Exemplo: referência de arquivos em vários artefatos](#).

Interface de usuário correspondente: guia Entradas

## Sources

(*DeployCloudFormationStack*/Inputs/Sources)

(Obrigatório se o seu AWS SAM modelo CloudFormation ou modelo estiver armazenado em um repositório de origem)

Se seu AWS SAM modelo CloudFormation ou modelo estiver armazenado em um repositório de origem, especifique o rótulo desse repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se o seu AWS SAM modelo CloudFormation ou modelo não estiver contido em um repositório de origem, ele deverá residir em um artefato gerado por outra ação ou em um bucket do Amazon S3.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*DeployCloudFormationStack*/Inputs/Artifacts)

(Obrigatório se seu AWS SAM modelo CloudFormation ou modelo estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se o AWS SAM modelo CloudFormation ou que você deseja implantar estiver contido em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se seu CloudFormation modelo não estiver contido em um artefato, ele deverá residir no seu repositório de origem ou em um bucket do Amazon S3.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

Configuration

(*DeployCloudFormationStack*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

name

(*DeployCloudFormationStack*/Configuration/name)

(Obrigatório)

Especifique um nome para a CloudFormation pilha que a ação Deploy CloudFormation stack cria ou atualiza.

Interface de usuário correspondente: guia Configuração/Nome da pilha

region

(*DeployCloudFormationStack*/Configuration/region)

(Obrigatório)

Especifique o Região da AWS local no qual a pilha será implantada. Para obter uma lista dos códigos das regiões, consulte [Endpoints regionais](#).

Interface de usuário correspondente: guia Configuração/Região da pilha

template

(*DeployCloudFormationStack*/Configuration/template)

## (Obrigatório)

Especifique o nome e o caminho para seu arquivo CloudFormation ou arquivo AWS SAM de modelo. O modelo pode estar no formato JSON ou YAML e pode residir em um repositório de origem, em um artefato de uma ação anterior ou em um bucket do Amazon S3. Se o arquivo de modelo estiver em um repositório ou artefato de origem, o caminho será relativo à origem ou à raiz do artefato. Se o modelo estiver em um bucket do Amazon S3, o caminho será o valor do URL do objeto do modelo.

Exemplos:

`./MyFolder/MyTemplate.json`

`MyFolder/MyTemplate.yml`

`https://MyBucket.s3.us-west-2.amazonaws.com/MyTemplate.yml`

### Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo do modelo para indicar em qual artefato ou origem encontrá-lo. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

Interface de usuário correspondente: guia Configuração/Modelo

role-arn

*(DeployCloudFormationStack/Configuration/role-arn)*

## (Obrigatório)

Especifique o Amazon Resource Name (ARN) da função da pilha. CloudFormation usa essa função para acessar e modificar recursos em sua pilha. Por exemplo:  
`arn:aws:iam::123456789012:role/StackRole`.

O perfil da pilha deve incluir:

- Uma ou mais políticas de permissões. As políticas dependem dos recursos que você tem na pilha. Por exemplo, se sua pilha inclui uma AWS Lambda função, você precisa adicionar permissões que concedam acesso ao Lambda. Se você seguiu o tutorial descrito em [Tutorial: Implantar uma aplicação sem servidor](#), ele inclui um procedimento intitulado [Para criar um perfil de pilha](#) que

lista as permissões que o perfil de pilha precisa se você estiver implantando uma pilha típica de aplicações sem servidor.

 Warning

Limite as permissões às exigidas pelo CloudFormation serviço para acessar os recursos em sua pilha. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

- A seguinte política de confiança:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "cloudformation.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

Se desejar, associe esse perfil à conexão da sua conta. Para saber mais sobre como associar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#). Se você não associar o perfil da pilha à conexão da conta, ele não aparecerá na lista suspensa Perfil da pilha no editor visual. No entanto, o ARN do perfil ainda poderá ser especificado no campo `role-arn` usando o editor YAML.

 Note

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões

de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: guia Configuração/Perfil da pilha - opcional

capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Obrigatório)

Especifique uma lista dos recursos do IAM necessários para permitir CloudFormation a criação de determinadas pilhas. Na maioria dos casos, você pode sair de capabilities com o valor padrão de CAPABILITY\_IAM, CAPABILITY\_NAMED\_IAM, CAPABILITY\_AUTO\_EXPAND.

Se você vir `##[error] requires capabilities: [capability-name]` nos logs da ação Implantar pilha do CloudFormation, consulte [Como faço para corrigir erros de recursos do IAM?](#) para ter informações sobre como corrigir o problema.

Para obter mais informações sobre os recursos do IAM, consulte Como [reconhecer recursos do IAM em CloudFormation modelos](#) no Guia do usuário do IAM.

Interface de usuário correspondente: guia Configuração/Avançada/Recursos

parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Optional)

Especifique parâmetros no seu CloudFormation ou no AWS SAM modelo que não tenham valores padrão ou para os quais você deseja especificar valores não padrão. Para obter mais informações sobre parâmetros, consulte [Parâmetros](#) no Guia AWS CloudFormation do usuário.

A propriedade parameter-overrides aceita:

- Um arquivo JSON contendo os parâmetros e valores.
- Uma lista separada por vírgulas de parâmetros e valores.

## Para especificar um arquivo JSON

1. Verifique se o arquivo JSON usa uma das seguintes sintaxes:

```
{
 "Parameters": {
 "Param1": "Value1",
 "Param2": "Value2",
 ...
 }
}
```

Ou...

```
[
 {
 "ParameterKey": "Param1",
 "ParameterValue": "Value1"
 },
 ...
]
```

(Existem outras sintaxes, mas elas não são suportadas CodeCatalyst no momento em que este artigo foi escrito.) Para obter mais informações sobre a especificação de CloudFormation parâmetros em um arquivo JSON, consulte [Sintaxe JSON compatível](#) na Referência de Comandos.AWS CLI

2. Especifique o caminho para o arquivo JSON usando um dos seguinte formatos:

- Se o arquivo JSON residir em um artefato de saída de uma ação anterior, use:

```
file:///artifacts/current-action-name/output-artifact-name/path-to-json-file
```

Consulte o Exemplo 1 para saber detalhes.

- Se o arquivo JSON residir no seu repositório de origem, use:

```
file:///sources/WorkflowSource/path-to-json-file
```

Consulte o Exemplo 2 para saber detalhes.



## Exemplo 1: o arquivo JSON reside em um artefato de saída

```
##My workflow YAML
...
Actions:
 MyBuildAction:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ParamArtifact
 Files:
 - params.json
 Configuration:
 ...
 MyDeployCFNStackAction:
 Identifier: aws/cfn-deploy@v1
 Configuration:
 parameter-overrides: file:///artifacts/MyDeployCFNStackAction/
ParamArtifact/params.json
```

## Exemplo 2: o arquivo JSON reside no repositório de origem, em uma pasta chamada my/folder

```
##My workflow YAML
...
Actions:
 MyDeployCloudFormationStack:
 Identifier: aws/cfn-deploy@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 parameter-overrides: file:///sources/WorkflowSource/my/folder/params.json
```

## Como usar uma lista de parâmetros separada por vírgula

- Adicione pares de nome-valor do parâmetro na propriedade `parameter-overrides` usando o seguinte formato:

*param-1=value-1,param-2=value-2*

Por exemplo, assumindo o seguinte CloudFormation modelo:

```
##My CloudFormation template

Description: My CloudFormation template

Parameters:
 InstanceType:
 Description: Defines the Amazon EC2 compute for the production server.
 Type: String
 Default: t2.micro
 AllowedValues:
 - t2.micro
 - t2.small
 - t3.medium

Resources:
...
```

... você pode definir a propriedade `parameter-overrides` da seguinte forma:

```
##My workflow YAML
...
Actions:
...
 DeployCloudFormationStack:
 Identifier: aws/cfn-deploy@v1
 Configuration:
 parameter-overrides: InstanceType=t3.medium,UseVPC=true
```

#### Note

Você pode especificar um nome de parâmetro sem um valor correspondente usando `undefined` como valor. Por exemplo:

`parameter-overrides: MyParameter=undefined`

O efeito é que, durante uma atualização da pilha, CloudFormation usa o valor do parâmetro existente para o nome do parâmetro fornecido.

Interface de usuário correspondente:

- Guia Configuração/Avançado/Substituições de parâmetros
- tab/Advanced/ParameterSubstituições de configuração/ Especifique substituições usando um arquivo
- tab/Advanced/ParameterSubstituições de configuração/ Especifique substituições usando um conjunto de valores

no-execute-changeset

(*DeployCloudFormationStack*/Configuration/no-execute-changeset)

(Optional)

Especifique se você CodeCatalyst deseja criar o conjunto de CloudFormation alterações e depois parar antes de executá-lo. Isso lhe dá a oportunidade de revisar o conjunto de alterações no CloudFormation console. Se você determinar que o conjunto de alterações parece bom, desative essa opção e execute novamente o fluxo de trabalho para que ele CodeCatalyst possa criar e executar o conjunto de alterações sem parar. O padrão é criar e executar o conjunto de alterações sem parar. Para obter mais informações, consulte o parâmetro CloudFormation [deploy](#) na Referência de AWS CLI Comandos. Para ter mais informações sobre a visualização de conjuntos de alterações, consulte [Visualização de um conjunto de alterações](#) no Guia do usuário do AWS CloudFormation .

Interface de usuário correspondente: guia Configuração/Avançado/Sem executar o conjunto de alterações

fail-on-empty-changeset

(*DeployCloudFormationStack*/Configuration/fail-on-empty-changeset)

(Optional)

Especifique se você deseja CodeCatalyst falhar na ação Deploy CloudFormation stack se o conjunto de CloudFormation alterações estiver vazio. (Se um conjunto de alterações estiver vazio, isso significa que não foram feitas alterações na pilha durante a implantação mais recente.) O padrão é permitir que a ação continue se o conjunto de alterações estiver vazio e retornar uma mensagem UPDATE\_COMPLETE mesmo que a pilha não tenha sido atualizada.

Para obter mais informações sobre essa configuração, consulte o parâmetro CloudFormation [deploy](#) na Referência de AWS CLI Comandos. Para ter mais informações sobre conjuntos de

alterações, consulte [Atualizar pilhas usando conjuntos de alterações](#) no Manual do usuário do AWS CloudFormation .

Interface de usuário correspondente: guia Configuração/Avançado/Falha no conjunto de alterações vazio

disable-rollback

(*DeployCloudFormationStack*/Configuration/disable-rollback)

(Optional)

Especifique se você CodeCatalyst deseja reverter a implantação da pilha se ela falhar. A reversão retorna a pilha ao último estado estável conhecido. O padrão é habilitar reversões. Para obter mais informações sobre essa configuração, consulte o parâmetro CloudFormation [deploy](#) na Referência de AWS CLI Comandos.

Para obter mais informações sobre como a ação Deploy CloudFormation stack lida com reversões, consulte. [Configurar reversões](#)

Para ter mais informações sobre como reverter uma pilha, consulte [Opções de falha de pilha](#) no Guia do usuário do AWS CloudFormation .

Interface de usuário correspondente: guia Configuração/Avançado/Desativar reversão

termination-protection

(*DeployCloudFormationStack*/Configuration/termination-protection)

(Optional)

Especifique se você deseja que a CloudFormation pilha Deploy adicione proteção de encerramento à pilha que está implantando. Se um usuário tentar excluir uma pilha com proteção contra encerramento ativada, a exclusão falhará e a pilha, incluindo o status dela, permanecerá inalterada. O padrão é desativar a proteção contra encerramento. Para ter mais informações, consulte [Proteção de uma pilha contra exclusão](#) no Guia do usuário do AWS CloudFormation .

Interface de usuário correspondente: guia Configuração/Avançado/Proteção contra encerramento

timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/timeout-in-minutes)

(Optional)

Especifique a quantidade de tempo, em minutos, que CloudFormation deve ser alocada antes de expirar as operações de criação da pilha e definir o status da pilha como `CREATE_FAILED`. Se o CloudFormation não puder criar a pilha inteira dentro do tempo alocado, além de não ser criada, a pilha será revertida.

Por padrão, não há tempo limite para criação da pilha. No entanto, recursos individuais podem ter seus próprios limites de tempo com base na natureza do serviço que implementam. Por exemplo, se um recurso individual em sua pilha expirar, a criação dela também expirará, ainda que o tempo limite especificado para a criação não tenha sido atingido.

UI correspondente: guia de configuração/avançado/tempo limite CloudFormation

notification-arns

(*DeployCloudFormationStack*/Configuration/notification-arns)

(Optional)

Especifique o ARN de um tópico do Amazon SNS para o qual você CodeCatalyst deseja enviar mensagens de notificação. Por exemplo, `.arn:aws:sns:us-east-1:111222333:MyTopic`. Quando a ação Deploy CloudFormation stack é executada, CodeCatalyst CloudFormation coordena-se para enviar uma notificação por CloudFormation evento que ocorre durante o processo de criação ou atualização da pilha. (Os eventos são visíveis na guia Eventos do CloudFormation console para a pilha.) Você pode definir até cinco tópicos. Para ter mais informações, consulte [O que é a Amazon SNS?](#)

UI correspondente: guia Configuração/Avançado/Notificação ARNs

monitor-alarm-arns

(*DeployCloudFormationStack*/Configuration/monitor-alarm-arns)

(Optional)

Especifique o Amazon Resource Name (ARN) de um CloudWatch alarme da Amazon para usar como gatilho de reversão. Por exemplo, `.arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Você pode ter no máximo cinco gatilhos de reversão.

 Note

Se você especificar um ARN de CloudWatch alarme, também precisará configurar permissões adicionais para permitir o acesso à ação. CloudWatch Para obter mais informações, consulte [Configurar reversões](#).

UI correspondente: guia de configuração/avançado/alarme de monitor ARNs

monitor-timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/monitor-timeout-in-minutes)

(Optional)

Especifique um período de tempo, de 0 a 180 minutos, durante o qual CloudFormation monitora os alarmes especificados. O monitoramento começa após a implantação de todos os recursos da pilha. Se o alarme ocorrer dentro do tempo de monitoramento especificado, a implantação falhará e CloudFormation reverterá toda a operação da pilha.

Padrão: 0. CloudFormation monitora apenas os alarmes enquanto os recursos da pilha estão sendo implantados, não depois.

Interface de usuário correspondente: guia Configuração/Avançado/Tempo de monitoramento

tags

(*DeployCloudFormationStack*/Configuration/tags)

(Optional)

Especifique as tags para anexar à sua CloudFormation pilha. As tags são pares de chave/valor arbitrários que você pode usar para identificar sua pilha em relação à finalidade, como alocação de custos. Para obter mais informações sobre o que são tags e como elas podem ser usadas, consulte Como [marcar seus recursos](#) no Guia do EC2 usuário da Amazon. Para obter mais informações sobre como marcar CloudFormation, consulte [Configuração das opções de CloudFormation pilha](#) no Guia do AWS CloudFormation usuário.

Uma chave pode ter caracteres alfanuméricos ou espaços e pode ter até 127 caracteres. Um valor pode ter caracteres alfanuméricos ou espaços e pode ter até 255 caracteres.

É possível adicionar até 50 tags exclusivas para cada pilha.

Interface de usuário correspondente: guia Configuração/Avançado/Tags

## Implantando um AWS CDK aplicativo com um fluxo de trabalho

Esta seção descreve como implantar um AWS Cloud Development Kit (AWS CDK) aplicativo em sua AWS conta usando um fluxo de trabalho. Para fazer isso, você deve adicionar a ação Implantação do AWS CDK ao seu fluxo de trabalho. A ação de AWS CDK implantação sintetiza e implanta seu AWS Cloud Development Kit (AWS CDK) aplicativo em AWS. Se seu aplicativo já existir em AWS, a ação o atualizará, se necessário.

Para obter informações gerais sobre como criar aplicativos usando o AWS CDK, consulte [O que é o AWS CDK?](#) no Guia do AWS Cloud Development Kit (AWS CDK) desenvolvedor.

### Tópicos

- [Quando usar a ação 'AWS CDK implantar'](#)
- [Como a ação 'AWS CDK implantar' funciona](#)
- [Versões da CLI do CDK usadas pela ação “Implantação do AWS CDK ”](#)
- [Imagem de tempo de execução usada pela ação 'AWS CDK implantar'](#)
- [Quantas pilhas a ação pode implantar?](#)
- [Exemplo: implantação de um aplicativo AWS CDK](#)
- [Adicionando a ação 'AWS CDK implantar'](#)
- [Variáveis de “Implantação do AWS CDK ”](#)
- [YAML da ação “Implantação do AWS CDK ”](#)

### Quando usar a ação 'AWS CDK implantar'

Use essa ação se você desenvolveu um aplicativo usando o AWS CDK e agora deseja implantá-lo automaticamente como parte do fluxo de trabalho de integração e entrega contínuas automatizadas (CI/CD). Por exemplo, talvez você queira implantar seu AWS CDK aplicativo automaticamente sempre que alguém mesclar uma pull request relacionada à fonte do seu AWS CDK aplicativo.

### Como a ação 'AWS CDK implantar' funciona

A Implantação do AWS CDK funciona da seguinte maneira:

1. [Em tempo de execução, se você especificou a versão 1.0.12 ou anterior da ação, a ação baixará a CLI CDK mais recente \(também chamada de Toolkit\) para a imagem AWS CDK do ambiente de tempo de execução. CodeCatalyst](#)

Se você especificou a versão 1.0.13 ou posterior, a ação vem junto com uma [versão específica](#) da CLI do CDK e, portanto, nenhum download ocorre.

2. A ação usa a CLI do CDK para executar o comando `cdk deploy`. Esse comando sintetiza e implanta seu AWS CDK aplicativo em. AWS Para ter mais informações sobre esse comando, consulte o tópico [Kit de ferramentas do AWS CDK \(comando cdk\)](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

## Versões da CLI do CDK usadas pela ação “Implantação do AWS CDK ”

A tabela a seguir mostra qual versão da CLI do CDK é usada por padrão por diferentes versões da ação Implantação do AWS CDK .

### Note

Talvez você consiga substituir o padrão. Para ter mais informações, consulte [CdkCliVersion](#) no [YAML da ação “Implantação do AWS CDK ”](#).

| Versão da ação “Implantação do AWS CDK ” | AWS CDK Versão CLI |
|------------------------------------------|--------------------|
| 1.0.0 – 1.0.12                           | mais recente       |
| 1.0.13 ou posterior                      | 2.99.1             |

## Imagem de tempo de execução usada pela ação 'AWS CDK implantar'

A tabela a seguir mostra as imagens do ambiente de execução CodeCatalyst usadas para executar diferentes versões da ação de AWS CDK implantação. As imagens incluem diferentes conjuntos de ferramentas pré-instaladas. Para obter mais informações, consulte [Imagens ativas](#).



**Note**

Recomendamos atualizar sua ação Implantação do AWS CDK para a versão 2.x para aproveitar as ferramentas mais recentes disponíveis na imagem de março de 2024. Para atualizar a ação, defina a propriedade `Identifier` como `aws/cdk-deploy@v2` no arquivo de definição de fluxo de trabalho. Para obter mais informações, consulte [YAML da ação “Implantação do AWS CDK”](#).

| Versão da ação “Implantação do AWS CDK” | Imagens de ambiente de runtime |
|-----------------------------------------|--------------------------------|
| 1.x                                     | Imagens de novembro de 2022    |
| 2.x                                     | Imagens de março de 2024       |

## Quantas pilhas a ação pode implantar?

A Implantação do AWS CDK pode implantar somente uma única pilha. Se seu AWS CDK aplicativo consistir em várias pilhas, você deverá criar uma pilha principal com pilhas aninhadas e implantar a principal usando essa ação.

## Exemplo: implantação de um aplicativo AWS CDK

O exemplo de fluxo de trabalho a seguir inclui a ação Implantação do AWS CDK, junto com a ação Inicialização do AWS CDK. O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Esse repositório contém seu AWS CDK aplicativo. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de AWS CDK bootstrap (CDKBootstrap) — No gatilho, a ação implanta a pilha de CDKToolkit bootstrap em AWS. Se a pilha do CDKToolkit já existir no ambiente, ela será atualizada se necessário; caso contrário, nada acontecerá e a ação será marcada como bem-sucedida.

- Uma ação de AWS CDK implantação (AWS CDK Deploy) — Ao concluir a ação de AWS CDK bootstrap, a ação de AWS CDK implantação sintetiza o código do seu AWS CDK aplicativo em um CloudFormation modelo e implanta a pilha definida no modelo em AWS

#### Note

O exemplo de fluxo de trabalho a seguir serve para fins ilustrativos e não funcionará sem configuração adicional.

#### Note

No código YAML a seguir, você pode omitir as seções `Connections:` se quiser. Se você omitir essas seções, deverá garantir que o perfil especificado no campo Perfil do IAM padrão em seu ambiente inclua as permissões e as políticas de confiança exigidas pelas ações Inicialização do AWS CDK e Implantação do AWS CDK. Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#). Para ter mais informações sobre as permissões e as políticas de confiança exigidas pelas ações Inicialização do AWS CDK e Implantação do AWS CDK, consulte a descrição da propriedade Role na [YAML da ação “Inicialização do AWS CDK”](#) e [YAML da ação “Implantação do AWS CDK”](#).

```
Name: codecatalyst-cdk-deploy-workflow
SchemaVersion: 1.0

Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 CDKBootstrap:
 Identifier: aws/cdk-bootstrap@v2
 Inputs:
 Sources:
 - WorkflowSource
 Environment:
 Name: codecatalyst-cdk-deploy-environment
 Connections:
```

```
- Name: codecatalyst-account-connection
 Role: codecatalyst-cdk-bootstrap-role
Configuration:
 Region: us-west-2

CDKDeploy:
 Identifier: aws/cdk-deploy@v2
 DependsOn:
 - CDKBootstrap
 Environment:
 Name: codecatalyst-cdk-deploy-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-cdk-deploy-role
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 StackName: my-app-stack
 Region: us-west-2
```

## Adicionando a ação 'AWS CDK implantar'

Use as instruções a seguir para adicionar a ação Implantação do AWS CDK ao seu fluxo de trabalho.

### Antes de começar

Antes de adicionar a ação Implantação do AWS CDK ao seu fluxo de trabalho, conclua as seguintes tarefas:

1. Tenha um AWS CDK aplicativo pronto. Você pode escrever seu AWS CDK aplicativo usando AWS CDK v1 ou v2, em qualquer linguagem de programação compatível com o. AWS CDK Verifique se os arquivos da aplicação AWS CDK estão disponíveis em:
  - Um [repositório de CodeCatalyst origem](#) ou
  - Um [artefato CodeCatalyst de saída](#) gerado por outra ação do fluxo de trabalho
2. Inicialize seu AWS ambiente. Para inicializar, você pode:
  - Usar um dos métodos descritos em [Como inicializar](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

- Usar a ação Inicialização do AWS CDK . Você pode adicionar essa ação no mesmo fluxo de trabalho da Implantação do AWS CDK ou em um diferente. Apenas garanta que a ação de inicialização seja executada pelo menos uma vez antes de executar a ação Implantação do AWS CDK para que os recursos necessários estejam disponíveis. Para obter mais informações sobre a ação de AWS CDK bootstrap, consulte [Inicializando um AWS CDK aplicativo com um fluxo de trabalho](#).

Para ter mais informações sobre inicialização, consulte [Inicialização](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

## Visual

Para adicionar a ação 'AWS CDK implantar' usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione Visual.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Implantação do AWS CDK e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Escolha Implantação do AWS CDK . A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [YAML da ação “Implantação do AWS CDK”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se a ação Implantação do AWS CDK falhar com um erro `npm install`, consulte [Como faço para corrigir erros de “instalação npm”?](#) para ter informações sobre como corrigir o erro.

## YAML

Para adicionar a ação 'AWS CDK implantar' usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Implantação do AWS CDK e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Implantação do AWS CDK . A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
  - (Opcional) Selecione Download para [visualizar o código-fonte da ação](#).
  - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- 10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [YAML da ação “Implantação do AWS CDK ”](#).
- 11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se a ação Implantação do AWS CDK falhar com um erro `npm install`, consulte [Como faço para corrigir erros de “instalação npm”?](#) para ter informações sobre como corrigir o erro.

## Variáveis de “Implantação do AWS CDK ”

A ação Implantação do AWS CDK produz e define as seguintes variáveis em runtime. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave    | Valor                                                                                                                                                                                                                                 |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stack-id | <p>O Amazon Resource Name (ARN) da pilha de AWS CDK aplicativos que foi implantada durante a execução do fluxo de trabalho.</p> <p>Exemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-app</code></p> |

| Chave                            | Valor                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | <code>-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>                                                                                                                                                                                                                                                                                                                           |
| <code>deployment-platform</code> | <p>O nome da plataforma de implantação.</p> <p>Codificado para <code>AWS::CloudFormation</code> .</p>                                                                                                                                                                                                                                                                              |
| <code>region</code>              | <p>O código da região em Região da AWS que foi implantado durante a execução do fluxo de trabalho.</p> <p>Exemplo: <code>us-west-2</code></p>                                                                                                                                                                                                                                      |
| <code>SKIP-DEPLOYMENT</code>     | <p>Um valor de <code>true</code> indica que a implantação da pilha de AWS CDK aplicativos foi ignorada durante a execução do fluxo de trabalho. Uma implantação da pilha será ignorada se não houver nenhuma alteração na pilha desde a última implantação.</p> <p>Essa variável só será produzida se o valor for <code>true</code>.</p> <p>Codificado para <code>true</code>.</p> |

| Chave                    | Valor                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CloudFormation variables | Além de gerar as variáveis listadas anteriormente, a ação de AWS CDK implantação também expõe as variáveis de CloudFormation saídas como variáveis de fluxo de trabalho para uso em ações subsequentes do fluxo de trabalho. Por padrão, a ação expõe somente as primeiras quatro (ou menos) CloudFormation variáveis encontradas. Para determinar quais estão expostas, execute a ação Implantação do AWS CDK uma vez e, depois, consulte a guia Variáveis da página de detalhes da execução. Se as variáveis listadas na guia Variáveis não forem as desejadas, você poderá configurar outras usando a propriedade YAML <code>CfnOutputVariables</code> . Para ter mais informações, consulte a descrição da propriedade de <a href="#">CfnOutputVariables</a> na <a href="#">YAML da ação “Implantação do AWS CDK”</a> . |

## YAML da ação “Implantação do AWS CDK”

Confira a seguir a definição de YAML da ação Implantação do AWS CDK. Para saber como usar essa ação, consulte [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.



```

The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
CDKDeploy_nn:
 Identifier: aws/cdk-deploy@v2
 DependsOn:
 - CDKBootstrap
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - artifact-name
 Outputs:
 Artifacts:
 - Name: cdk_artifact
 Files:
 - "cdk.out/**/*"
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
 Configuration:
 StackName: my-cdk-stack
 Region: us-west-2
 Tags: '{"key1": "value1", "key2": "value2"}'
 Context: '{"key1": "value1", "key2": "value2"}'
 CdkCliVersion: version
 CdkRootPath: directory-containing-cdk.json-file
 CfnOutputVariables: '["CfnOutputKey1", "CfnOutputKey2", "CfnOutputKey3"]'
 CloudAssemblyRootPath: path-to-cdk.out

```

## CDKDeploy

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: CDKDeploy\_nn.

Interface de usuário correspondente: guia Configuração/Nome da ação

Identifier

(*CDKDeploy*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

### Note

A especificação de `aws/cdk-deploy@v2` faz com que a ação seja executada na [imagem de março de 2024](#), que inclui ferramentas mais recentes, como Node.js 18. A especificação de `aws/cdk-deploy@v1` faz com que a ação seja executada na [imagem de novembro de 2022](#), que inclui ferramentas mais antigas, como Node.js 16.

Padrão: `aws/cdk-deploy@v2`.

Interface de usuário correspondente: Diagrama de fluxo de trabalho/CDKDeploy\_nn/rótulo `aws/cdk-deploy@v2`

DependsOn

(*CDKDeploy*/DependsOn)

(Optional)

Especifique uma ação ou um grupo de ação que deve ser executado para que a ação Implantação do AWS CDK seja executada. Recomendamos especificar a ação Inicialização do AWS CDK na propriedade DependsOn, da seguinte forma:

```
CDKDeploy:
 Identifier: aws/cdk-deploy@v2
 DependsOn:
 - CDKBootstrap
```

#### Note

A [inicialização é um pré-requisito obrigatório para a implantação de um aplicativo](#). AWS CDK Se você não incluir a ação Inicialização do AWS CDK no fluxo de trabalho, deverá encontrar outra maneira de implantar a pilha de inicialização do AWS CDK antes de executar a ação Implantação do AWS CDK. Para obter mais informações, consulte [Adicionando a ação 'AWS CDK implantar'](#) em [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#).

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*CDKDeploy*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Type

(*CDKDeploy*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: Configuração tab/Advanced - opcional/Tipo de computação

Fleet

(*CDKDeploy*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

UI correspondente: configuração tab/Advanced - opcional/frota de computação

Timeout

(*CDKDeploy*/Timeout)

(Obrigatório)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Inputs

(*CDKDeploy*/Inputs)

(Optional)

A seção Inputs define os dados que CDKDeploy precisa durante a execução de um fluxo de trabalho.

### Note

Somente uma entrada (uma fonte ou um artefato) é permitida para cada ação Implantação do AWS CDK .

Interface de usuário correspondente: guia Entradas

## Sources

(*CDKDeploy*/Inputs/Sources)

(Obrigatório se o AWS CDK aplicativo que você deseja implantar estiver armazenado em um repositório de origem)

Se seu AWS CDK aplicativo estiver armazenado em um repositório de origem, especifique o rótulo desse repositório de origem. A ação Implantação do AWS CDK sintetiza a aplicação nesse repositório antes de iniciar o processo de implantação. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seu AWS CDK aplicativo não estiver contido em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

Artifacts - input

(*CDKDeploy*/Inputs/Artifacts)

(Obrigatório se o AWS CDK aplicativo que você deseja implantar estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se seu AWS CDK aplicativo estiver contido em um artefato gerado por uma ação anterior, especifique esse artefato aqui. A ação de AWS CDK implantação sintetiza o aplicativo no artefato especificado em um CloudFormation modelo antes de iniciar o processo de implantação. Se seu AWS CDK aplicativo não estiver contido em um artefato, ele deverá residir no seu repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Entradas/Artefatos - opcional

Outputs

(*CDKDeploy*/Outputs)

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

Artifacts - output

(*CDKDeploy*/Outputs/Artifacts)

(Optional)

Especifique os artefatos gerados pela ação. Você pode referenciar esses artefatos como entrada em outras ações.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

## Interface de usuário correspondente: guia Saídas/Artefatos

### Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique o nome do artefato que conterá o CloudFormation modelo sintetizado pela ação de AWS CDK implantação em tempo de execução. O valor padrão é `cdk_artifact`. Se você não especificar um artefato, a ação sintetizará o modelo, mas não o salvará em um artefato. Pense em salvar o modelo sintetizado em um artefato para preservar um registro dele para fins de teste ou solução de problemas.

UI correspondente: artefato de `tab/Artifacts/Add saída/Nome do artefato de construção`

### Files

(*CDKDeploy*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `"cdk.out/**/*"` para incluir o modelo sintetizado CloudFormation do seu AWS CDK aplicativo.

#### Note

`cdk.out` é o diretório padrão no qual os arquivos sintetizados são salvos. Se você especificou um diretório de saída diferente de `cdk.out` no arquivo `cdk.json`, especifique esse diretório aqui em vez de `cdk.out`.

UI correspondente: `produz tab/Artifacts/Add artefatos/arquivos produzidos pela compilação`

### Environment

(*CDKDeploy*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM

especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*CDKDeploy*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*CDKDeploy*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.



Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/ Environment/account/role AWS '/' conexão de conta

Name

(*CDKDeploy*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/ Environment/account/role AWS '/' conexão de conta

Role

(*CDKDeploy*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Especifique o nome da função do IAM que a ação de AWS CDK implantação usa para acessar AWS e implantar a pilha de AWS CDK aplicativos. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

 Warning

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor0",
 "Effect": "Allow",
 "Action": [
 "cloudformation:DescribeStackEvents",
 "cloudformation:DescribeChangeSet",
 "cloudformation:DescribeStacks",
 "cloudformation:ListStackResources"
],
 "Resource": "*"
 },
 {
 "Sid": "VisualEditor1",
 "Effect": "Allow",
 "Action": "sts:AssumeRole",
 "Resource": "arn:aws:iam::111122223333:role/cdk-*"
 }
]
}
```

- A política de confiança personalizada a seguir:

 Note

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões

de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/ ' / Função Environment/account/role

## Configuration

(*CDKDeploy*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## StackName

(*CDKDeploy*/Configuration/StackName)

(Obrigatório)

O nome da sua pilha de AWS CDK aplicativos, conforme aparece no arquivo do ponto de entrada no diretório do seu AWS CDK aplicativo. bin O exemplo a seguir mostra o conteúdo de um arquivo de TypeScript ponto de entrada, com o nome da pilha destacado em *red italics*. Se o arquivo do ponto de entrada estiver em outra linguagem, ele terá uma aparência semelhante.

```
import * as cdk from 'aws-cdk-lib';
import { CdkWorkshopTypescriptStack } from '../lib/cdk_workshop_typescript-stack';

const app = new cdk.App();
new CdkWorkshopTypescriptStack(app, 'CdkWorkshopTypescriptStack');
```

Você pode especificar apenas uma pilha.

 Tip

Se você tiver várias pilhas, poderá criar uma pilha principal com pilhas aninhadas. Em seguida, você pode especificar a pilha principal nessa ação para implantar todas as pilhas.

Interface de usuário correspondente: guia Configuração/Nome da pilha

Region

(*CDKDeploy*/Configuration/Region)

(Optional)

Especifique Região da AWS no qual a pilha de AWS CDK aplicativos será implantada. Para obter uma lista dos códigos das regiões, consulte [Endpoints regionais](#).

Se você não especificar uma região, a ação de implantação será AWS CDK implantada na região especificada em seu AWS CDK código. Para ter mais informações, consulte [Ambientes](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

Interface de usuário correspondente: guia Configuração/Região

Tags

(*CDKDeploy*/Configuration/Tags)

(Optional)

Especifique as tags que você deseja aplicar aos AWS recursos na pilha de AWS CDK aplicativos. As tags são aplicadas à pilha em si, bem como aos recursos individuais na pilha. Para ter mais informações sobre marcação, consulte [Marcação](#), no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

UI correspondente: Configuração tab/Advanced - opcional/Tags

Context

(*CDKDeploy*/Configuration/Context)

(Optional)

Especifique contextos, na forma de pares de valores-chave, para associar à pilha de aplicativos. AWS CDK Para ter mais informações sobre contextos, consulte [Contextos de runtime](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

UI correspondente: Configuração tab/Advanced - opcional/Contexto

CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Optional)

Essa propriedade está disponível com a versão 1.0.13 ou posterior da ação Implantação do AWS CDK e a versão 1.0.8 ou posterior da ação Inicialização do AWS CDK .

Especifique um dos seguintes:

- A versão completa da interface de linha de AWS Cloud Development Kit (AWS CDK) comando (CLI) (também chamada de AWS CDK kit de ferramentas) que você deseja que essa ação use. Exemplo: 2.102.1. Considere especificar uma versão completa para garantir consistência e estabilidade ao criar e implantar a aplicação.

Ou

- latest. Pense em especificar latest para aproveitar os recursos e correções mais recentes da CLI do CDK.

A ação baixará a versão especificada (ou a versão mais recente) da AWS CDK CLI para a [imagem de CodeCatalyst compilação](#) e, em seguida, usará essa versão para executar os comandos necessários para implantar seu aplicativo CDK ou inicializar seu ambiente. AWS

Para ter uma lista das versões compatíveis da CLI do CDK que você pode usar, consulte [Versões do AWS CDK](#).

Se você omitir essa propriedade, a ação usará uma versão padrão da AWS CDK CLI descrita em um dos tópicos a seguir:

- [Versões da CLI do CDK usadas pela ação “Implantação do AWS CDK ”](#)
- [Versões da CLI do CDK usadas pela ação “Inicialização do AWS CDK ”](#)

UI correspondente: guia de configuração/versão AWS CDK CLI

## CdkRootPath

(*CDKDeploy*/Configuration/CdkRootPath)

(Optional)

O caminho para o diretório que contém o `cdk.json` arquivo do seu AWS CDK projeto. A ação Implantação do AWS CDK é executada nessa pasta e todas as saídas criadas pela ação serão adicionadas a esse diretório. Se não for especificada, a ação de AWS CDK implantação pressupõe que o `cdk.json` arquivo esteja na raiz do seu AWS CDK projeto.

Interface de usuário correspondente: guia Configuração/Diretório em que o `cdk.json` reside

## CfnOutputVariables

(*CDKDeploy*/Configuration/CfnOutputVariables)

(Optional)

Especifique quais `CfnOutput` construções no código do AWS CDK aplicativo você deseja expor como variáveis de saída do fluxo de trabalho. Depois, você pode referenciar as variáveis de saída do fluxo de trabalho em ações subsequentes no fluxo de trabalho. Para obter mais informações sobre variáveis em CodeCatalyst, consulte [Uso de variáveis em fluxos de trabalho](#).

Por exemplo, se o código do seu AWS CDK aplicativo tiver a seguinte aparência:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
 constructor(scope: Construct, id: string, props?: StackProps) {
 super(scope, id, props);
 const bucket = new s3.Bucket(this, 'amzn-s3-demo-bucket', {
 removalPolicy: RemovalPolicy.DESTROY,
 });
 new CfnOutput(this, 'bucketName', {
 value: bucket.bucketName,
 description: 'The name of the s3 bucket',
 exportName: 'amzn-s3-demo-bucket',
 });
 const table = new dynamodb.Table(this, 'todos-table', {
```

```

 partitionKey: {name: 'todoId', type: dynamodb.AttributeType.NUMBER},
 billingMode: dynamodb.BillingMode.PAY_PER_REQUEST,
 removalPolicy: RemovalPolicy.DESTROY,
 })
 new CfnOutput(this, 'tableName', {
 value: table.tableName,
 description: 'The name of the dynamodb table',
 exportName: 'myDynamoDbTable',
 });
 ...
}
}

```

... e a propriedade `CfnOutputVariables` for semelhante a:

```

Configuration:
 ...
 CfnOutputVariables: '["bucketName","tableName"]'

```

... a ação gera as seguintes variáveis de saída do fluxo de trabalho:

| Chave      | Valor             |
|------------|-------------------|
| bucketName | bucket.bucketName |
| tableName  | table.tableName   |

Depois, você pode referenciar as variáveis `bucketName` e `tableName` em ações subsequentes. Para saber como referenciar variáveis de saída do fluxo de trabalho em ações subsequentes, consulte [Referência a uma variável predefinida](#).

Se você não especificar nenhuma `CfnOutput` construção na `CfnOutputVariables` propriedade, a ação expõe as primeiras quatro (ou menos) variáveis de saída encontradas como variáveis CloudFormation de saída do fluxo de trabalho. Para obter mais informações, consulte [Variáveis de “Implantação do AWS CDK”](#).

#### Tip

Para obter uma lista de todas as variáveis de CloudFormation saída que a ação produz, execute o fluxo de trabalho que contém a ação de AWS CDK implantação uma vez e, em

seguida, consulte a guia Registros da ação. Os registros contêm uma lista de todas as variáveis CloudFormation de saída associadas ao seu AWS CDK aplicativo. Depois de saber quais são todas as CloudFormation variáveis, você pode especificar quais deseja converter em variáveis de saída do fluxo de trabalho usando a `CfnOutputVariables` propriedade.

Para obter mais informações sobre variáveis de CloudFormation saída, consulte a documentação da `CfnOutput` construção, disponível em [class CfnOutput \(construct\)](#) na Referência da AWS Cloud Development Kit (AWS CDK) API.

UI correspondente: guia de configuração/variáveis CloudFormation de saída

CloudAssemblyRootPath

(*CDKDeploy*/Configuration/CloudAssemblyRootPath)

(Optional)

Se você já sintetizou a pilha do seu AWS CDK aplicativo em uma montagem em nuvem (usando a `cdk synth` operação), especifique o caminho raiz do diretório de montagem em nuvem (`cdk.out`). O CloudFormation modelo localizado no diretório de montagem de nuvem especificado será implantado pela ação de AWS CDK implantação em você Conta da AWS usando o `cdk deploy --app` comando. Quando a opção `--app` está presente, a operação `cdk synth` não ocorre.

Se você não especificar um diretório de assembly de nuvem, a ação Implantação do AWS CDK executará o comando `cdk deploy` sem a opção `--app`. Sem a `--app` opção, a `cdk deploy` operação sintetizará (`cdk synth`) e implantará seu AWS CDK aplicativo no seu. Conta da AWS

Por que eu especificaria um conjunto de nuvem existente e sintetizado quando a ação “AWS CDK implantar” pode fazer a síntese em tempo de execução?

Talvez você queira especificar um assembly de nuvem sintetizado existente para:

- Garanta que exatamente o mesmo conjunto de recursos seja implantado sempre que a ação “AWS CDK implantar” for executada

Se você não especificar um assembly de nuvem, é possível que a ação Implantação do AWS CDK sintetize e implante arquivos diferentes, dependendo de quando ela for executada. Por exemplo, a ação Implantação do AWS CDK pode sintetizar um assembly de nuvem com um conjunto de dependências durante um estágio de teste e outro conjunto de dependências durante um estágio de produção (se essas dependências mudarem entre os estágios). Para garantir a paridade



exata entre o que foi testado e o que foi implantado, recomendamos sintetizar uma vez e depois usar o campo Caminho para o diretório de assembly de nuvem (editor visual) ou a propriedade `CloudAssemblyRootPath` (editor YAML) para especificar o assembly de nuvem já sintetizado.

- Usar gerenciadores de pacotes e ferramentas não padrão com a aplicação AWS CDK

Durante uma operação `synth`, a ação Implantação do AWS CDK tenta executar a aplicação usando ferramentas padrão, como `npm` ou `pip`. Se a ação não conseguir executar a aplicação usando essas ferramentas, a síntese não ocorrerá e a ação falhará. Para contornar esse problema, você pode especificar os comandos exatos necessários para executar seu aplicativo com êxito no `cdk.json` arquivo do AWS CDK aplicativo e, em seguida, sintetizar seu aplicativo usando um método que não envolva a AWS CDK ação de implantação. Depois que o assembly de nuvem for gerado, você poderá especificá-lo no campo Caminho para o diretório de assembly de nuvem (editor visual) ou na propriedade `CloudAssemblyRootPath` (editor YAML) da ação Implantação do AWS CDK.

Para obter informações sobre como configurar o `cdk.json` arquivo para incluir comandos para instalar e executar seu AWS CDK aplicativo, consulte [Especificação do comando do aplicativo](#).

Para ter informações sobre os comandos `cdk deploy` e `cdk synth`, bem como sobre a opção `--app`, consulte [Implantação de pilhas](#), [Sintetização de pilhas](#) e [Como ignorar a síntese](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK).

Para ter informações sobre assemblies de nuvem, consulte [Assembly de nuvem](#) na AWS Cloud Development Kit (AWS CDK) API Reference.

Interface de usuário correspondente: guia Configuração/Caminho para o diretório de assembly de nuvem

## Inicializando um AWS CDK aplicativo com um fluxo de trabalho

Esta seção descreve como inicializar um AWS CDK aplicativo usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar a ação Inicialização do AWS CDK ao seu fluxo de trabalho. A ação Inicialização do AWS CDK provisiona uma pilha de inicialização no ambiente da AWS usando o [modelo moderno](#). Se já existir uma pilha de inicialização, a ação a atualizará, se necessário. Ter uma pilha de bootstrap presente AWS é um pré-requisito para implantar um aplicativo. AWS CDK

Para ter mais informações sobre inicialização, consulte [Inicialização](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK).

## Tópicos

- [Quando usar a ação 'AWS CDK bootstrap'](#)
- [Como funciona a ação 'AWS CDK bootstrap'](#)
- [Versões da CLI do CDK usadas pela ação “Inicialização do AWS CDK ”](#)
- [Imagem de tempo de execução usada pela ação 'AWS CDK bootstrap'](#)
- [Exemplo: inicialização de um aplicativo AWS CDK](#)
- [Adicionando a ação 'AWS CDK bootstrap'](#)
- [variáveis de “Inicialização do AWS CDK ”](#)
- [YAML da ação “Inicialização do AWS CDK ”](#)

## Quando usar a ação 'AWS CDK bootstrap'

Use essa ação se você tiver um fluxo de trabalho que implanta um AWS CDK aplicativo e quiser implantar (e atualizar, se necessário) a pilha de bootstrap ao mesmo tempo. Nesse caso, você adicionaria a ação de AWS CDK bootstrap ao mesmo fluxo de trabalho que implanta seu AWS CDK aplicativo.

Não use essa ação se uma das seguintes opções se aplicar:

- Você já implantou uma pilha de inicialização usando outro mecanismo e deseja mantê-la intacta (sem atualizações).
- Você quer usar um [modelo de inicialização personalizado](#), que não é compatível com a ação Inicialização do AWS CDK .

## Como funciona a ação 'AWS CDK bootstrap'

A Inicialização do AWS CDK funciona da seguinte maneira:

1. [Em tempo de execução, se você especificou a versão 1.0.7 ou anterior da ação, a ação baixará a CLI CDK mais recente \(também chamada de AWS CDK Toolkit\) para a imagem de compilação. CodeCatalyst](#)

Se você especificou a versão 1.0.8 ou posterior, a ação vem junto com uma [versão específica](#) da CLI do CDK e, portanto, nenhum download ocorre.

2. A ação usa a CLI do CDK para executar o comando `cdk bootstrap`. Esse comando executa as tarefas de inicialização descritas no tópico [Inicialização](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK) .

## Versões da CLI do CDK usadas pela ação “Inicialização do AWS CDK ”

A tabela a seguir mostra qual versão da CLI do CDK é usada por padrão por diferentes versões da ação Inicialização do AWS CDK .

### Note

Talvez você consiga substituir o padrão. Para ter mais informações, consulte [CdkCliVersion](#) no [YAML da ação “Inicialização do AWS CDK ”](#).

| versão de ação “Inicialização do AWS CDK ” | AWS CDK Versão CLI |
|--------------------------------------------|--------------------|
| 1.0.0 – 1.0.7                              | mais recente       |
| 1.0.8 ou posterior                         | 2.99.1             |

## Imagem de tempo de execução usada pela ação 'AWS CDK bootstrap'

A tabela a seguir mostra as imagens do ambiente de execução CodeCatalyst usadas para executar diferentes versões da ação de AWS CDK bootstrap. As imagens incluem diferentes conjuntos de ferramentas pré-instaladas. Para obter mais informações, consulte [Imagens ativas](#).

### Note

Recomendamos atualizar sua ação Inicialização do AWS CDK para a versão 2.x para aproveitar as ferramentas mais recentes disponíveis na imagem de março de 2024. Para atualizar a ação, defina a propriedade `Identifier` como `aws/cdk-bootstrap@v2` no arquivo de definição de fluxo de trabalho. Para obter mais informações, consulte [YAML da ação “Implantação do AWS CDK ”](#).

| versão de ação “Inicialização do AWS CDK ” | Imagens de ambiente de runtime |
|--------------------------------------------|--------------------------------|
| 1.x                                        | Imagens de novembro de 2022    |
| 2.x                                        | Imagens de março de 2024       |

## Exemplo: inicialização de um aplicativo AWS CDK

Consulte o [Exemplo: implantação de um aplicativo AWS CDK](#) na [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#) para obter um fluxo de trabalho que inclui a ação AWS CDK bootstrap.

### Adicionando a ação 'AWS CDK bootstrap'

Use as instruções a seguir para adicionar a ação Inicialização do AWS CDK ao seu fluxo de trabalho.

#### Antes de começar

Antes de usar a ação Inicialização do AWS CDK , verifique se você tem uma aplicação AWS CDK pronta. A ação bootstrap sintetizará o AWS CDK aplicativo antes da inicialização. Você pode escrever a aplicação em qualquer linguagem de programação compatível com o AWS CDK.

Verifique se os arquivos AWS CDK do seu aplicativo estão disponíveis em:

- Um [repositório de CodeCatalyst origem](#) ou
- Um [artefato CodeCatalyst de saída](#) gerado por outra ação do fluxo de trabalho

#### Visual

Para adicionar a ação 'AWS CDK bootstrap' usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

5. Escolha Editar.
  6. Selecione Visual.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Inicialização do AWS CDK e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione Inicialização do AWS CDK . A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
  10. Nas guias Entradas, Configuração e Saídas, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [YAML da ação “Inicialização do AWS CDK”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se a ação Inicialização do AWS CDK falhar com um erro `npm install`, consulte [Como faço para corrigir erros de “instalação npm”?](#) para ter informações sobre como corrigir o erro.

## YAML

Para adicionar a ação 'AWS CDK bootstrap' usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação Inicialização do AWS CDK e selecione + para adicioná-la ao diagrama do fluxo de trabalho e abrir o painel de configuração.
10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [YAML da ação “Inicialização do AWS CDK”](#).
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se a ação Inicialização do AWS CDK falhar com um erro `npm install`, consulte [Como faço para corrigir erros de “instalação npm”?](#) para ter informações sobre como corrigir o erro.

## variáveis de “Inicialização do AWS CDK”

A ação Inicialização do AWS CDK produz e define as seguintes variáveis em runtime. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave               | Valor                                |
|---------------------|--------------------------------------|
| deployment-platform | O nome da plataforma de implantação. |

| Chave                        | Valor                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | Codificado para <code>AWS::CloudFormation</code> .                                                                                                                                                                                                                                                                                                                                   |
| <code>region</code>          | <p>O código de região em Região da AWS que a pilha de AWS CDK bootstrap foi implantada durante a execução do fluxo de trabalho.</p> <p>Exemplo: <code>us-west-2</code></p>                                                                                                                                                                                                           |
| <code>stack-id</code>        | <p>O Amazon Resource Name (ARN) da pilha de bootstrap implantada. AWS CDK</p> <p>Exemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code></p>                                                                                                                                                 |
| <code>SKIP-DEPLOYMENT</code> | <p>Um valor de <code>true</code> indica que a implantação da sua pilha de AWS CDK bootstrap foi ignorada durante a execução do fluxo de trabalho. Uma implantação da pilha será ignorada se não houver nenhuma alteração na pilha desde a última implantação.</p> <p>Essa variável só será produzida se o valor for <code>true</code>.</p> <p>Codificado para <code>true</code>.</p> |

## YAML da ação “Inicialização do AWS CDK ”

Confira a seguir a definição de YAML da ação Inicialização do AWS CDK . Para saber como usar essa ação, consulte [Iniciando um AWS CDK aplicativo com um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

**Note**

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
CDKBootstrapAction_nn:
 Identifier: aws/cdk-bootstrap@v2
 DependsOn:
 - action-name
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - artifact-name
 Outputs:
 Artifacts:
 - Name: cdk_bootstrap_artifacts
 Files:
 - "cdk.out/**/*"
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
 Configuration:
 Region: us-west-2
 CdkCliVersion: version
```



## CDKBootstrapAction

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: CDKBootstrapAction\_nn.

Interface de usuário correspondente: guia Configuração/Nome de exibição da ação

Identifier

(*CDKBootstrapAction*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

### Note

A especificação de `aws/cdk-bootstrap@v2` faz com que a ação seja executada na [imagem de março de 2024](#), que inclui ferramentas mais recentes, como Node.js 18. A especificação de `aws/cdk-bootstrap@v1` faz com que a ação seja executada na [imagem de novembro de 2022](#), que inclui ferramentas mais antigas, como Node.js 16.

Padrão: `aws/cdk-bootstrap@v2`.

Interface de usuário correspondente: Diagrama de fluxo de trabalho/CDKBootstrapAction\_nn/rótulo `aws/cdk-bootstrap@v2`

DependsOn

(*CDKBootstrapAction*/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*CDKBootstrapAction*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Type

(*CDKBootstrapAction*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: Configuração tab/Advanced - opcional/Tipo de computação

## Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

UI correspondente: configuração tab/Advanced - opcional/frota de computação

## Timeout

(*CDKBootstrapAction*/Timeout)

(Obrigatório)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Inputs

(*CDKBootstrapAction*/Inputs)

(Optional)

A seção `Inputs` define os dados que a ação Inicialização do AWS CDK precisa durante a execução de um fluxo de trabalho.

## Interface de usuário correspondente: guia Entradas

### Note

Somente uma entrada (uma fonte ou um artefato) é permitida para cada ação Inicialização do AWS CDK .

## Sources

(*CDKBootstrapAction*/Inputs/Sources)

(Obrigatório se seu AWS CDK aplicativo estiver armazenado em um repositório de origem)

Se seu AWS CDK aplicativo estiver armazenado em um repositório de origem, especifique o rótulo desse repositório de origem. A ação Inicialização do AWS CDK sintetiza a aplicação nesse repositório antes de iniciar o processo de inicialização. Atualmente, o único rótulo do repositório compatível é `WorkflowSource`.

Se seu AWS CDK aplicativo não estiver contido em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*CDKBootstrapAction*/Inputs/Artifacts)

(Obrigatório se seu AWS CDK aplicativo estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se seu AWS CDK aplicativo estiver contido em um artefato gerado por uma ação anterior, especifique esse artefato aqui. A ação AWS CDK bootstrap sintetiza o aplicativo no artefato especificado em um CloudFormation modelo antes de iniciar o processo de inicialização. Se a aplicação AWS CDK não estiver em um artefato, ela deverá residir em seu repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Entradas/Artefatos - opcional

Outputs

(*CDKBootstrapAction*/Outputs)

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

Artifacts - output

(*CDKBootstrapAction*/Outputs/Artifacts)

(Optional)

Especifique os artefatos gerados pela ação. Você pode referenciar esses artefatos como entrada em outras ações.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Saídas/Artefatos

Name

(*CDKBootstrapAction*/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique o nome do artefato que conterá o CloudFormation modelo sintetizado pela ação de AWS CDK bootstrap em tempo de execução. O valor padrão é `cdk_bootstrap_artifacts`. Se você não especificar um artefato, a ação sintetizará o modelo, mas não o salvará em um artefato. Pense em salvar o modelo sintetizado em um artefato para preservar um registro dele para fins de teste ou solução de problemas.

UI correspondente: artefato de tab/Artifacts/Add saída/Nome do artefato de construção

Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `"cdk.out/**/*"` para incluir o modelo sintetizado CloudFormation do seu AWS CDK aplicativo.

 Note

`cdk.out` é o diretório padrão no qual os arquivos sintetizados são salvos. Se você especificou um diretório de saída diferente de `cdk.out` no arquivo `cdk.json`, especifique esse diretório aqui em vez de `cdk.out`.

UI correspondente: produz `tab/Artifacts/Add artefatos/arquivos` produzidos pela compilação

Environment

(*CDKBootstrapAction*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*CDKBootstrapAction*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*CDKBootstrapAction*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em `Environment`.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da função do IAM que a ação AWS CDK bootstrap usa para acessar AWS e adicionar a pilha de bootstrap. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas apropriadas.

Você pode usar o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' '/' Função Environment/account/role

Configuration

(*CDKBootstrapAction*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.



Interface de usuário correspondente: guia Configuração

Region

(*CDKBootstrapAction*/Configuration/Region)

(Obrigatório)

Especifique Região da AWS no qual a pilha de bootstrap será implantada. Essa região deve corresponder àquela na qual seu AWS CDK aplicativo está implantado. Para obter uma lista dos códigos das regiões, consulte [Endpoints regionais](#).

Interface de usuário correspondente: guia Configuração/Região

CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

(Optional)

Essa propriedade está disponível com a versão 1.0.13 ou posterior da ação Implantação do AWS CDK e a versão 1.0.8 ou posterior da ação Inicialização do AWS CDK .

Especifique um dos seguintes:

- A versão completa da interface de linha de AWS Cloud Development Kit (AWS CDK) comando (CLI) (também chamada de AWS CDK kit de ferramentas) que você deseja que essa ação use. Exemplo: 2.102.1. Considere especificar uma versão completa para garantir consistência e estabilidade ao criar e implantar a aplicação.

Ou

- latest. Pense em especificar latest para aproveitar os recursos e correções mais recentes da CLI do CDK.

A ação baixará a versão especificada (ou a versão mais recente) da AWS CDK CLI para a [imagem de CodeCatalyst compilação](#) e, em seguida, usará essa versão para executar os comandos necessários para implantar seu aplicativo CDK ou inicializar seu ambiente. AWS

Para ter uma lista das versões compatíveis da CLI do CDK que você pode usar, consulte [Versões do AWS CDK](#).

Se você omitir essa propriedade, a ação usará uma versão padrão da AWS CDK CLI descrita em um dos tópicos a seguir:

- [Versões da CLI do CDK usadas pela ação “Implantação do AWS CDK ”](#)
- [Versões da CLI do CDK usadas pela ação “Inicialização do AWS CDK ”](#)

UI correspondente: guia de configuração/versão AWS CDK CLI

## Publicação de arquivos no Amazon S3 com um fluxo de trabalho

Esta seção descreve como publicar arquivos no Amazon S3 usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar a ação Publicação no Amazon S3 ao seu fluxo de trabalho. A ação Publicar do Amazon S3 copia arquivos de um diretório de origem para um bucket do Amazon S3. O diretório de origem pode residir em:

- Um [repositório de origem](#) ou
- Um [artefato de saída](#) gerado por outra ação do fluxo de trabalho

### Tópicos

- [Quando usar a ação “Publicação no Amazon S3”](#)
- [Imagem de runtime usada pela ação “Publicação no Amazon S3”](#)
- [Exemplo: publicação de arquivos no Amazon S3](#)
- [Adicionar a ação “Publicação no Amazon S3”](#)
- [Ação YAML de “Publicação no Amazon S3”](#)

## Quando usar a ação “Publicação no Amazon S3”

Use esta ação se:

- Você tiver um fluxo de trabalho que gera arquivos que você deseja armazenar no Amazon S3.

Por exemplo, você pode ter um fluxo de trabalho que cria um site estático que deseja hospedar no Amazon S3. Nesse caso, seu fluxo de trabalho incluiria uma [ação de criação](#) para compilar os arquivos HTML e de suporte do site e uma ação Publicação no Amazon S3 para copiar os arquivos no Amazon S3.

- Você tem um repositório de origem que contém arquivos que deseja armazenar no Amazon S3.

Por exemplo, você pode ter um repositório de origem com arquivos de origem da aplicação que deseja arquivar todas as noites no Amazon S3.

## Imagem de runtime usada pela ação “Publicação no Amazon S3”

A ação Publicação no Amazon S3 é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Exemplo: publicação de arquivos no Amazon S3

O exemplo de fluxo de trabalho a seguir inclui a ação Publicação no Amazon S3, junto com uma ação de criação. O fluxo de trabalho compila um site de documentação estática e depois o publica no Amazon S3, onde está hospedado. O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de criação (BuildDocs) – No gatilho, a ação compila um site de documentação estática (`mkdocs build`) e adiciona os arquivos HTML associados e os metadados de suporte a um artefato chamado MyDocsSite. Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).
- Uma ação Publicação no Amazon S3 (PublishToS3) – Ao concluir a ação de criação, essa ação copia o site no artefato MyDocsSite para o Amazon S3 para hospedagem.

### Note

O exemplo de fluxo de trabalho a seguir serve para fins ilustrativos e não funcionará sem configuração adicional.

### Note

No código YAML a seguir, você pode omitir a seção `Connections:` se quiser. Se você omitir esta seção, deverá garantir que o perfil especificado no campo Perfil do IAM padrão em seu ambiente inclua as permissões e políticas de confiança exigidas pela ação

Publicação no Amazon S3. Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#). Para ter mais informações sobre as permissões e as políticas de confiança exigidas pela ação Publicação no Amazon S3, consulte a descrição da propriedade [Role](#) em [Ação YAML de “Publicação no Amazon S3”](#).

Name: codecatalyst-s3-publish-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH
- Branches:
  - main

Actions:

BuildDocs:

Identifier: aws/build@v1

Inputs:

Sources:

- WorkflowSource

Configuration:

Steps:

- Run: echo BuildDocs started on `date`
- Run: pip install --upgrade pip
- Run: pip install mkdocs
- Run: mkdocs build
- Run: echo BuildDocs completed on `date`

Outputs:

Artifacts:

- Name: MyDocsSite

Files:

- "site/\*\*/\*"

PublishToS3:

Identifier: aws/s3-publish@v1

Environment:

Name: codecatalyst-s3-publish-environment

Connections:

- Name: codecatalyst-account-connection
- Role: codecatalyst-s3-publish-build-role

Inputs:

Sources:

- WorkflowSource

```
Artifacts:
 - MyDocsSite
Configuration:
 DestinationBucketName: amzn-s3-demo-bucket
 SourcePath: /artifacts/PublishToS3/MyDocSite/site
 TargetPath: my/docs/site
```

## Adicionar a ação “Publicação no Amazon S3”

Use as instruções a seguir para adicionar a ação Publicação no Amazon S3 ao seu fluxo de trabalho.

### Visual

Para adicionar a ação “Publicação no Amazon S3” usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione Visual.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Publicação no Amazon S3 e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione Publicação no Amazon S3. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

10. Nas guias Entradas, Configuração e Saídas, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [Ação YAML de “Publicação no Amazon S3”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação “Publicação no Amazon S3” usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione YAML.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Publicação no Amazon S3 e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione Publicação no Amazon S3. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [Ação YAML de “Publicação no Amazon S3”](#).
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Ação YAML de “Publicação no Amazon S3”

Confira a seguir a definição de YAML da ação Publicação no Amazon S3. Para saber como usar essa ação, consulte [Publicação de arquivos no Amazon S3 com um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.
```

```
Name: MyWorkflow
SchemaVersion: 1.0
Actions:
```

```
The action definition starts here.
```

```
S3Publish_nn:
 Identifier: aws/s3-publish@v1
 DependsOn:
 - build-action
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
```

Inputs:Sources:

- *source-name-1*

Artifacts:

- *artifact-name*

Variables:

- Name: *variable-name-1*  
Value: *variable-value-1*
- Name: *variable-name-2*  
Value: *variable-value-2*

Environment:

Name: *environment-name*

Connections:

- Name: *account-connection-name*  
Role: *iam-role-name*

Configuration:

SourcePath: *my/source*

DestinationBucketName: *amzn-s3-demo-bucket*

TargetPath: *my/target*

## S3Publish

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: S3Publish\_nn.

Interface de usuário correspondente: guia Configuração/Nome da ação

Identifier

(*S3Publish*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/s3-publish@v1.



Interface de usuário correspondente: Diagrama de fluxo de trabalho/S3Publish\_nn/rótulo aws/s3-publish@v1

DependsOn

(*S3Publish*/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*S3Publish*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Type

(*S3Publish*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

Interface de usuário correspondente: guia Configuração/Tipo de computação

Fleet

(*S3Publish*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

Interface de usuário correspondente: guia Configuração/Frota de computação

Timeout

(*S3Publish*/Timeout)

(Obrigatório)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Inputs

(*S3Publish*/Inputs)

(Optional)

A seção Inputs define os dados que S3Publish precisa durante a execução de um fluxo de trabalho.

### Note

São permitidas no máximo quatro entradas (uma origem e três artefatos) para cada ação Implantação do AWS CDK . As variáveis não são contadas nesse total.

Se você precisar consultar arquivos que residem em entradas diferentes (digamos, uma origem e um artefato), a entrada de origem será a entrada primária e o artefato será a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferirem das primárias. Para obter detalhes, consulte [Exemplo: referência de arquivos em vários artefatos](#).

Interface de usuário correspondente: guia Entradas

## Sources

(*S3Publish*/Inputs/Sources)

(Obrigatório se os arquivos que você deseja publicar no Amazon S3 estiverem armazenados em um repositório de origem)

Se os arquivos que você deseja publicar no Amazon S3 estiverem armazenados em um repositório de origem, especifique o rótulo desse repositório de origem. Atualmente, o único rótulo compatível é WorkflowSource.

Se os arquivos que você deseja publicar no Amazon S3 não estiverem contidos em um repositório de origem, eles devem residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*S3Publish*/Inputs/Artifacts)

(Obrigatório se os arquivos que você deseja publicar no Amazon S3 estiverem armazenados em um [artefato de saída](#) de uma ação anterior)

Se os arquivos que você deseja publicar no Amazon S3 estiverem em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se as aplicações não estiverem em um artefato, elas deverão residir no repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

## Variables - input

(*S3Publish*/Inputs/Variables)

(Optional)

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Variáveis - opcional

## Environment

(*S3Publish*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM

especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*S3Publish*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

Connections

(*S3Publish*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Name

(*S3Publish*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

Role

(*S3Publish*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da função do IAM que a ação de publicação do Amazon S3 usa para acessar AWS e copiar arquivos para o Amazon S3. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

**⚠ Warning**

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

**JSON**

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor0",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject",
 "s3:ListBucket",
 "s3:DeleteObject"
],
 "Resource": [
 "arn:aws:s3:::bucket-name",
 "arn:aws:s3:::bucket-name/*"
]
 }
]
}
```

- A política de confiança personalizada a seguir:

**📘 Note**

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' / Função Environment/account/role

## Configuration

(*S3Publish*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## SourcePath

(*S3Publish*/Configuration/SourcePath)

(Obrigatório)

Especifique o nome e o caminho de um diretório ou arquivo que você deseja publicar no Amazon S3. O diretório ou arquivo pode residir em um repositório de origem ou em um artefato de uma ação anterior e é relativo ao repositório de origem ou à raiz do artefato.

Exemplos:

A especificação `./myFolder/` copia o conteúdo de `/myFolder` para o Amazon S3 e preserva a estrutura de diretórios subjacente.

A especificação `./myFolder/myfile.txt` copia somente `myfile.txt` para o Amazon S3. (A estrutura de diretórios foi removida.)

Não é possível usar curingas.

### Note

Talvez seja necessário adicionar um prefixo ao diretório ou caminho do arquivo para indicar em qual artefato ou origem encontrá-lo. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).



Interface de usuário correspondente: guia Configuração/Caminho de origem

DestinationBucketName

(*S3Publish*/Configuration/DestinationBucketName)

(Obrigatório)

Especifique o nome do bucket do Amazon S3 em que você deseja publicar os arquivos.

Interface de usuário correspondente: guia Configuração/Bucket de destino - opcional

TargetPath

(*S3Publish*/Configuration/TargetPath)

(Optional)

Especifique o nome e o caminho do diretório no Amazon S3 em que você deseja publicar seus arquivos. Se o diretório não existir, ele será criada. O caminho do diretório não deve incluir o nome do bucket.

Exemplos:

myS3Folder

./myS3Folder/myS3Subfolder

Interface de usuário correspondente: guia Configuração/Diretório de destino - opcional

## Implantação em e Contas da AWS VPCs

Usando [CodeCatalyst fluxos de trabalho](#), você pode implantar aplicativos e outros recursos para atingir Conta da AWS e Amazon VPCs na AWS nuvem. Para habilitar essas implantações, você deve configurar CodeCatalyst ambientes.

Um CodeCatalyst ambiente, que não deve ser confundido com um [ambiente de desenvolvimento](#), define o Amazon VPC de destino Conta da AWS e opcional ao qual um CodeCatalyst [fluxo de trabalho](#) se conecta. Um ambiente também define a [função do IAM](#) que um fluxo de trabalho precisa para acessar os AWS serviços e recursos na conta de destino.

É possível configurar vários ambientes e atribuir a eles nomes, como desenvolvimento, teste, preparação e produção. Quando você implanta nesses ambientes, as informações sobre as implantações aparecem nas CodeCatalyst guias Atividade de implantação e Destinos de implantação no ambiente.

## Como começo a usar ambientes?

As etapas de alto nível para adicionar e usar um CodeCatalyst ambiente são as seguintes:

1. No seu CodeCatalyst espaço, conecte uma ou mais AWS contas. Durante esse processo, adicione os perfis do IAM que seu fluxo de trabalho exige para acessar recursos na sua Conta da AWS. Para obter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).
2. Em seu CodeCatalyst projeto, crie um ambiente que inclua uma das funções Conta da AWS e IAM da etapa 1. Para obter mais informações, consulte [Criar um ambiente](#).
3. Em seu CodeCatalyst projeto, em um fluxo de trabalho, adicione uma [ação](#) que aponte para o ambiente que você criou na etapa 2. Para obter mais informações, consulte [Adição de uma ação a um fluxo de trabalho](#).

Agora você configurou um ambiente. A ação agora pode implantar recursos na Conta da AWS especificada no ambiente.

### Note

Você também pode adicionar uma Amazon VPC ao ambiente. Para obter mais informações, consulte [Adicionar conexões VPC a um espaço](#) no Guia de CodeCatalyst Administração e [Associação de uma VPC a um ambiente](#).

## Podem existir vários ambientes em um único fluxo de trabalho?

Sim. Se um fluxo de trabalho incluir várias ações, cada uma dessas ações poderá ser atribuída a um ambiente. Por exemplo, você pode ter um fluxo de trabalho que inclua duas ações de implantação, em que uma é atribuída a um ambiente my-staging-environment e outra a um ambiente my-production-environment.

## Quais ações de fluxo de trabalho oferecem suporte aos ambientes?

Qualquer ação de fluxo de trabalho que implanta recursos na AWS nuvem ou se comunica com os AWS serviços por outros motivos (como monitoramento e geração de relatórios) oferece suporte aos ambientes.

## Quais ações permitem que suas informações de implantação sejam exibidas CodeCatalyst?

Das ações de fluxo de trabalho que oferecem suporte a ambientes, apenas algumas oferecem suporte para que suas informações de implantação sejam exibidas nas páginas Atividade de implantação e Destinos de implantação do CodeCatalyst console.

As ações de fluxo de trabalho a seguir permitem que as informações de implantação sejam exibidas:

- Deploy CloudFormation stack — Para obter mais informações, consulte [Implantação de uma pilha CloudFormation](#)
- Implantar no Amazon ECS: para ter mais informações, consulte [Implantação no Amazon ECS com um fluxo de trabalho](#)
- Implantar no cluster do Kubernetes: para ter mais informações, consulte [Implantar no Amazon EKS com um fluxo de trabalho](#)
- AWS CDK implantar — Para obter mais informações, consulte [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#)

## Regiões do compatíveis

A página Ambientes pode exibir recursos em qualquer região da AWS .

## Um ambiente é obrigatório?

Um ambiente é obrigatório se a ação do fluxo de trabalho à qual ele está atribuído implantar recursos na AWS nuvem ou se comunicar com os AWS serviços por outros motivos (como monitoramento e geração de relatórios).

Por exemplo, se você tem uma ação de criação que cria um aplicativo, mas não precisa se comunicar com sua VPC Conta da AWS ou com a Amazon VPC, não é necessário atribuir um ambiente à ação. Se, no entanto, a ação de criação enviar registros para o CloudWatch serviço da Amazon em seu Conta da AWS, a ação deverá ter um ambiente atribuído.

## Tópicos

- [Criar um ambiente](#)
- [Associação de um ambiente a uma ação](#)
- [Associação de uma VPC a um ambiente](#)
- [Associando um Conta da AWS a um ambiente](#)
- [Alteração do perfil do IAM de uma ação](#)

## Criar um ambiente

Use as instruções a seguir para criar um ambiente que você possa associar posteriormente a uma ação de fluxo de trabalho.

Antes de começar

Você precisará do seguinte:

- Um CodeCatalyst espaço. Para obter mais informações, consulte [Configuração e login no CodeCatalyst](#).
- Um CodeCatalyst projeto. Para obter mais informações, consulte [Criar um projeto com um esquema](#).
- Uma conexão de AWS conta que inclui as funções do IAM que sua ação de fluxo de trabalho precisará acessar AWS. Para ter informações sobre a criação de uma conexão de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Você pode usar no máximo uma conexão de conta por ambiente.

### Note

Você pode criar um ambiente sem uma conexão de conta. No entanto, você precisará voltar e adicionar a conexão posteriormente.

- Uma das seguintes CodeCatalyst funções:
  - Administrador do espaço
  - Administrador do projeto
  - Contributor (Colaborador)

 Note

Se você tiver o Perfil de colaborador, poderá criar um ambiente, mas não poderá associá-lo a uma conexão de Conta da AWS . Você precisará pedir a alguém com a função de administrador do Space ou administrador do projeto que associe o ambiente a uma Conta da AWS conexão.

Para ter mais informações sobre permissões e perfis, consulte [Concessão de permissões de projeto aos usuários](#).

Para criar um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.
4. Em Nome do ambiente, insira um nome, como **Production** ou **Staging**.
5. Em Tipo de ambiente, selecione uma das seguintes opções:
  - Não produção: um ambiente em que você pode testar a aplicação para garantir que ela esteja funcionando conforme o esperado antes de colocá-la em produção.
  - Produção: um ambiente “ativo” que está disponível publicamente e hospeda a aplicação finalizada.

Se você selecionar Produção, um selo Produção aparecerá na interface de usuário ao lado de todas as ações às quais o ambiente esteja associado. O selo ajuda você a ver rapidamente quais ações estão sendo implantadas na produção. Além da aparência do selo, não há diferenças entre ambientes de produção e não produção.

6. (Opcional) Em Descrição, insira uma descrição como **Production environment for the hello-world app**.
7. Em Conta da AWS conexão - opcional, escolha a conexão de AWS conta que você deseja associar a esse ambiente. As ações de fluxo de trabalho atribuídas a esse ambiente poderão se conectar à Conta da AWS associada. Para obter mais informações sobre a criação de Conta da AWS conexões em CodeCatalyst, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Se a Conta da AWS conexão que você deseja usar não estiver listada, talvez seja porque ela não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

8. Em Perfil do IAM padrão, escolha o perfil do IAM que você deseja associar a esse ambiente. As ações de fluxo de trabalho atribuídas a esse ambiente herdarão essa função do IAM e poderão usá-la para se conectar aos serviços e recursos em seu Conta da AWS.

Se você precisar atribuir o ambiente a várias ações e essas ações precisarem de perfis do IAM diferentes do padrão especificado aqui, você poderá especificar os diferentes perfis na guia Configuração de cada ação, usando a opção Alternar perfil. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Se a função do IAM que você deseja usar como padrão não estiver listada, pode ser porque você ainda não a adicionou à sua Conta da AWS conexão. Para adicionar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#).

9. (Opcional) Em Conexão VPC, escolha uma conexão VPC que você deseja associar a esse ambiente. Para obter mais informações sobre a criação de conexões VPC, consulte Gerenciando [Amazon Virtual Private Clouds](#) no Amazon CodeCatalyst Administrator Guide.

Se a conexão VPC que você deseja usar não estiver listada, talvez seja porque ela inclui uma Conta da AWS conexão que não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

10. Escolha Criar ambiente. CodeCatalyst cria um ambiente vazio.

## Próximas etapas

- Agora que criou um ambiente, você pode associá-lo a uma ação de fluxo de trabalho. Para obter mais informações, consulte [Associação de um ambiente a uma ação](#).

## Associação de um ambiente a uma ação

Quando você associa um ambiente a uma [ação de fluxo de trabalho compatível](#), a função padrão do IAM do ambiente e a Amazon VPC opcional são atribuídas à ação. Conta da AWS A ação pode se conectar e implantar na Conta da AWS usando o perfil do IAM e também se conectar à Amazon VPC opcional.

Use as instruções a seguir para associar um ambiente a uma ação.

### Etapa 1: associar o ambiente a uma ação de fluxo de trabalho

Use o procedimento a seguir para associar um ambiente a uma ação de fluxo de trabalho.

#### Visual

Para associar um ambiente a uma ação de fluxo de trabalho usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, escolha uma ação compatível com ambientes. Para obter mais informações, consulte [Quais ações permitem que suas informações de implantação sejam exibidas CodeCatalyst?](#)
8. Escolha a guia Configuração e especifique as informações no campo Ambiente, da seguinte forma.

#### Ambiente

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

#### Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

9. (Opcional) Altere o perfil do IAM associado à ação. Você talvez queira alterar o perfil se ele contiver o conjunto errado de permissões para a ação.

Para alterar o perfil:

1. No What's in **my-environment**? caixa e escolha o ícone de elipse vertical ( ).

⋮

2. Escolha uma das seguintes opções:

- Alternar perfil. Escolha essa opção para alterar o perfil do IAM usado por essa ação, e somente essa ação. Outras ações continuam usando o perfil do IAM padrão especificado no ambiente associado. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).
- Editar ambiente. Escolha essa opção para alterar o perfil do IAM padrão listado em seu ambiente. Quando você escolhe essa opção, sua ação — e qualquer outra ação associada ao mesmo ambiente — começa a usar o novo perfil do IAM padrão.



#### Important

Tenha cuidado ao atualizar o perfil do IAM padrão. Alterar o perfil pode levar a falhas de ação se as permissões no perfil não forem suficientes para todas as ações que compartilham o ambiente.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para associar um ambiente a uma ação de fluxo de trabalho usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.



4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Na ação do fluxo de trabalho que você deseja associar a um ambiente, adicione um código semelhante ao seguinte:

```
action-name:
 Environment:
 Name: environment-name
```

Para ter mais informações, consulte o tópico [Tipos de ação](#). Este tópico tem links para a documentação de cada ação, incluindo a referência YAML.

8. (Opcional) Se você quiser que a ação use um perfil diferente do perfil do IAM padrão listado no ambiente, adicione uma seção `Connections:` que inclua o perfil que você deseja usar. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Etapa 2: preencher a página de atividades de implantação

Depois de associar um ambiente a uma ação de fluxo de trabalho, você pode preencher as páginas Atividade de implantação e Destino de implantação na seção Ambientes do CodeCatalyst console com informações de implantação. Use as instruções a seguir para preencher essas páginas.

### Note

Apenas algumas ações permitem que suas informações de implantação sejam exibidas no CodeCatalyst console. Para obter mais informações, consulte [Quais ações permitem que suas informações de implantação sejam exibidas CodeCatalyst?](#).

## Para adicionar informações de implantação ao CodeCatalyst

1. Se a execução de um fluxo de trabalho não foi iniciada automaticamente quando você confirmou as alterações em [Etapa 1: associar o ambiente a uma ação de fluxo de trabalho](#), inicie manualmente a execução da seguinte forma:
  - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - b. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  - c. Escolha Executar.

A execução do fluxo de trabalho inicia uma nova implantação, o que faz com que CodeCatalyst adicione as informações de implantação ao CodeCatalyst console.

2. Verifique se a atividade de implantação foi adicionada ao CodeCatalyst console:
  - a. No painel de navegação, selecione CI/CD e Ambientes.
  - b. Escolha seu ambiente (por exemplo, Production).
  - c. Escolha a guia Atividade de implantação e verifique se uma implantação aparece com o Status BEM-SUCEDIDO. Isso indica que a execução de um fluxo de trabalho implantou os recursos da aplicação.
  - d. Escolha a guia Destinos de implantação e verifique se os recursos da aplicação aparecem.

## Associação de uma VPC a um ambiente

Quando uma ação é configurada com um ambiente que tem uma conexão VPC, a ação é executada conectada à VPC, aderindo às regras de rede e aos recursos de acesso especificados pela VPC associada. A mesma conexão VPC pode ser usada por um ou mais ambientes.

Use as instruções a seguir para associar uma conexão VPC a um ambiente.

### Para associar uma conexão VPC a um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.

4. Escolha seu ambiente (por exemplo, `Production`).
5. Escolha a guia Propriedades do ambiente.
6. Selecione Gerenciar conexão VPC, escolha a conexão VPC desejada e selecione Confirmar. Isso associa a conexão VPC selecionada a esse ambiente.

 **Note**

Se a conexão VPC que você deseja usar não estiver listada, talvez seja porque ela inclui uma Conta da AWS conexão que não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

Para obter mais informações, consulte [Gerenciando Amazon Virtual Private Clouds](#) no Guia CodeCatalyst do Administrador.

## Associando um Conta da AWS a um ambiente

Use as instruções a seguir para associar um Conta da AWS a um ambiente. Quando você associa um Conta da AWS a um ambiente, as ações do fluxo de trabalho atribuídas ao ambiente poderão se conectar ao Conta da AWS.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Antes de começar

Você precisará do seguinte:

- Uma conexão de AWS conta que inclui as funções do IAM que sua ação de fluxo de trabalho precisará acessar AWS. Para ter informações sobre a criação de uma conexão de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Você pode usar no máximo uma conexão de conta por ambiente.
- Uma das seguintes CodeCatalyst funções: administrador do espaço ou administrador do projeto. Para obter mais informações, consulte [Concessão de permissões de projeto aos usuários](#).

Para associar um Conta da AWS a um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.
4. Escolha seu ambiente (por exemplo, Production).
5. Selecione Editar ambiente.
6. Em Propriedades do ambiente, na lista suspensa Conexão da Conta da AWS - opcional, escolha a Conta da AWS desejada.

Se a Conta da AWS conexão que você deseja usar não estiver listada, talvez seja porque ela não é permitida em seu projeto. Para obter mais informações, consulte [Configuração de conexões de contas restritas a projetos](#) no Amazon CodeCatalyst Administrator Guide.

7. Em Perfil do IAM padrão, escolha o perfil do IAM que você deseja associar a esse ambiente. As ações de fluxo de trabalho atribuídas a esse ambiente herdarão essa função do IAM e poderão usá-la para se conectar aos serviços e recursos em seu Conta da AWS.

Se a função do IAM que você deseja usar como padrão não estiver listada, pode ser porque você ainda não a adicionou à sua Conta da AWS conexão. Para adicionar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#).

## Alteração do perfil do IAM de uma ação

Por padrão, quando você associa um [ambiente](#) a uma [ação](#) de fluxo de trabalho, a ação herda o perfil do IAM padrão especificado no ambiente. Você pode alterar esse comportamento para que a ação use um perfil diferente. Talvez você queira que uma ação use um perfil diferente se o perfil do IAM padrão não tiver as permissões de que a ação precisa para operar na nuvem da AWS .

Para atribuir um perfil do IAM diferente a uma ação, você pode usar a opção Alternar perfil no editor visual ou a propriedade `Connections:` no editor YAML. O novo perfil substitui o perfil do IAM padrão especificado no ambiente, permitindo que você mantenha o perfil do IAM padrão como está. Talvez você queira manter o perfil do IAM padrão como está se houver outras ações que o usem.

Use as instruções a seguir para configurar uma ação para usar um perfil do IAM diferente do especificado no ambiente.

### Visual

Para atribuir um perfil do IAM diferente a uma ação (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Escolha a caixa que representa a ação cujo perfil do IAM você deseja atualizar.
7. Escolha a guia Configuração.
8. No What's in **my-environment**? caixa, escolha o ícone de elipse vertical ( ).  
⋮
9. Selecione Alternar perfil.
10. Na caixa de diálogo Alternar perfil, na lista suspensa Perfil do IAM, escolha o perfil do IAM que você deseja que a ação use. Esse perfil substituirá o perfil do IAM padrão no ambiente. Se o perfil que você deseja usar não estiver na lista, adicione-o ao seu espaço. Para obter mais informações, consulte [Adicionar perfis do IAM às conexões da conta](#).

A função escolhida agora aparece na seção What's in **my-environment**? caixa junto com um selo Definido no fluxo de trabalho. O perfil também aparece no arquivo de definição do fluxo de trabalho, na seção Connections:.

11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para atribuir um perfil do IAM diferente a uma ação (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.

6. Selecione YAML.
7. Na ação do fluxo de trabalho em que você deseja usar um perfil do IAM diferente, adicione uma seção `Connections`: semelhante à seguinte:

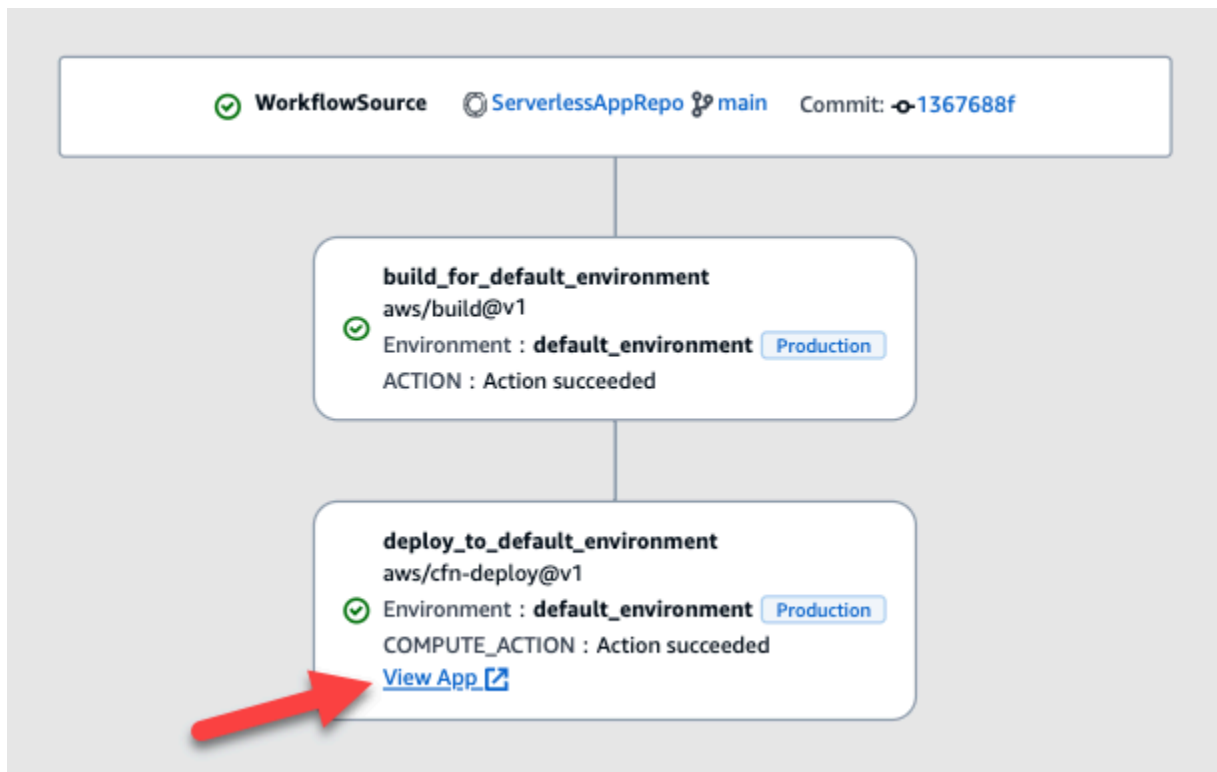
```
action-name:
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
```

No código anterior, *account-connection-name* substitua pelo nome da [conexão da conta](#) que contém a função do IAM e *iam-role-name* substitua pelo nome da função do IAM que você deseja que a ação use. Esse perfil substituirá o perfil do IAM padrão no ambiente. Verifique se adicionou o perfil ao seu espaço. Para obter mais informações, consulte [Adicionar perfis do IAM às conexões da conta](#).

Para ter mais informações, consulte o tópico [Tipos de ação](#). Este tópico tem links para a documentação de cada ação, incluindo a referência YAML.

## Exibir o URL da aplicação no diagrama do fluxo de trabalho

Se o seu fluxo de trabalho implantar um aplicativo, você poderá configurar CodeCatalyst a Amazon para exibir a URL do aplicativo como um link clicável. Esse link aparece no CodeCatalyst console, dentro da ação que o implantou. O diagrama de fluxo de trabalho a seguir mostra o URL de Exibir aplicação que aparece na parte inferior de uma ação.



Ao tornar essa URL clicável no CodeCatalyst console, você pode verificar rapidamente a implantação do seu aplicativo.

**Note**

O URL da aplicação não é compatível com a ação Implantar no Amazon ECS.

Para habilitar esse recurso, adicione uma variável de saída à sua ação com um nome que contenha `appurl` ou `endpointurl`. Você pode usar um nome com ou sem um traço (-), sublinhado (\_) ou espaço ( ). A string diferencia maiúsculas e minúsculas. Defina o valor da variável como o URL `http` ou `https` da aplicação implantada.

**Note**

Se você estiver atualizando uma variável de saída existente para incluir o `app url`, ou a string `endpoint url`, atualize todas as referências a essa variável para usar o novo nome da variável.

Para ver as etapas detalhadas, consulte um destes procedimentos:

- [Para exibir o URL do aplicativo na ação “AWS CDK implantar”](#)
- [Para exibir o URL do aplicativo na ação “Implantar CloudFormation pilha”](#)
- [Para exibir o URL da aplicação em todas as outras ações](#)

Quando terminar de configurar o URL, verifique se ele aparece conforme o esperado seguindo estas instruções:

- [Como verificar se o URL da aplicação foi adicionado](#)

Para exibir o URL do aplicativo na ação “AWS CDK implantar”

1. Se você estiver usando a ação de AWS CDK implantação, adicione uma `CfnOutput` construção (que é um par de valores-chave) no código do seu AWS CDK aplicativo:
  - O nome da chave deve conter `appurl`, `ouendpointurl`, com ou sem um traço (-), sublinhado (\_) ou espaço ( ). A string diferencia maiúsculas e minúsculas.
  - O valor deve ser o URL `http` ou `https` da aplicação implantada.

Por exemplo, seu AWS CDK código pode ter a seguinte aparência:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
 constructor(scope: Construct, id: string, props?: StackProps) {
 super(scope, id, props);
 const bucket = new s3.Bucket(this, 'amzn-s3-demo-bucket', {
 removalPolicy: RemovalPolicy.DESTROY,
 });
 new CfnOutput(this, 'APP-URL', {
 value: https://mycompany.myapp.com,
 description: 'The URL of the deployed application',
 exportName: 'myApp',
 });
 ...
 }
}
```



Para obter mais informações sobre a `CfnOutput` construção, consulte a [interface `CfnOutputProps`](#) na Referência AWS Cloud Development Kit (AWS CDK) da API.

2. Salve e confirme seu código.
3. Vá para [Como verificar se o URL da aplicação foi adicionado](#).

Para exibir o URL do aplicativo na ação “Implantar CloudFormation pilha”

1. Se você estiver usando a ação Deploy CloudFormation stack, adicione uma saída à Outputs seção em seu CloudFormation modelo ou AWS SAM modelo com estas características:
  - A chave (também chamada de ID lógico) deve conter `appurl`, `ouendpointurl`, com ou sem um traço (-), sublinhado (\_) ou espaço ( ). A string diferencia maiúsculas e minúsculas.
  - O valor deve ser o URL `http` ou `https` da aplicação implantada.

Por exemplo, seu CloudFormation modelo pode ter a seguinte aparência:

```
"Outputs" : {
 "APP-URL" : {
 "Description" : "The URL of the deployed app",
 "Value" : "https://mycompany.myapp.com",
 "Export" : {
 "Name" : "My App"
 }
 }
}
```

Para obter mais informações sobre CloudFormation saídas, consulte [Saídas](#) no Guia do AWS CloudFormation usuário.

2. Salve e confirme seu código.
3. Vá para [Como verificar se o URL da aplicação foi adicionado](#).

Para exibir o URL da aplicação em todas as outras ações

Se você estiver usando outra ação para implantar seu aplicativo, como a ação de criação ou GitHub Ações, faça o seguinte para que a URL do aplicativo seja exibida.

1. Defina uma variável de ambiente na seção Inputs ou Steps da ação no arquivo de definição do fluxo de trabalho. A variável deve ter estas características:
  - O name deve conter `appurl`, ou `endpointurl`, com ou sem um traço (-), sublinhado (\_) ou espaço ( ). A string diferencia maiúsculas e minúsculas.
  - O valor deve ser o URL `http` ou `https` da aplicação implantada.

Por exemplo, uma ação de criação pode ser semelhante a esta:

```
Build-action:
 Identifier: aws/build@v1
 Inputs:
 Variables:
 - Name: APP-URL
 Value: https://mycompany.myapp.com
```

... Ou esta:

```
Actions:
 Build:
 Identifier: aws/build@v1
 Configuration:
 Steps:
 - Run: APP-URL=https://mycompany.myapp.com
```

Para ter mais informações sobre definição de variáveis de ambiente, consulte [Definição de uma variável](#).

2. Exporte a variável.

Por exemplo, sua ação de criação pode ser semelhante a esta:

```
Build-action:
 ...
 Outputs:
 Variables:
 - APP-URL
```

Para ter informações sobre exportação de variáveis, consulte [Exportação de uma variável para que outras ações possam usá-la](#).

3. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
4. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.
5. Vá para [Como verificar se o URL da aplicação foi adicionado](#).

Como verificar se o URL da aplicação foi adicionado

- Inicie a execução de um fluxo de trabalho, caso ele não tenha sido iniciado automaticamente. A nova execução deve ter o URL da aplicação exibido como um link clicável no diagrama de fluxo de trabalho. Para ter mais informações sobre como iniciar execuções, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Remoção de um destino de implantação

Você pode remover um destino de implantação, como um cluster ou CloudFormation pilha do Amazon ECS, da página de destinos de implantação no CodeCatalyst console.

### Important

Quando você remove um destino de implantação, ele é removido do CodeCatalyst console, mas permanece disponível no AWS serviço que o hospeda (se ainda existir).

Considere remover um alvo de implantação se o alvo estiver obsoleto. CodeCatalyst Os destinos podem ficar obsoletos se:

- Você excluir o fluxo de trabalho implantado no destino.
- Você alterar a pilha ou o cluster no qual está implantando.
- Você excluiu a pilha ou o cluster do serviço Amazon ECS CloudFormation ou do Amazon ECS no AWS console.

Para remover um destino de implantação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.

4. Escolha o nome do ambiente que contém o destino de implantação que você deseja remover. Para ter informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#).
5. Escolha a guia Destinos de implantação.
6. Marque a caixa de opção ao lado do destino de implantação que você deseja remover.
7. Escolha Remover .

O destino é removido da página.

## Rastreamento do status de implantação por confirmação

A qualquer momento do ciclo de vida do desenvolvimento, é importante conhecer o status de implantação de confirmações específicas, como correções de bugs, novos recursos ou outras mudanças impactantes. Pense nos seguintes cenários nos quais o recurso de rastreamento do status de implantação é útil para as equipes de desenvolvimento:

- Como desenvolvedor, você fez uma correção para solucionar um bug e deseja relatar o status da sua implantação nos ambientes de implantação da sua equipe.
- Como gerente de lançamento, você deseja visualizar uma lista de confirmações implantadas para rastrear e relatar o status de implantação.

CodeCatalyst fornece uma visualização que você pode usar para determinar rapidamente onde confirmações ou alterações individuais foram implantadas e em qual ambiente. Essa visualização inclui:

- Uma lista de confirmações.
- O status das implantações que incluem as confirmações.
- Os ambientes em que as confirmações são implantadas com sucesso.
- O status de todos os testes executados em relação aos commits em seu CI/CD fluxo de trabalho.

O procedimento a seguir detalha como navegar até essa visualização e usá-la para rastrear alterações em seu projeto.

 Note

O rastreamento do status de implantação por confirmação só é compatível com [CodeCatalyst repositórios](#). Você não pode usar esse recurso com um [GitHub repositório](#), [repositório Bitbucket](#) ou [GitLab repositório de projetos](#).

Para rastrear o status de implantação por confirmação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Alterar rastreamento.
4. Nas duas listas suspensas na parte superior do painel principal, escolha o repositório de origem e a ramificação que contêm os commits cujo status de lançamento você deseja visualizar.
5. Selecione Visualizar alterações.

Uma lista de confirmações é exibida.

Para cada confirmação, é possível visualizar o seguinte:

- Confirme informações, como ID, autor, mensagem e quando foram confirmadas. Para obter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).
- O status das implantações em cada ambiente. Para obter mais informações, consulte [Implantação em e Contas da AWS VPCs](#).
- Resultados do teste e da cobertura do código. Para obter mais informações, consulte [Teste com fluxos de trabalho](#).

 Note

Os resultados da Análise de composição de software (SCA) não são exibidos.

6. (Opcional) Para visualizar mais informações sobre as alterações relacionadas a uma confirmação específica, incluindo a implantação mais recente e informações detalhadas sobre a cobertura do código e o teste de unidade, selecione Visualizar detalhes para essa confirmação.

## Visualização dos logs de implantação

Você pode visualizar registros relacionados a ações de implantação específicas para solucionar problemas na Amazon CodeCatalyst.

Você pode visualizar os logs em um [fluxo de trabalho](#) ou um [ambiente](#).

Para visualizar os logs de uma ação de implantação em um fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Selecione Execuções.
6. Escolha a execução do fluxo de trabalho que implantou a aplicação.
7. No diagrama do fluxo de trabalho, escolha a ação cujos logs você deseja exibir.
8. Escolha a guia Logs e expanda as seções para revelar as mensagens de log.
9. Para ver mais registros, escolha a guia Resumo e, em seguida, escolha Exibir em CloudFormation (se estiver disponível) para ver mais registros lá. Talvez seja necessário fazer login na AWS.

Para visualizar os logs de uma ação de implantação em um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.
4. Escolha o ambiente no qual a aplicação foi implantada.
5. Em Atividade de implantação, encontre a coluna ID de execução do fluxo de trabalho e escolha a execução do fluxo de trabalho que implantou sua pilha.
6. No diagrama do fluxo de trabalho, escolha a ação cujos logs você deseja exibir.
7. Escolha a guia Logs e expanda as seções para revelar as mensagens de log.

8. Para ver mais registros, escolha a guia Resumo e, em seguida, escolha Exibir em CloudFormation (se estiver disponível) para ver mais registros lá. Talvez seja necessário fazer login na AWS.

## Visualizar informações de implantação

Você pode ver as seguintes informações sobre uma implantação na Amazon CodeCatalyst:

- Atividade de implantação, incluindo o status da implantação, a hora de início, a hora de término, o histórico e a duração dos eventos.
- Nome da pilha Região da AWS, horário da última atualização e fluxos de trabalho associados.
- Confirmações e solicitações pull.
- Informações específicas da ação, por exemplo, CloudFormation eventos e resultados.

Você pode visualizar as informações de implantação em um [fluxo de trabalho](#), um [ambiente](#) ou uma [ação](#) de fluxo de trabalho.

Para visualizar as informações de implantação em um fluxo de trabalho

- Acesse a execução do fluxo de trabalho que implantou a aplicação. Para instruções, consulte [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Para visualizar as informações de implantação em um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, selecione CI/CD e Ambientes.
4. Escolha o ambiente em que sua pilha foi implantada, por exemplo, Production.
5. Selecione Atividade de implantação para visualizar o histórico de implantação de suas pilhas, o status das implantações (por exemplo, SUCCEEDED ou FAILED) e outras informações relacionadas à implantação.
6. Selecione Destino de implantação para visualizar informações sobre as pilhas, clusters ou outros destinos implantados no ambiente. Você pode visualizar informações como nome da pilha, região, provedor e identificador.

Para visualizar as informações de implantação em uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. No diagrama do fluxo de trabalho, escolha a ação do fluxo de trabalho que implantou a aplicação. Por exemplo, você pode escolher DeployCloudFormationStack.
6. Revise o conteúdo no painel direito para ter informações de implantação específicas da ação.

## Criação de um fluxo de trabalho

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

### Tip

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Use o procedimento a seguir para criar um fluxo de trabalho em CodeCatalyst. O fluxo de trabalho será armazenado como um arquivo YAML em uma pasta `~/.codecatalyst/workflows/` no repositório de origem escolhido. Você também pode armazenar o fluxo de trabalho em uma subpasta de `~/.codecatalyst/workflows/` ou colocando o nome da pasta antes do nome do arquivo do fluxo de trabalho ao confirmá-lo. Para ter mais informações, consulte as instruções a seguir.

Para obter mais informações sobre fluxos de trabalho, consulte [Compilação, teste e implantação com fluxos de trabalho](#).



## Visual

Como criar um fluxo de trabalho usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione Criar fluxo de trabalho.

A caixa de diálogo Criar fluxo de trabalho é exibida.

5. No campo Repositório de origem, selecione um repositório de origem no qual o arquivo de definição do fluxo de trabalho residirá. Se não existir nenhum repositório de origem, [crie um](#).
6. No campo Ramificação, selecione uma ramificação na qual o arquivo de definição do fluxo de trabalho residirá.
7. Escolha Criar.

A Amazon CodeCatalyst salva as informações do repositório e da filial na memória, mas o fluxo de trabalho ainda não está comprometido.

8. Selecione Visual.
9. Crie o fluxo de trabalho:
  - a. (Opcional) No diagrama do fluxo de trabalho, selecione a caixa Origem e Gatilhos. Um painel Gatilhos é exibido. Selecione Adicionar gatilho para adicionar um gatilho. Para obter mais informações, consulte [Adição de gatilhos aos fluxos de trabalho](#).
  - b. Selecione + Ações (canto superior esquerdo). O catálogo de ações é exibido.
  - c. Selecione o sinal de adição (+) dentro de uma ação para adicioná-la ao fluxo de trabalho. Use o painel à direita para configurar a ação. Para obter mais informações, consulte [Adição de uma ação a um fluxo de trabalho](#).
  - d. (Opcional) Selecione Propriedades do fluxo de trabalho (canto superior direito). Um painel de Propriedades do fluxo de trabalho é exibido. Configure o nome do fluxo de trabalho, o modo de execução e a computação. Para obter mais informações, consulte [Configurar o comportamento de enfileiramento das execuções](#) e [Configuração de imagens de computação e runtime](#).
10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar e, na caixa de diálogo Confirmar fluxo de trabalho, faça o seguinte:

- a. Em Nome do arquivo do fluxo de trabalho, deixe o nome padrão ou digite o seu próprio. O arquivo será armazenado como uma pasta `~/.codecatalyst/workflows/` no repositório de origem e ramificação selecionados. Coloque uma pasta ou uma subpasta antes do nome do arquivo. Exemplos:
  - A especificação de `my-workflow` (sem pasta) armazena o arquivo como `~/.codecatalyst/workflows/my-workflow.yaml`
  - A especificação de `folder/subfolder/my-workflow` armazena o arquivo como `~/.codecatalyst/workflows/folder/subfolder/my-workflow.yaml`
- b. Em Confirmar mensagem, deixe a mensagem padrão ou insira a sua.
- c. Em Repositório e Ramificação, selecione o repositório e a ramificação de origem para o arquivo de definição do fluxo de trabalho. Esses campos devem ser definidos para o repositório e a ramificação especificados anteriormente na caixa de diálogo Criar fluxo de trabalho. Se desejar, altere o repositório e a ramificação agora.

 Note

Depois de confirmar o arquivo de definição de fluxo de trabalho, ele não poderá ser associado a outro repositório ou ramificação, portanto, escolha-os com cuidado.

- d. Selecione Confirmar para confirmar o arquivo de definição do fluxo de trabalho.

## YAML

Como criar um fluxo de trabalho usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione Criar fluxo de trabalho.

A caixa de diálogo Criar fluxo de trabalho é exibida.

5. No campo Repositório de origem, selecione um repositório de origem no qual o arquivo de definição do fluxo de trabalho residirá. Se não existir nenhum repositório de origem, [crie um](#).

6. No campo Ramificação, selecione uma ramificação na qual o arquivo de definição do fluxo de trabalho residirá.
7. Escolha Criar.

A Amazon CodeCatalyst salva as informações do repositório e da filial na memória, mas o fluxo de trabalho ainda não está comprometido.

8. Selecione YAML.
9. Crie o fluxo de trabalho:
  - a. (Opcional) Adicione um gatilho ao código YAML. Para obter mais informações, consulte [Adição de gatilhos aos fluxos de trabalho](#).
  - b. Selecione + Ações (canto superior esquerdo). O catálogo de ações é exibido.
  - c. Selecione o sinal de adição (+) dentro de uma ação para adicioná-la ao fluxo de trabalho. Use o painel à direita para configurar a ação. Para obter mais informações, consulte [Adição de uma ação a um fluxo de trabalho](#).
  - d. (Opcional) Selecione Propriedades do fluxo de trabalho (canto superior direito). Um painel de Propriedades do fluxo de trabalho é exibido. Configure o nome do fluxo de trabalho, o modo de execução e a computação. Para obter mais informações, consulte [Configurar o comportamento de enfileiramento das execuções](#) e [Configuração de imagens de computação e runtime](#).
10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar e, na caixa de diálogo Confirmar fluxo de trabalho, faça o seguinte:
  - a. Em Nome do arquivo do fluxo de trabalho, deixe o nome padrão ou digite o seu próprio. O arquivo será armazenado como uma pasta `~/.codecatalyst/workflows/` no repositório de origem e ramificação selecionados. Coloque uma pasta ou uma subpasta antes do nome do arquivo. Exemplos:
    - A especificação de `my-workflow` (sem pasta) armazena o arquivo como `~/.codecatalyst/workflows/my-workflow.yaml`
    - A especificação de `folder/subfolder/my-workflow` armazena o arquivo como `~/.codecatalyst/workflows/folder/subfolder/my-workflow.yaml`
  - b. Em Confirmar mensagem, deixe a mensagem padrão ou insira a sua.
  - c. Em Repositório e Ramificação, selecione o repositório e a ramificação de origem para o arquivo de definição do fluxo de trabalho. Esses campos devem ser definidos para o

repositório e a ramificação especificados anteriormente na caixa de diálogo Criar fluxo de trabalho. Se desejar, altere o repositório e a ramificação agora.

 Note

Depois de confirmar o arquivo de definição de fluxo de trabalho, ele não poderá ser associado a outro repositório ou ramificação, portanto, escolha-os com cuidado.

- d. Selecione Confirmar para confirmar o arquivo de definição do fluxo de trabalho.

## Executar um fluxo de trabalho

Execução é uma iteração única de um fluxo de trabalho. Durante uma execução, CodeCatalyst executa as ações definidas no arquivo de configuração do fluxo de trabalho e gera os registros, artefatos e variáveis associados.

Você pode iniciar uma execução manual ou automaticamente por meio de um gatilho de fluxo de trabalho. Um exemplo de gatilho de fluxo de trabalho pode ser um desenvolvedor de software enviando uma confirmação para sua ramificação principal.

Você também pode parar manualmente a execução de um fluxo de trabalho em andamento, caso a tenha iniciado por engano.

Se várias execuções de fluxo de trabalho forem iniciadas ao mesmo tempo, você poderá configurar como deseja que essas execuções sejam enfileiradas. Use o comportamento de enfileiramento padrão, em que as execuções são enfileiradas uma após a outra na ordem em que foram iniciadas, ou você pode fazer com que uma execução posterior substitua (ou “assuma o controle”) de uma anterior para acelerar sua execução durante todo o processo. Configurar as execuções do fluxo de trabalho para que ocorram paralelamente, de forma que nenhuma execução espere por outra, também é possível.

Depois de iniciar a execução de um fluxo de trabalho, manual ou automaticamente, você pode ver o status da execução e outros detalhes. Por exemplo, você pode ver quando ela foi iniciada, por quem ela foi iniciada e se ainda está em execução.

### Tópicos

- [Iniciar um fluxo de trabalho executado manualmente](#)

- [Início da execução automática de um fluxo de trabalho usando gatilhos](#)
- [Configurar gatilhos somente manuais](#)
- [Parada de uma execução de fluxo de trabalho](#)
- [Bloquear uma execução de fluxo de trabalho](#)
- [Solicitar aprovações em execuções de fluxo de trabalho](#)
- [Configurar o comportamento de enfileiramento das execuções](#)
- [Armazenar arquivos em cache entre execuções de fluxo de trabalho](#)
- [Visualização do status e detalhes de execução do fluxo de trabalho](#)

## Iniciar um fluxo de trabalho executado manualmente

Na Amazon CodeCatalyst, você pode iniciar um fluxo de trabalho executado manualmente a partir do CodeCatalyst console.

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

### Note

Você também pode iniciar a execução de um fluxo de trabalho manualmente ao [configurar um gatilho](#).

Como iniciar um fluxo de trabalho executado manualmente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Executar.

## Início da execução automática de um fluxo de trabalho usando gatilhos

Você pode iniciar a execução automática de um CodeCatalyst fluxo de trabalho da Amazon com um gatilho de fluxo de trabalho.

Um gatilho de fluxo de trabalho, ou simplesmente um gatilho, permite que você inicie a execução automática de um fluxo de trabalho quando determinados eventos ocorrerem, como um envio de código. Talvez você queira configurar gatilhos para liberar seus desenvolvedores de software da necessidade de iniciar execuções de fluxo de trabalho manualmente por meio do CodeCatalyst console.

É possível usar três tipos de gatilho:

- **Envio:** um gatilho de envio de código faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma confirmação é enviada.
- **Solicitação pull:** um gatilho de solicitação pull faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma solicitação pull é criada, revisada ou fechada.
- **Programação:** um gatilho de programação faz com que a execução de um fluxo de trabalho comece em uma programação definida por você. Pense em usar um gatilho de programação para executar compilações noturnas do software para que a versão mais recente esteja pronta para os desenvolvedores de software trabalharem na manhã seguinte.

É possível usar gatilhos de envio, solicitação pull e programação de maneira independente ou combinados no mesmo fluxo de trabalho.

Os gatilhos são opcionais. Se você não configurar nenhum, só poderá iniciar um fluxo de trabalho manualmente.

### Tip

Para ver um gatilho em ação, inicie um projeto com um esquema. A maioria dos esquemas contém um fluxo de trabalho com um gatilho. Procure a propriedade `Trigger` no arquivo de definição do fluxo de trabalho do esquema. Para obter mais informações sobre esquemas, consulte [Criar um projeto com um esquema](#).

### Tópicos

- [Exemplos: gatilhos em fluxos de trabalho](#)

- [Diretrizes para o uso de gatilhos e ramificações](#)
- [Adição de gatilhos aos fluxos de trabalho](#)

## Exemplos: gatilhos em fluxos de trabalho

Os exemplos a seguir mostram como adicionar diferentes tipos de gatilhos em um arquivo de definição de fluxo de trabalho do Amazon CodeCatalyst.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

### Tópicos

- [Exemplo: um simples gatilho de envio de código](#)
- [Exemplo: um gatilho simples “enviar para principal”](#)
- [Exemplo: um simples gatilho de solicitação pull](#)
- [Exemplo: um gatilho de programação simples](#)
- [Exemplo: um gatilho com uma programação e ramificações](#)
- [Exemplo: um gatilho com uma programação, um envio e ramificações](#)
- [Exemplo: um gatilho com uma extração e ramificações](#)
- [Exemplo: um gatilho com uma extração, ramificações e um evento “FECHADO”](#)
- [Exemplo: um gatilho com um envio, ramificações e arquivos](#)
- [Exemplo: um gatilho manual](#)
- [Exemplo: gatilhos em uma configuração de vários fluxos de trabalho de CI/CD](#)

### Exemplo: um simples gatilho de envio de código

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que o código é enviado para qualquer ramificação no seu repositório de origem.

Quando esse gatilho é ativado, o CodeCatalyst inicia a execução de um fluxo de trabalho usando os arquivos na ramificação para a qual você está enviando (ou seja, a ramificação de destino).

Por exemplo, se você enviar uma confirmação para `main`, o CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem ativados em `main`.

Outro exemplo, se você enviar uma confirmação para `feature-branch-123`, o CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem ativados em `feature-branch-123`.

**Triggers:**

- Type: PUSH

 **Note**

Se você quiser que a execução de um fluxo de trabalho inicie somente quando enviar para `main`, consulte [Exemplo: um gatilho simples “enviar para principal”](#).

### Exemplo: um gatilho simples “enviar para principal”

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que o código é enviado para a ramificação `main` – e apenas a ramificação `main` – no seu repositório de origem.

**Triggers:**

- Type: PUSH
- Branches:
  - `main`

### Exemplo: um simples gatilho de solicitação pull

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma solicitação pull é criada ou revisada no seu repositório de origem.

Quando esse gatilho é ativado, o CodeCatalyst inicia uma execução de fluxo de trabalho usando o arquivo de definição de fluxo de trabalho e outros arquivos de origem na ramificação de onde você está extraindo (ou seja, a ramificação de origem).

Por exemplo, se você criar uma solicitação pull com uma ramificação de origem chamada `feature-123` e uma ramificação de destino chamada `main`, o CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem em `feature-123`.

**Triggers:**

- Type: PULLREQUEST



**Events:**

- OPEN
- REVISION

Exemplo: um gatilho de programação simples

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho à meia-noite (UTC+0) de segunda a sexta.

Quando esse gatilho é ativado, o CodeCatalyst inicia uma única execução de fluxo de trabalho para cada ramificação em seu repositório de origem que contém um arquivo de definição de fluxo de trabalho com esse gatilho.

Por exemplo, se você tiver três ramificações em seu repositório de origem, `main`, `release-v1` e `feature-123`, e cada uma dessas ramificações contiver um arquivo de definição de fluxo de trabalho com o gatilho a seguir, o CodeCatalyst iniciará três execuções de fluxo de trabalho: uma usando os arquivos em `main`, outra usando os arquivos em `release-v1` e outra usando os arquivos em `feature-123`.

**Triggers:**

- Type: SCHEDULE  
Expression: "`0 0 ? * MON-FRI *`"

Para ver mais exemplos de expressões cron que podem ser usadas na propriedade `Expression`, consulte [Expression](#).

Exemplo: um gatilho com uma programação e ramificações

O exemplo a seguir mostra um gatilho que inicia uma execução de fluxo de trabalho às 18h15 (UTC +0) todos os dias.

Quando esse gatilho é ativado, o CodeCatalyst inicia a execução de um fluxo de trabalho usando os arquivos na ramificação `main` e inicia execuções adicionais para cada ramificação que começa com `release-`.

Por exemplo, se você tiver ramificações chamadas `main`, `release-v1`, `bugfix-1` e `bugfix-2` no seu repositório de origem, o CodeCatalyst iniciará duas execuções de fluxo de trabalho: uma usando os arquivos em `main` e outra usando os arquivos em `release-v1`. Ele não inicia as execuções do fluxo de trabalho para as ramificações `bugfix-1` e `bugfix-1`.

**Triggers:**

```
- Type: SCHEDULE
 Expression: "15 18 * * ? *"
 Branches:
 - main
 - release\-.*
```

Para ver mais exemplos de expressões cron que podem ser usadas na propriedade `Expression`, consulte [Expression](#).

Exemplo: um gatilho com uma programação, um envio e ramificações

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho à meia-noite (UTC+0) todos os dias e sempre que o código é enviado para a ramificação `main`.

Neste exemplo:

- A execução do fluxo de trabalho começa à meia-noite todos os dias. A execução do fluxo de trabalho usa o arquivo de definição do fluxo de trabalho e outros arquivos de origem na ramificação `main`.
- A execução do fluxo de trabalho também é iniciada sempre que você envia uma confirmação para a ramificação `main`. A execução do fluxo de trabalho usa o arquivo de definição do fluxo de trabalho e outros arquivos de origem na ramificação de destino (`main`).

```
Triggers:
- Type: SCHEDULE
 Expression: "0 0 * * ? *"
 Branches:
 - main
- Type: PUSH
 Branches:
 - main
```

Para ver mais exemplos de expressões cron que podem ser usadas na propriedade `Expression`, consulte [Expression](#).

Exemplo: um gatilho com uma extração e ramificações

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que alguém abre ou modifica uma solicitação pull com uma ramificação de destino chamada `main`. Embora a ramificação especificada na configuração `Triggers` seja `main`, a execução do fluxo de

trabalho usará o arquivo de definição do fluxo de trabalho e outros arquivos de origem na ramificação de origem (que é a ramificação da qual você está extraindo).

```
Triggers:
- Type: PULLREQUEST
 Branches:
 - main
 Events:
 - OPEN
 - REVISION
```

Exemplo: um gatilho com uma extração, ramificações e um evento “FECHADO”

O exemplo a seguir mostra um gatilho que inicia uma execução de fluxo de trabalho sempre que uma solicitação pull é fechada em uma ramificação que começa com `main`.

Neste exemplo:

- Quando você fecha uma solicitação pull com uma ramificação de destino que começa com `main`, a execução de um fluxo de trabalho é iniciada automaticamente usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem na ramificação de origem (agora fechada).
- Se você configurou seu repositório de origem para excluir ramificações automaticamente após a mesclagem de uma solicitação pull, essas ramificações nunca terão a chance de entrar no estado `CLOSED`. Isso significa que as ramificações mescladas não ativarão o gatilho `CLOSED` da solicitação pull. A única maneira de ativar o gatilho `CLOSED` nesse cenário é fechar a solicitação pull sem mesclá-la.

```
Triggers:
- Type: PULLREQUEST
 Branches:
 - main.*
 Events:
 - CLOSED
```

Exemplo: um gatilho com um envio, ramificações e arquivos

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma alteração é feita no arquivo `filename.txt`, ou em qualquer arquivo no diretório `src`, na ramificação `main`.

Quando esse gatilho é ativado, o CodeCatalyst inicia uma execução de fluxo de trabalho usando o arquivo de definição de fluxo de trabalho e outros arquivos de origem na ramificação `main`.

```
Triggers:
- Type: PUSH
 Branches:
 - main
 FilesChanged:
 - filename.txt
 - src\/*.*
```

#### Exemplo: um gatilho manual

Para configurar um gatilho manual, omita a seção `Triggers` do arquivo de definição do fluxo de trabalho. Sem essa seção, os usuários são forçados a iniciar o fluxo de trabalho manualmente selecionando o botão Executar no console do CodeCatalyst. Para obter mais informações, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

#### Exemplo: gatilhos em uma configuração de vários fluxos de trabalho de CI/CD

Este exemplo descreve como configurar gatilhos quando você deseja usar fluxos de trabalho separados do Amazon CodeCatalyst para integração contínua (CI) e implantação contínua (CD).

Nesse cenário, você configura dois fluxos de trabalho:

- um fluxo de trabalho de CI: esse fluxo de trabalho cria e testa a aplicação quando uma solicitação pull é criada ou revisada.
- um fluxo de trabalho de CD: esse fluxo de trabalho compila e implanta a aplicação quando uma solicitação pull é mesclada.

O arquivo de definição do fluxo de trabalho de CI seria semelhante a este:

```
Triggers:
- Type: PULLREQUEST
 Branches:
 - main
 Events:
 - OPEN
 - REVISION
Actions:
```

```
BuildAction:
 instructions-for-building-the-app
TestAction:
 instructions-for-test-the-app
```

O código `Triggers` indica que um fluxo de trabalho deve ser executado automaticamente sempre que um desenvolvedor de software cria uma solicitação pull (ou [modifica uma](#)) solicitando a mesclagem de sua ramificação de recursos com a ramificação `main`. O CodeCatalyst inicia a execução do fluxo de trabalho usando o código-fonte na ramificação de origem (que é a ramificação de recursos).

O arquivo de definição do fluxo de trabalho de CD seria semelhante a este:

```
Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 BuildAction:
 instructions-for-building-the-app
 DeployAction:
 instructions-for-deploying-the-app
```

O código `Triggers` indica que o fluxo de trabalho deve ser iniciado automaticamente quando ocorre uma mesclagem para `main`. O CodeCatalyst inicia a execução do fluxo de trabalho usando o código-fonte na ramificação `main`.

## Diretrizes para o uso de gatilhos e ramificações

Esta seção descreve algumas das principais diretrizes ao configurar os gatilhos do Amazon CodeCatalyst que incluem ramificações.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

- Diretriz 1: para os gatilhos de envio e solicitação pull, se você for especificar uma ramificação, deverá especificar a ramificação de destino (ou “para”) na configuração do gatilho. Nunca especifique a ramificação de origem (ou “de”).

No exemplo a seguir, um envio de qualquer ramificação para `main` ativa o fluxo de trabalho.

```
Triggers:
- Type: PUSH
 Branches:
 - main
```

No exemplo a seguir, uma solicitação pull de qualquer ramificação para main ativa o fluxo de trabalho.

```
Triggers:
- Type: PULLREQUEST
 Branches:
 - main
 Events:
 - OPEN
 - REVISION
```

- Diretriz 2: para gatilhos de envio, depois que o fluxo de trabalho for ativado, ele será executado usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação de destino.
- Diretriz 3: para gatilhos de solicitação pull, depois que o fluxo de trabalho for ativado, ele será executado usando o arquivo de definição de fluxo de trabalho e os arquivos de origem na ramificação de origem (mesmo que você tenha especificado a ramificação de destino na configuração do gatilho).
- Diretriz 4: o mesmo gatilho em uma ramificação pode não ser executado em outra ramificação.

Considere o gatilho de envio a seguir:

```
Triggers:
- Type: PUSH
 Branches:
 - main
```

Se o arquivo de definição do fluxo de trabalho que contém esse gatilho existir em main e for clonado para test, o fluxo de trabalho nunca começará a usar automaticamente os arquivos em test (embora você possa iniciar o fluxo de trabalho manualmente para que ele use os arquivos em test). Revise a Diretriz 2 para entender por que o fluxo de trabalho nunca será executado automaticamente usando os arquivos em test.

Considere também o seguinte gatilho de solicitação pull:

Triggers:

- Type: PULLREQUEST

Branches:

- main

Events:

- OPEN
- REVISION

Se o arquivo de definição do fluxo de trabalho contendo esse gatilho existir em `main`, o fluxo de trabalho nunca será executado usando os arquivos em `main`. (No entanto, se você criar uma ramificação `test` a partir de `main`, o fluxo de trabalho será executado usando os arquivos em `test`.) Revise a Diretriz 3 para entender o porquê.

## Adição de gatilhos aos fluxos de trabalho

Use as instruções a seguir para adicionar um gatilho push, pull ou schedule ao seu CodeCatalyst fluxo de trabalho da Amazon.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

### Visual

Para adicionar um gatilho (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a caixa Origem e Gatilhos.
8. No painel de configuração, selecione Adicionar gatilho.

9. Na caixa de diálogo Adicionar gatilho, forneça as informações nos campos, da seguinte forma.

### Tipo de gatilho

Especifique o tipo de gatilho. É possível usar um dos seguintes valores:

- Envio (editor visual) ou PUSH (editor YAML)

Um gatilho de envio inicia a execução de um fluxo de trabalho quando uma alteração é enviada para seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação para a qual você está enviando (ou seja, a ramificação de destino).

- Solicitação pull (editor visual) ou PULLREQUEST (editor YAML)

Um gatilho de solicitação pull inicia a execução de um fluxo de trabalho quando uma solicitação pull é aberta, atualizada ou fechada no seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação da a qual você está extraindo (ou seja, a ramificação de origem).

- Programação (editor visual) ou SCHEDULE (editor YAML)

Um gatilho de programação inicia a execução do fluxo de trabalho em uma programação definida por uma expressão cron especificada por você. Uma execução de fluxo de trabalho separada será iniciada para cada ramificação em seu repositório de origem usando os arquivos da ramificação. (Para limitar as ramificações nas quais o gatilho é ativado, use o campo Ramificações (editor visual) ou a propriedade Branches (editor YAML).)

Ao configurar um gatilho de programação, siga estas instruções:

- Use apenas um gatilho de programação por fluxo de trabalho.
- Se você definiu vários fluxos de trabalho em seu CodeCatalyst espaço, recomendamos que você agende no máximo 10 deles para começarem simultaneamente.
- Certifique-se de configurar a expressão cron do gatilho com tempo adequado entre as execuções. Para obter mais informações, consulte [Expression](#).

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

### Eventos para solicitação pull



Esse campo só aparece se você selecionou o tipo de gatilho de solicitação pull.

Especifique o tipo de eventos de solicitação pull que iniciarão a execução de um fluxo de trabalho. Os valores válidos são os seguintes:

- A solicitação pull é criada (editor visual) ou OPEN (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma solicitação pull é criada.

- A solicitação pull é fechada (editor visual) ou CLOSED (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma solicitação pull é fechada. O comportamento do evento CLOSED é complicado e é melhor compreendido por meio de um exemplo. Consulte [Exemplo: um gatilho com uma extração, ramificações e um evento “FECHADO”](#) para obter mais informações.

- Uma nova revisão é feita para solicitação pull (editor visual) ou REVISION (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma revisão da solicitação pull é criada. A primeira revisão é criada quando a solicitação pull é criada. Depois disso, uma nova revisão é criada sempre que alguém envia uma nova confirmação para a ramificação de origem especificada na solicitação pull. Se você incluir o evento REVISION em seu gatilho de solicitação pull, poderá omitir o evento OPEN, pois REVISION é um superconjunto de OPEN.

Você pode especificar vários eventos no mesmo gatilho de solicitação pull.

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

## Programação

Esse campo só aparece se você selecionou o tipo de gatilho Programação.

Especifique a expressão cron que descreve quando você deseja que suas execuções de fluxo de trabalho programadas ocorram.

As expressões Cron CodeCatalyst usam a seguinte sintaxe de seis campos, em que cada campo é separado por um espaço:

*minutes hours days-of-month month days-of-week year*

## Exemplos de expressões cron

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano | Significado                                                             |
|---------|-------|-------------|-----|----------------|-----|-------------------------------------------------------------------------|
| 0       | 0     | ?           | *   | SEG-SEX        | *   | Executada um fluxo de trabalho à meia-noite (UTC+0) de segunda a sexta. |
| 0       | 2     | *           | *   | ?              | *   | Executada um fluxo de trabalho às 2h (UTC+0) todos os dias.             |
| 15      | 22    | *           | *   | ?              | *   | Executada um fluxo de trabalho às 22h15 (UTC+0) todos os dias.          |

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano       | Significado                                                                                                                |
|---------|-------|-------------|-----|----------------|-----------|----------------------------------------------------------------------------------------------------------------------------|
| 0/30    | 22-2  | ?           | *   | SÁB-DOM        | *         | Executa um fluxo de trabalho a cada 30 minutos de sábado a domingo, entre 22h do dia inicial e 2h do dia seguinte (UTC+0). |
| 45      | 13    | L           | *   | ?              | 2023-2027 | Executa um fluxo de trabalho às 13h45 (UTC+0) no último dia do mês entre os anos de 2023 e 2027, inclusive.                |

Ao especificar expressões cron em CodeCatalyst, siga estas diretrizes:

- Especifique uma única expressão cron por gatilho SCHEDULE.

- Coloque a expressão cron entre aspas duplas (") no editor YAML.
- Especifique o horário em Tempo Universal Coordenado (UTC). Outros fusos horários não são suportados.
- Configure pelo menos 30 minutos entre as execuções. Não há suporte para uma cadência mais rápida.
- Especifique o *days-of-week* campo *days-of-month* ou, mas não ambos. Se você especificar um valor ou asterisco (\*) em um dos campos, deverá usar um ponto de interrogação (?) no outro. O asterisco significa "tudo" e o ponto de interrogação significa "qualquer".

Para obter mais exemplos de expressões cron e informações sobre curingas como ?, , e \*L, consulte a [referência de expressões Cron no Guia](#) do usuário da Amazon EventBridge . As expressões Cron CodeCatalyst funcionam exatamente da mesma maneira. EventBridge

Para ver exemplos de gatilhos de programação, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

## Ramificações e padrão de ramificação

(Optional)

Especifique as ramificações em seu repositório de origem que o acionador monitora para saber quando iniciar a execução de um fluxo de trabalho. Você pode usar padrões regex para definir os nomes das ramificações. Por exemplo, use `main.*` para combinar todas as ramificações que começam com `main`.

As ramificações a serem especificadas são diferentes dependendo do tipo de gatilho:

- Para um gatilho de envio, especifique as ramificações para as quais você está enviando, ou seja, as ramificações de destino. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando os arquivos na ramificação correspondente.

Exemplos: `main.*`, `mainline`

- Para um gatilho de solicitação pull, especifique as ramificações para as quais você está enviando, ou seja, as ramificações de destino. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando o arquivo de definição do fluxo

de trabalho e os arquivos de origem na ramificação de origem (não a ramificação correspondente).

Exemplos: `main.*`, `mainline`, `v1\-.*` (corresponde às ramificações que começam com `v1-`)

- Para um gatilho de programação, especifique as ramificações que contêm os arquivos que devem ser usados pela sua execução programada. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação correspondente.

Exemplos: `main.*`, `version\-1\.`

#### Note

Se você não especificar ramificações, o gatilho monitorará todas as ramificações em seu repositório de origem e iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e os arquivos de origem em:

- A ramificação para a qual você está enviando (para gatilhos de envio). Para obter mais informações, consulte [Exemplo: um simples gatilho de envio de código](#).
- A ramificação da qual você está extraindo (para gatilhos de solicitação pull). Para obter mais informações, consulte [Exemplo: um simples gatilho de solicitação pull](#).
- Todas as ramificações (para gatilhos de programação). Uma execução de fluxo de trabalho será iniciada por ramificação em seu repositório de origem. Para obter mais informações, consulte [Exemplo: um gatilho de programação simples](#).

Para ter mais informações sobre ramificações e gatilhos, consulte [Diretrizes para o uso de gatilhos e ramificações](#).

Para obter mais exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

## Arquivos alterados

Esse campo só aparece se você selecionou o tipo de gatilho de Envio ou Solicitação pull.

Especifique os arquivos ou pastas em seu repositório de origem que o acionador monitora para saber quando iniciar a execução de um fluxo de trabalho. Você pode usar expressões regulares para corresponder nomes ou caminhos de arquivos.

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar um gatilho (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Adicione uma seção de Triggers e propriedades subjacentes usando o exemplo a seguir como guia. Para obter mais informações, consulte o [Triggers](#) no [Definição do YAML do fluxo de trabalho](#).

Um gatilho de envio de código pode ter a seguinte aparência:

```
Triggers:
 - Type: PUSH
 Branches:
 - main
```

Um gatilho de solicitação pull pode ter a seguinte aparência:

```
Triggers:
 - Type: PULLREQUEST
```

```
Branches:
- main.*
Events:
- OPEN
- REVISION
- CLOSED
```

Um gatilho de programação pode ter a seguinte aparência:

```
Triggers:
- Type: SCHEDULE
 Branches:
 - main.*
 # Run the workflow at 1:15 am (UTC+0) every Friday until the end of 2023
 Expression: "15 1 ? * FRI 2022-2023"
```

Para ver mais exemplos de expressões cron que podem ser usadas na propriedade Expression, consulte [Expression](#).

Para ver mais exemplos de gatilhos de envio, solicitação pull e programação, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Configurar gatilhos somente manuais

Você pode limitar um fluxo de trabalho para que ele só possa ser iniciado manualmente pela sua equipe usando o botão Executar no CodeCatalyst console. Para configurar essa funcionalidade, remova a seção Triggers no arquivo de definição do fluxo de trabalho. A seção Triggers é incluída por padrão quando você cria um fluxo de trabalho, mas a seção é opcional e pode ser removida.

Use as instruções a seguir para remover a seção Triggers no arquivo de definição do fluxo de trabalho para que o fluxo de trabalho só possa ser iniciado manualmente.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

Para ter mais informações sobre a execução de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

## Visual

Como remover a seção “Gatilhos” (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. Selecione a caixa Origem no diagrama do fluxo de trabalho.
8. Em Gatilhos, selecione o ícone da lixeira para remover a seção Triggers do fluxo de trabalho.
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como remover a seção “Gatilhos” (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.



7. Encontre e remova a seção `Triggers`.
8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Parada de uma execução de fluxo de trabalho

Use o procedimento a seguir para parar a execução de um fluxo de trabalho em andamento. Talvez você queira parar uma execução se ela tiver sido iniciada por acidente.

Quando você interrompe a execução de um fluxo de trabalho, CodeCatalyst aguarda a conclusão das ações em andamento antes de marcar a execução como Parada no CodeCatalyst console. Todas as ações que não tiveram a chance de começar não serão iniciadas e serão marcadas como Abandonadas.

### Note

Se uma execução estiver na fila (ou seja, não tiver ações em andamento), a execução será parada imediatamente.

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

Como interromper uma execução de fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Em Fluxos de trabalho, selecione Execuções e a execução em andamento na lista.
5. Escolha Parar.

## Bloquear uma execução de fluxo de trabalho

Portão é um componente do fluxo de trabalho que pode ser usado para impedir que a execução do fluxo de trabalho continue, a menos que determinadas condições sejam atendidas. Um exemplo de

porta é a porta de aprovação, na qual os usuários devem enviar uma aprovação no CodeCatalyst console antes que a execução do fluxo de trabalho possa continuar.

É possível adicionar portões entre sequências de ações a um fluxo de trabalho ou antes da primeira ação (executada imediatamente após o download da Origem). Também será possível adicionar portões após a última ação, se necessário.

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

## Tópicos

- [Tipos de portões](#)
- [Posso configurar um portão para ser executado paralelamente a outra ação?](#)
- [Posso usar um portão para impedir o início da execução de um fluxo de trabalho?](#)
- [Limitações de portões](#)
- [Adicionar um portão a um fluxo de trabalho](#)
- [Sequenciar portões e ações](#)
- [Especificar a versão de um portão](#)

## Tipos de portões

Atualmente, a Amazon CodeCatalyst oferece suporte a um tipo de portão: o portão de aprovação. Para obter mais informações, consulte [Solicitar aprovações em execuções de fluxo de trabalho](#).

Posso configurar um portão para ser executado paralelamente a outra ação?

Não. Os portões só podem ser executados antes ou depois de uma ação. Para obter mais informações, consulte [Sequenciar portões e ações](#).

Posso usar um portão para impedir o início da execução de um fluxo de trabalho?

Sim, com qualificações.

Você pode evitar que a execução de um fluxo de trabalho execute tarefas, o que é um pouco diferente de impedir que ela seja iniciada.

Para evitar que um fluxo de trabalho execute tarefas, adicione um portão antes da primeira ação em um fluxo de trabalho. Nesse cenário, a execução de um fluxo de trabalho será iniciada, o que

significa que ela baixará os arquivos do repositório de origem, mas será impedida de realizar tarefas até que o portão seja desbloqueado.

#### Note

Os fluxos de trabalho que são iniciados e depois são bloqueados por um portão ainda contam para a cota de Número máximo de execuções simultâneas de fluxo de trabalho por espaço e outras cotas. Para garantir que você não exceda as cotas de fluxo de trabalho, pense em usar um gatilho de fluxo de trabalho para iniciar condicionalmente um fluxo de trabalho em vez de usar um portão. Pense também em usar uma regra de aprovação de solicitação pull em vez de um portão. Para ter mais informações sobre cotas, gatilhos e regras de aprovação de solicitação pull, consulte [Cotas para fluxos de trabalho em CodeCatalyst](#), [Início da execução automática de um fluxo de trabalho usando gatilhos](#) e [Gerenciamento de requisitos para mesclar uma solicitação pull com regras de aprovação](#).

## Limitações de portões

Os portões têm as seguintes limitações:

- Portões não podem ser usados com o recurso de compartilhamento de computação. Para ter mais informações sobre esse recurso, consulte [Compartilhamento de computação entre ações](#).
- Portões não podem ser usados em grupos de ação. Para ter mais informações sobre os grupos de ações, consulte [Agrupar ações em grupos de ações](#).

## Adicionar um portão a um fluxo de trabalho

No Amazon CodeCatalyst, você pode adicionar um portão a um fluxo de trabalho para impedir que ele continue, a menos que determinadas condições sejam atendidas. Use as instruções a seguir para adicionar um portão a um fluxo de trabalho.

Para ter mais informações sobre portões, consulte [Bloquear uma execução de fluxo de trabalho](#).

Como adicionar e configurar um portão

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Selecione Editar.
6. Selecione Visual.
7. No painel esquerdo, selecione Portões.
8. No catálogo de portões, pesquise um portão e escolha o sinal de adição (+) para adicioná-lo ao seu fluxo de trabalho.
9. Configure o portão. Selecione Visual para usar o editor visual ou YAML para usar o editor YAML. Para ter instruções detalhadas, consulte:
  - [Adição de um portão de “aprovação”](#)
10. (Opcional) Selecione Validar para garantir que o código YAML seja válido.
11. Selecione Confirmar para confirmar suas alterações.

## Sequenciar portões e ações

No Amazon CodeCatalyst, você pode configurar um portão para ser executado antes ou depois de uma ação de fluxo de trabalho, grupo de ação ou portão. Por exemplo, é possível configurar um portão `Approval` para ser executado antes de uma ação `Deploy`. Nesse caso, a ação `Deploy` depende do portão `Approval`.

Para configurar dependências entre portões e ações, configure a propriedade `Depende de` do portão ou ação. Para instruções, consulte [Configurar dependências entre ações](#). As instruções referidas dizem respeito às ações do fluxo de trabalho, mas se aplicam igualmente aos portões.

Para ter um exemplo de como configurar a propriedade `Depende de` com um portão, consulte [Exemplo: um portão de “aprovação”](#).

Para ter mais informações sobre portões, consulte [Bloquear uma execução de fluxo de trabalho](#).

Para ter mais informações sobre ações de fluxos de trabalho, consulte [Configuração de ações de fluxo de trabalho](#).

## Especificar a versão de um portão

Por padrão, quando você adiciona um portão a um fluxo de trabalho, o CodeCatalyst adiciona a versão completa ao arquivo de definição do fluxo de trabalho usando o formato:

*vmajor.minor.patch*

Por exemplo:

```
My-Gate:
 Identifier: aws/approval@v1
```

Você pode aumentar a versão para que o fluxo de trabalho use uma versão principal ou secundária específica do portão. Para instruções, consulte [Especificação da versão da ação a ser usada](#). O tópico referido diz respeito às ações do fluxo de trabalho, mas se aplica igualmente aos portões.

Para ter mais informações sobre portões no CodeCatalyst, consulte [Bloquear uma execução de fluxo de trabalho](#).

## Solicitar aprovações em execuções de fluxo de trabalho

É possível configurar a execução de um fluxo de trabalho para solicitar uma aprovação antes de continuar. Para fazer isso, adicione um [portão](#) de aprovação ao fluxo de trabalho. Um portão de aprovação impede que um fluxo de trabalho continue até que um usuário ou um conjunto de usuários envie uma ou mais aprovações no console do CodeCatalyst. Depois que todas as aprovações são concedidas, o portão é “desbloqueado” e a execução do fluxo de trabalho pode ser retomada.

Use um portão de aprovação no fluxo de trabalho para fornecer às equipes de desenvolvimento, de operações e de liderança a chance de revisar suas alterações antes que elas sejam implantadas em um público mais amplo.

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

### Tópicos

- [Como faço para desbloquear um portão de aprovação?](#)
- [Quando usar o portão de “Aprovação”](#)
- [Quem pode fornecer uma aprovação?](#)
- [Como faço para notificar os usuários de que uma aprovação é necessária?](#)
- [Posso usar um portão de “aprovação” para impedir o início da execução de um fluxo de trabalho?](#)
- [Como as aprovações de fluxo de trabalho funcionam com os modos de execução em fila, substituída e paralela?](#)
- [Exemplo: um portão de “aprovação”](#)

- [Adição de um portão de “aprovação”](#)
- [Configuração de notificações de aprovação](#)
- [Aprovação ou rejeição da execução de um fluxo de trabalho](#)
- [Portão de “Aprovação” YAML](#)

## Como faço para desbloquear um portão de aprovação?

Para desbloquear um portão de aprovação, todas as seguintes condições devem ser atendidas:

- Condição 1: o número necessário de aprovações deve ser enviado. O número necessário de aprovações é configurável e cada usuário pode enviar uma única aprovação.
- Condição 2: todas as aprovações devem ser enviadas antes que o portão expire. O portão expira 14 dias após ser ativado. Esse período não é configurável.
- Condição 3: ninguém deve rejeitar a execução do fluxo de trabalho. Uma única rejeição fará com que a execução do fluxo de trabalho falhe.
- Condição 4: (aplicada apenas se você estiver usando o modo de execução substituído.) A execução não deve ser substituída por uma execução posterior. Para obter mais informações, consulte [Como as aprovações de fluxo de trabalho funcionam com os modos de execução em fila, substituída e paralela?](#).

Se alguma das condições não for atendida, o CodeCatalyst interromperá o fluxo de trabalho e definirá o status de execução como Reprovada (no caso das Condições 1 a 3) ou Substituída (no caso da Condição 4).

## Quando usar o portão de “Aprovação”

Normalmente, você usaria um portão de aprovação em um fluxo de trabalho que implante aplicações e outros recursos em um servidor de produção ou em qualquer ambiente em que os padrões de qualidade devam ser validados. Ao colocar o portão antes da implantação para produção, você oferece aos revisores a chance de validar sua nova revisão de software antes que ela se torne disponível ao público.

## Quem pode fornecer uma aprovação?

Qualquer usuário que seja membro do seu projeto e que tenha o perfil colaborador ou administrador do projeto pode fornecer uma aprovação. Os usuários com o perfil administrador do espaço que pertencem ao espaço do projeto também podem fornecer uma aprovação.

**Note**

Usuários com o perfil revisor não podem fornecer aprovações.

## Como faço para notificar os usuários de que uma aprovação é necessária?

Para notificar os usuários de que uma aprovação é necessária, você deve:

- Fazer com que o CodeCatalyst envie a eles uma notificação do Slack. Para obter mais informações, consulte [Configuração de notificações de aprovação](#).
- Acesse a página no console do CodeCatalyst onde estão os botões Aprovar e Rejeitar e cole o URL dessa página em uma aplicação de e-mail ou de mensagem endereçado aos aprovadores. Para ter mais informações sobre como acessar essa página, consulte [Aprovação ou rejeição da execução de um fluxo de trabalho](#).

## Posso usar um portão de “aprovação” para impedir o início da execução de um fluxo de trabalho?

Sim, com qualificações. Para obter mais informações, consulte [Posso usar um portão para impedir o início da execução de um fluxo de trabalho?](#).

## Como as aprovações de fluxo de trabalho funcionam com os modos de execução em fila, substituída e paralela?

Ao usar o modo de execução em fila, substituída ou paralela, o portão de aprovação funciona de forma semelhante às [ações](#). Sugerimos a leitura das seções [Sobre o modo de execução em fila](#), [Sobre o modo de execução substituída](#) e [Sobre o modo de execução paralela](#) para conhecer esses modos de execução. Depois de ter uma compreensão básica deles, retorne a esta seção para descobrir como esses modos de execução funcionam quando o portão de aprovação está presente.

Quando o portão de aprovação está presente, as execuções são processadas da seguinte forma:

- Se você estiver usando o [modo de execução em fila](#), as execuções ficarão na fila atrás da execução que está aguardando aprovação no portão. Quando esse portão é desbloqueado (ou seja, todas as aprovações foram concedidas), a próxima execução na fila avança até o portão e aguarda as aprovações. Esse processo continua com as execuções em fila sendo processadas pelo portão uma a uma. [Figure 1](#) ilustra esse processo.

- Se você estiver usando o [modo de execução substituída](#), o comportamento será o mesmo que o modo de execução em fila, exceto que, em vez de as execuções se acumularem na fila do portão, as execuções mais recentes substituirão as anteriores. Não há filas, e qualquer execução que esteja aguardando aprovação no portão será cancelada e substituída por uma mais recente. [Figure 2](#) ilustra esse processo.
- Se você estiver usando o [modo de execução paralela](#), as execuções começarão em paralelo e não formarão filas. Cada execução é processada pelo portão imediatamente, pois não há nenhuma execução à sua frente. [Figure 3](#) ilustra esse processo.

Figura 1: “Modo de execução em fila” e um portão de aprovação

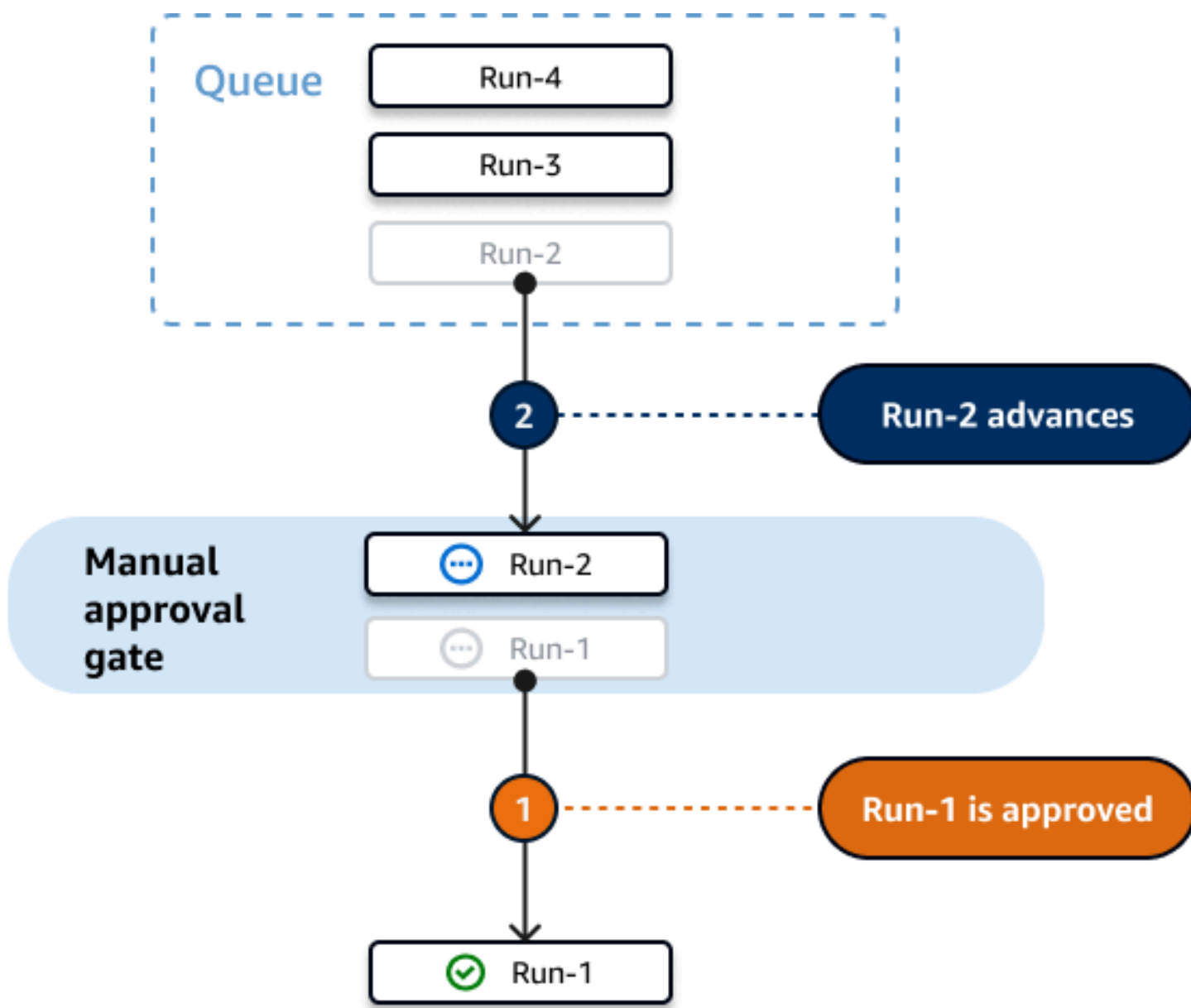




Figura 2: “Modo de execução substituída” e um portão de aprovação

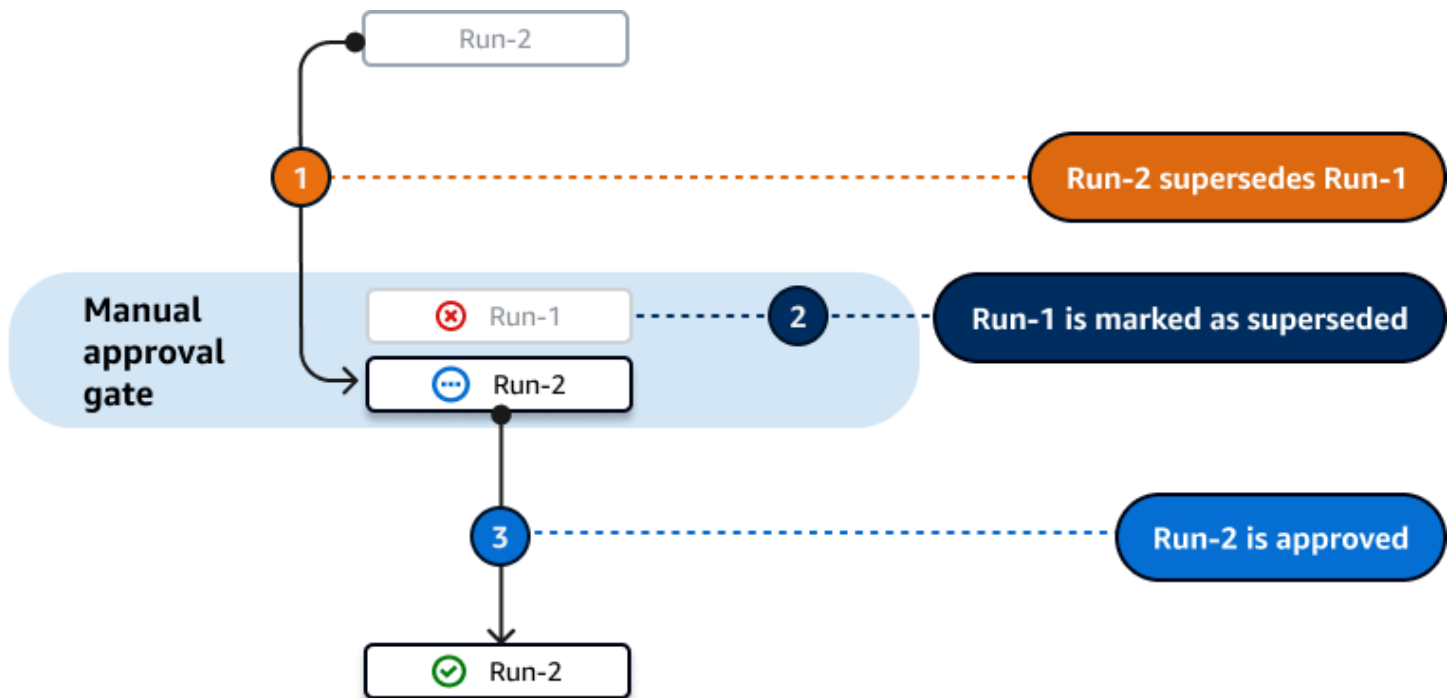
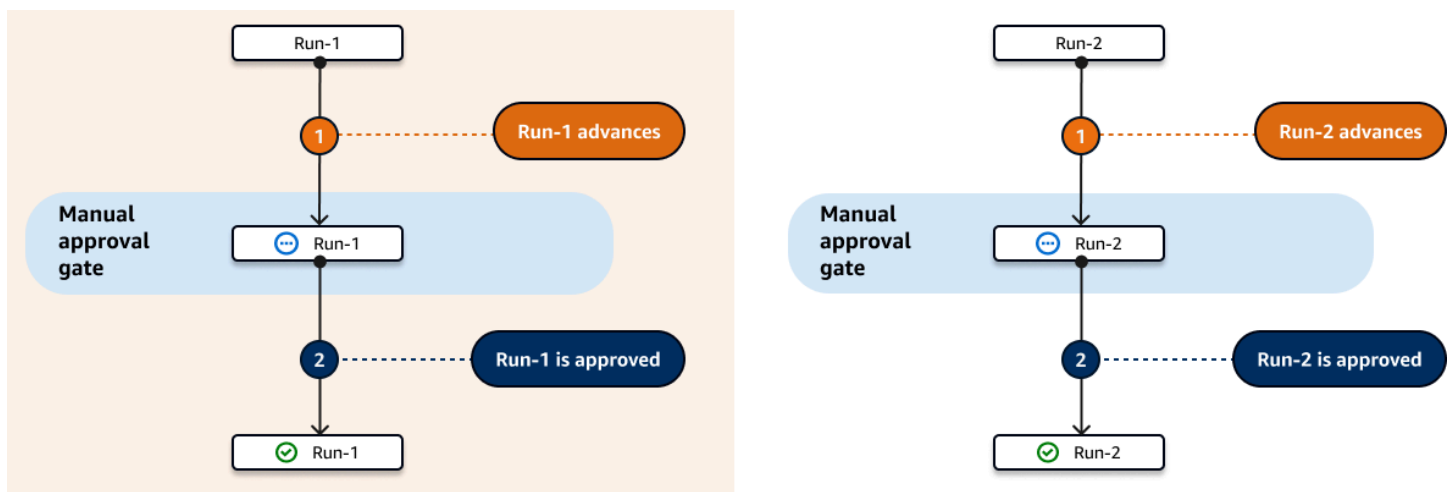


Figura 3: “Modo de execução paralela” e um portão de aprovação



### Exemplo: um portão de “aprovação”

O exemplo a seguir mostra como adicionar um portão de aprovação chamado `Approval_01` entre duas ações chamadas `Staging` e `Production`. A ação `Staging` é executada primeiro, o portão `Approval_01` depois e a ação `Production` por último. A ação `Production` só será executada se o portão `Approval_01` estiver desbloqueado. A propriedade `DependsOn` garante que as fases `Staging`, `Approval_01` e `Production` sejam executadas em ordem sequencial.

Para ter mais informações sobre o portão de aprovação, consulte [Solicitar aprovações em execuções de fluxo de trabalho](#).

**Actions:**

```
Staging: # Deploy to a staging server
 Identifier: aws/ecs-deploy@v1
 Configuration:
 ...
Approval_01:
 Identifier: aws/approval@v1
 DependsOn:
 - Staging
 Configuration:
 ApprovalsRequired: 2
Production: # Deploy to a production server
 Identifier: aws/ecs-deploy@v1
 DependsOn:
 - Approval_01
 Configuration:
 ...
```

## Adição de um portão de “aprovação”

Para configurar seu fluxo de trabalho para exigir uma aprovação, adicione o portão de aprovação ao fluxo de trabalho. Use as instruções a seguir para adicionar um portão de aprovação ao seu fluxo de trabalho.

Para ter mais informações sobre esse portão, consulte [Solicitar aprovações em execuções de fluxo de trabalho](#).

### Visual

Como adicionar um portão de “aprovação” a um fluxo de trabalho (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

5. Escolha Editar.
6. No canto superior esquerdo, selecione Portões.
7. No catálogo Portões, em Aprovação, selecione o sinal de adição (+).
8. Selecione Entradas e, no campo Dependente de, faça o seguinte.

Especifique uma ação, um grupo de ação ou um portão que deve ser executado para que esse portão seja executado. Por padrão, quando você adiciona um portão a um fluxo de trabalho, o portão é configurado para depender da última ação no fluxo de trabalho. Se você remover essa propriedade, o portão não dependerá de nada e será executado primeiro, antes de outras ações.

 **Note**

Um portão deve ser configurado para ser executado antes ou depois de uma ação, um grupo de ação ou um portão. Ele não pode ser configurado para ser executado em paralelo com outras ações, grupos de ação e portões.

Para ter mais informações sobre a funcionalidade Dependente de, consulte [Sequenciar portões e ações](#).

9. Escolha a guia Configuração.
10. No campo Nome do portão, faça o seguinte.

Especifique o nome que você deseja dar ao portão. Todos os nomes de portão devem ser exclusivos no fluxo de trabalho. Os nomes de portão são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de portão.

11. (Opcional) No campo Número de aprovações, faça o seguinte.

Especifique o número mínimo de aprovações necessárias para desbloquear o portão de aprovação. O mínimo é 1. O máximo é 2. Se for omitido, o padrão será 1.

 **Note**

Se você quiser omitir a propriedade `ApprovalsRequired`, remova a seção `Configuration` do portão do arquivo de definição do fluxo de trabalho.

12. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
13. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar um portão de “aprovação” a um fluxo de trabalho (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Adicione uma seção de Approval e propriedades subjacentes usando o exemplo a seguir como guia. Para obter mais informações, consulte o [Portão de “Aprovação” YAML](#) no [Definição do YAML do fluxo de trabalho](#).

```
Actions:
 MyApproval_01:
 Identifier: aws/approval@v1
 DependsOn:
 - PreviousAction
 Configuration:
 ApprovalsRequired: 2
```

Para obter outro exemplo, consulte [Exemplo: um portão de “aprovação”](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Configuração de notificações de aprovação

Você pode fazer com que o CodeCatalyst envie uma notificação para um canal do Slack informando aos usuários que a execução de um fluxo de trabalho exige aprovação. Os usuários veem a notificação e clicam no link dentro dela. O link os leva a uma página de aprovações do CodeCatalyst, onde eles podem aprovar ou rejeitar o fluxo de trabalho.

Você também pode configurar notificações para informar aos usuários que um fluxo de trabalho foi aprovado, rejeitado ou que a solicitação de aprovação expirou.

Use as instruções a seguir para configurar notificações do Slack.

Antes de começar

Certifique-se de ter adicionado um portão de aprovação ao seu fluxo de trabalho. Para obter mais informações, consulte [Adição de um portão de “aprovação”](#).

Para enviar notificações de aprovação do fluxo de trabalho para um canal do Slack

1. Configure o CodeCatalyst com o Slack. Para obter mais informações, consulte [Conceitos básicos das notificações do Slack](#).
2. No projeto CodeCatalyst que contém o fluxo de trabalho que requer aprovação, ative as notificações, se elas ainda não estiverem ativadas. Como ativar notificações:
  - a. Navegue até seu projeto e, no painel de navegação, selecione Configurações do projeto.
  - b. Na parte superior, selecione Notificações.
  - c. Em Eventos de notificação, selecione Editar notificações.
  - d. Ative a Aprovação do fluxo de trabalho pendente e selecione um canal do Slack para o qual o CodeCatalyst enviará a notificação.
  - e. (Opcional) Ative notificações adicionais para alertar as pessoas sobre aprovações aprovadas, rejeitadas e expiradas. Você pode ativar a Execução do fluxo de trabalho aprovada, Execução fluxo de trabalho rejeitada, Aprovação do fluxo de trabalho substituída e Tempo limite da aprovação do fluxo de trabalho. Ao lado de cada notificação, selecione o canal do Slack para o qual o CodeCatalyst enviará a notificação.
  - f. Escolha Salvar.

## Aprovação ou rejeição da execução de um fluxo de trabalho

As execuções de fluxo de trabalho que incluam o portão de aprovação precisarão ser aprovadas ou rejeitadas. Os usuários podem fornecer sua aprovação ou rejeição em:

- console do CodeCatalyst
- link fornecido por um membro da equipe
- notificação automática do Slack

Depois que um usuário fornece sua aprovação ou rejeição, essa decisão não pode ser desfeita.

### Note

Somente alguns usuários podem aprovar ou rejeitar a execução de um fluxo de trabalho. Para obter mais informações, consulte [Quem pode fornecer uma aprovação?](#).

### Antes de começar

Certifique-se de ter adicionado um portão de aprovação ao seu fluxo de trabalho. Para obter mais informações, consulte [Adição de um portão de “aprovação”](#).

Para aprovar ou rejeitar a execução de um fluxo de trabalho a partir do console do CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. No diagrama do fluxo de trabalho, selecione a caixa que representa o portão de aprovação.

Um painel lateral é exibido.

 Note

Neste ponto, você poderá enviar o URL desta página para outros aprovadores, se desejar.

6. Em Revisar decisão, selecione Aprovar ou Rejeitar.
7. (Opcional) Em Comentário – opcional, insira um comentário indicando por que você aprovou ou rejeitou a execução do fluxo de trabalho.
8. Selecione Enviar.

Para aprovar ou rejeitar a execução de um fluxo de trabalho a partir de um link fornecido por um membro da equipe

1. Selecione o link enviado a você pelo membro da sua equipe. (Você pode fazer com que seu membro da equipe leia o procedimento anterior para obter o link.)
2. Faça login no CodeCatalyst, se solicitado.

Você será redirecionado para a página de aprovação de execução de fluxo de trabalho.

3. Em Revisar decisão, selecione Aprovar ou Rejeitar.
4. (Opcional) Em Comentário – opcional, insira um comentário indicando por que você aprovou ou rejeitou a execução do fluxo de trabalho.
5. Selecione Enviar.

Para aprovar ou rejeitar a execução de um fluxo de trabalho a partir de uma notificação automática do Slack

1. Certifique-se de que as notificações do Slack estejam configuradas. Consulte [Configuração de notificações de aprovação](#).
2. No Slack, no canal para o qual a notificação de aprovação foi enviada, selecione o link na notificação de aprovação.
3. Faça login no CodeCatalyst, se solicitado.

Você será redirecionado para a página de execução de fluxo de trabalho.

4. No diagrama do fluxo de trabalho, selecione o portão de aprovação.

5. Em Revisar decisão, selecione Aprovar ou Rejeitar.
6. (Opcional) Em Comentário – opcional, insira um comentário indicando por que você aprovou ou rejeitou a execução do fluxo de trabalho.
7. Selecione Enviar.

## Portão de “Aprovação” YAML

Confira a seguir a definição YAML do portão de Aprovação. Para saber como usar esse portão, consulte [Solicitar aprovações em execuções de fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The 'Approval' gate definition starts here.
Approval:
 Identifier: aws/approval@v1
 DependsOn:
 - another-action
 Configuration:
 ApprovalsRequired: number
```

## Approval

(Obrigatório)



Especifique o nome que você deseja dar ao portão. Todos os nomes de portão devem ser exclusivos no fluxo de trabalho. Os nomes de portão são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de portão.

Padrão: Approval\_nn.

Interface de usuário correspondente: guia Configuração/Nome do portão

Identifier

(Approval/Identifier)

(Obrigatório)

Identifica o portão. O portão de Aprovação comporta a versão 1.0.0. Não altere essa propriedade, a menos que você queira encurtar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/approval@v1.

Interface de usuário correspondente: rótulo do diagrama de fluxo de trabalho/Approval\_nn/aws/approval@v1

DependsOn

(Approval/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado para que esse portão seja executado. Por padrão, quando você adiciona um portão a um fluxo de trabalho, o portão é configurado para depender da última ação no fluxo de trabalho. Se você remover essa propriedade, o portão não dependerá de nada e será executado primeiro, antes de outras ações.

 Note

Um portão deve ser configurado para ser executado antes ou depois de uma ação, um grupo de ação ou um portão. Ele não pode ser configurado para ser executado em paralelo com outras ações, grupos de ação e portões.

Para ter mais informações sobre a funcionalidade Dependência, consulte [Sequenciar portões e ações](#).

Interface de usuário correspondente: guia Entradas/Depende de

Configuration

(*Approval*/Configuration)

(Optional)

Uma seção na qual você pode definir as propriedades de configuração do portão.

Interface de usuário correspondente: guia Configuração

ApprovalsRequired

(*Approval*/Configuration/ApprovalsRequired)

(Optional)

Especifique o número mínimo de aprovações necessárias para desbloquear o portão de Aprovação. O mínimo é 1. O máximo é 2. Se for omitido, o padrão será 1.

 Note

Se você quiser omitir a propriedade ApprovalsRequired, remova a seção Configuration do portão do arquivo de definição do fluxo de trabalho.

Interface de usuário correspondente: guia Configuração/Número de aprovações

## Configurar o comportamento de enfileiramento das execuções

Por padrão, no Amazon CodeCatalyst, quando várias execuções do fluxo de trabalho ocorrem simultaneamente, o CodeCatalyst as coloca em fila e as processa uma a uma, na ordem em que foram iniciadas. Esse comportamento padrão pode ser alterado especificando um modo de execução. Existem alguns modos de execução:

- (Padrão) Modo de execução em fila: os processos do CodeCatalyst são executados um por um.
- Modo de execução substituída: os processos do CodeCatalyst são executados um por um, com execuções mais recentes substituindo as mais antigas.

- Modo de execução paralela: os processos do CodeCatalyst são executados paralelamente.

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

## Tópicos

- [Sobre o modo de execução em fila](#)
- [Sobre o modo de execução substituída](#)
- [Sobre o modo de execução paralela](#)
- [Configurar o modo de execução](#)

## Sobre o modo de execução em fila

No modo de execução em fila, as execuções ocorrem em série, com as execuções em espera formando uma fila.

As filas se formam nos pontos de entrada para ações e grupos de ações, para que você possa ter várias filas no mesmo fluxo de trabalho (consulte [Figure 1](#)). Quando uma execução em fila aciona uma ação, ela é bloqueada e nenhuma outra execução pode ser realizada. Quando a execução termina e sai da ação, ela fica desbloqueada e pronta para a próxima execução.

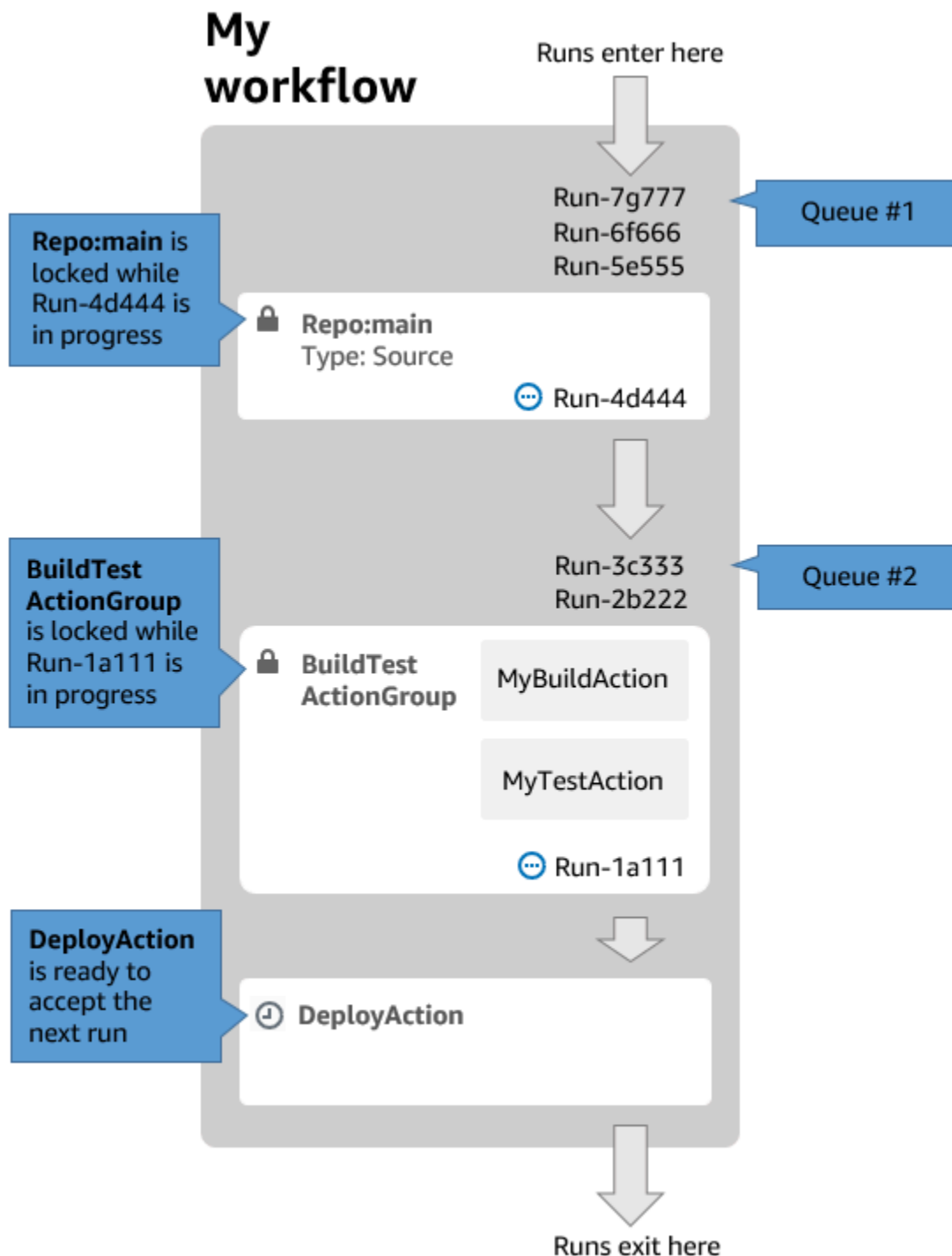
[Figure 1](#) ilustra um fluxo de trabalho configurado no modo de execução em fila. Mostra:

- Sete execuções percorrendo o fluxo de trabalho.
- Duas filas: uma fora da entrada da fonte de entrada (Repo:main) e outra fora da entrada da ação BuildTestActionGroup.
- Dois blocos bloqueados: a origem da entrada (Repo:main) e BuildTestActionGroup.

Veja o que ocorrerá quando o processamento das execuções do fluxo de trabalho for concluído:

- Quando o Run-4d444 terminar de clonar o repositório de origem, ele sairá da origem da entrada e entrará na fila atrás de Run-3c333. Depois, Run-5e555 entrará na origem da entrada.
- Quando Run-1a111 terminar de compilar e testar, ele sairá da ação BuildTestActionGroup e acionará a ação DeployAction. Depois, o Run-2b222 acionará a ação BuildTestActionGroup.

Figura 1: um fluxo de trabalho configurado em “modo de execução em fila”



Use o modo de execução em fila se:

- Você deseja manter um relacionamento um para um entre recursos e execuções: esses recursos podem ser agrupados quando o modo de substituição é utilizado. Por exemplo, quando você mescla o recurso 1 na confirmação 1, a execução 1 é iniciada, e quando você mescla o recurso 2 na confirmação 2, a execução 2 é iniciada e assim por diante. Se você usar o modo de

substituição em vez do modo em fila, os recursos (e as confirmações) serão agrupados na execução que substituirá as outras.

- Você deseja evitar condições de corrida e problemas inesperados que possam ocorrer no uso do modo paralelo. Por exemplo, se dois desenvolvedores de software, Wang e Saanvi, iniciarem a execução do fluxo de trabalho aproximadamente ao mesmo tempo para implantação em um cluster do Amazon ECS, a execução de Wang poderá iniciar os testes de integração no cluster, enquanto a execução de Saanvi implantará um novo código de aplicação no cluster, fazendo com que os testes de Wang falhem ou testem o código errado. Como outro exemplo, é possível ter um destino sem um mecanismo de bloqueio. Nesse caso, as duas execuções podem substituir as alterações uma da outra de maneiras inesperadas.
- Você deseja limitar a carga nos recursos computacionais que o CodeCatalyst usa para processar as execuções. Por exemplo, se você tiver três ações no fluxo de trabalho, poderá ter no máximo três execuções ao mesmo tempo. A imposição de um limite no número de execuções que podem ocorrer simultaneamente torna o throughput de execução mais previsível.
- Você deseja restringir o número de solicitações feitas a serviços de terceiros pelo fluxo de trabalho. Por exemplo, o fluxo de trabalho pode ter uma ação de criação que inclua instruções para extrair uma imagem do Docker Hub. [O Docker Hub limita o número de solicitações pull](#) que você pode fazer a determinado número por hora por conta, e você será bloqueado se ultrapassar o limite. Usar o modo de execução em fila para diminuir o throughput de execução terá o efeito de gerar menos solicitações para o Docker Hub por hora, limitando assim o potencial de bloqueios e gerando falhas de criação e de execução.

Tamanho máximo da fila: 50

Notas sobre o tamanho máximo da fila:

- O tamanho máximo da fila refere-se ao número máximo de execuções permitidas em todas as filas no fluxo de trabalho.
- Se uma fila tiver mais de 50 execuções, o CodeCatalyst descartará a 51ª execução e as subsequentes.

Comportamento com falha:

Se uma execução deixar de responder enquanto estiver sendo processada por uma ação, as execuções atrás dela serão mantidas na fila até que a ação expire. As ações expiram após uma hora.

Se uma execução falhar dentro de uma ação, a primeira execução em fila após ela poderá continuar.

## Sobre o modo de execução substituída

O modo de execução substituída é o mesmo que o modo de execução em fila, exceto que:

- Se uma execução em fila alcançar outra, a execução posterior substituirá a execução anterior, e a execução anterior será cancelada e marcada como “substituída”.
- Como resultado do comportamento descrito no primeiro item, uma fila só pode incluir uma execução quando o modo de execução substituída é usado.

Usando o fluxo de trabalho em [Figure 1](#) como guia, a aplicação do modo de execução substituída a esse fluxo de trabalho ocasionaria o seguinte:

- Run-7g777 substituiria as outras duas execuções na fila e seria a única execução restante na Fila 1. Run-6f666 e Run-5e555 seriam canceladas.
- Run-3c333 substituiria Run-2b222 e seria a única execução restante na Fila 2. Run-2b222 seria cancelada.

Se desejar, use o modo de execução substituída:

- melhor throughput do que no modo em fila
- ainda menos solicitações em serviços de terceiros do que no modo em fila; isso será vantajoso se o serviço de terceiros tiver limites de taxa, como o Docker Hub.

## Sobre o modo de execução paralela

No modo de execução paralela, as execuções são independentes umas das outras e não esperam que outras execuções sejam concluídas antes de começar. Não há filas e o throughput da execução é limitado apenas pela rapidez com que as ações dentro do fluxo de trabalho são concluídas.

Use o modo de execução paralela em ambientes de desenvolvimento em que cada usuário tem a própria ramificação de recursos e implanta em destinos que não são compartilhados por outros usuários.

### Important

Se você tiver um destino compartilhado em que vários usuários possam implantar, como uma função do Lambda em um ambiente de produção, não use o modo paralelo, pois podem ocorrer condições de corrida. Uma condição de corrida ocorre quando execuções paralelas do fluxo de trabalho tentam alterar um recurso compartilhado ao mesmo tempo, gerando resultados imprevisíveis.

Número máximo de execuções paralelas: mil por espaço do CodeCatalyst

## Configurar o modo de execução

É possível definir o modo de execução como em fila, substituída ou paralela. O padrão é em fila.

Quando você altera o modo de execução de em fila ou substituída para paralela, o CodeCatalyst cancela as execuções que estão na fila e permite que as execuções processadas atualmente por uma ação terminem antes de cancelá-las.

Quando você altera o modo de execução de paralela para em fila ou substituída, o CodeCatalyst permite que todas as execuções paralelas em execução no momento sejam concluídas. Todas as execuções que você iniciar depois de alterar o modo de execução para em fila ou substituída usarão o novo modo.

## Visual

Como alterar o modo de execução usando o editor visual

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. No canto superior direito, selecione Propriedades do fluxo de trabalho.
7. Expanda Avançado e, em Modo de execução, selecione uma das seguintes opções:

- a. Em fila: consulte [Sobre o modo de execução em fila](#)
  - b. Substituída: consulte [Sobre o modo de execução substituída](#)
  - c. Paralela: consulte [Sobre o modo de execução paralela](#)
8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como alterar o modo de execução usando o editor YAML

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Adicione a propriedade RunMode:

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED/SUPERSEDED/PARALLEL
```

Para ter mais informações, consulte a descrição da propriedade RunMode na seção [Propriedades de nível superior](#) de [Definição do YAML do fluxo de trabalho](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.



## Armazenar arquivos em cache entre execuções de fluxo de trabalho

Quando o armazenamento de arquivos em cache está ativado, as ações de compilação e teste salvam os arquivos em disco em um cache e os restauram desse cache em execuções subsequentes do fluxo de trabalho. O armazenamento em cache reduz a latência causada pela criação ou download de dependências que não foram alteradas entre as execuções. CodeCatalyst também suporta caches alternativos, que podem ser usados para restaurar caches parciais contendo algumas das dependências necessárias. Isso ajuda a reduzir os impactos de latência de uma perda de cache.

### Note

O armazenamento em cache de arquivos só está disponível com as ações de CodeCatalyst [criação](#) e [teste](#) da Amazon e somente quando elas estão configuradas para usar o tipo de EC2 [computação](#).

### Tópicos

- [Sobre o armazenamento de arquivos em cache](#)
- [Criar um cache](#)
- [Restrições de armazenamento de arquivos em cache](#)

### Sobre o armazenamento de arquivos em cache

O armazenamento de arquivos em cache permite que você organize os dados em vários caches, cada um referido na propriedade `FileCaching`. Cada cache salva um diretório especificado por determinado caminho. O diretório especificado será restaurado em futuras execuções do fluxo de trabalho. Veja a seguir um exemplo de trecho YAML para armazenamento em cache com vários caches chamados `cacheKey1` e `cacheKey2`.

```
Actions:
 BuildMyNpmApp:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
```

**Steps:**

- Run: `npm install`
- Run: `npm run test`

**Caching:****FileCaching:****cacheKey1:**

Path: `file1.txt`

**RestoreKeys:**

- `restoreKey1`

**cacheKey2:**

Path: `/root/repository`

**RestoreKeys:**

- `restoreKey2`
- `restoreKey3`

 **Note**

CodeCatalyst usa cache multicamada, que consiste em um cache local e um cache remoto. Quando frota provisionada ou máquinas sob demanda encontram uma perda de cache em um cache local, as dependências são restauradas a partir de um cache remoto. Como resultado, algumas ações podem apresentar latência ao baixar um cache remoto.

CodeCatalyst aplica restrições de acesso ao cache para garantir que uma ação em um fluxo de trabalho não possa modificar os caches de um fluxo de trabalho diferente. Isso protege cada fluxo de trabalho de outros que podem enviar dados incorretos que afetem compilações ou implantações. As restrições são aplicadas com escopos de cache que isolam os caches de cada fluxo de trabalho e emparelhamento de ramificações. Por exemplo, `workflow-A` na ramificação `feature-A` tem um cache de arquivo diferente de `workflow-A` na ramificação `feature-B` irmã.

As falhas de cache ocorrem quando um fluxo de trabalho procura um cache de arquivo especificado e não consegue encontrá-lo. Isso pode ocorrer por vários motivos, como quando uma ramificação é criada ou quando um novo cache é referido e ainda não foi criado. Também pode ocorrer quando um cache expira, o que, por padrão, ocorre 14 dias após o último uso. Para mitigar as falhas de cache e aumentar a taxa de acessos ao cache, CodeCatalyst oferece suporte a caches alternativos. Os caches de fallback oferecem uma oportunidade de restaurar caches parciais, que podem ser uma versão mais antiga de um cache. Um cache é restaurado pesquisando primeiro por uma correspondência em `FileCaching` pelo nome da propriedade e, se não for encontrado, `RestoreKeys` será avaliado. Se houver uma falha de cache tanto no nome da propriedade quanto

em todas as opções de `RestoreKeys`, o fluxo de trabalho continuará sendo executado, pois o armazenamento em cache é o melhor esforço e não é garantido.

## Criar um cache

Use as instruções a seguir para adicionar um cache ao fluxo de trabalho.

### Visual

Como adicionar um cache usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação onde deseja adicionar o cache.
8. Escolher configuração.
9. Em Cache de arquivos – opcional, selecione Adicionar cache e insira as informações nos campos, da seguinte forma:

#### Chave

Especifique o nome da propriedade de cache principal. Os nomes de propriedade de cache devem ser exclusivos no fluxo de trabalho. Cada ação pode ter até cinco entradas em `FileCaching`.

#### Path

Especifique o caminho associado ao cache.

#### Chaves de restauração – opcional

Especifique a chave de restauração a ser usada como fallback quando a propriedade do cache principal não puder ser encontrada. Os nomes de chave de restauração devem ser exclusivos no fluxo de trabalho. Cada cache pode ter até cinco entradas em `RestoreKeys`.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como adicionar um cache usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação de fluxo de trabalho, adicione um código semelhante ao seguinte:

```
action-name:
 Configuration:
 Steps: ...
 Caching:
 FileCaching:
 key-name:
 Path: file-path
 # # Specify any additional fallback caches
 # RestoreKeys:
 # - restore-key
```

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Restrições de armazenamento de arquivos em cache

Veja a seguir as restrições para o nome da propriedade e RestoreKeys:

- Os nomes devem ser exclusivos no fluxo de trabalho.
- Os nomes são limitados a caracteres alfanuméricos (A-Z, a-z, 0-9), hifens (-) e sublinhados (\_).
- Os nomes podem ter até 180 caracteres.
- Cada ação pode ter até cinco caches em FileCaching.
- Cada cache pode ter até cinco entradas em RestoreKeys.

Veja a seguir as restrições para caminhos:

- Asteriscos (\*) não são permitidos.
- Os caminhos podem ter até 255 caracteres.

## Visualização do status e detalhes de execução do fluxo de trabalho

No Amazon CodeCatalyst, você pode visualizar o status e os detalhes de uma única execução de fluxo de trabalho ou de várias execuções ao mesmo tempo.

Para ver uma lista de possíveis estados de execução, consulte [Estados de execução do fluxo de trabalho](#).

### Note

Você também pode visualizar o status do fluxo de trabalho, que é diferente do status de execução. Para obter mais informações, consulte [Visualização do status de um fluxo de trabalho](#).

Para ter mais informações sobre execuções de fluxos de trabalho, consulte [Executar um fluxo de trabalho](#).

### Tópicos

- [Visualização do status e detalhes de uma única execução](#)
- [Visualizar o status e detalhes de todas as execuções em seu projeto](#)
- [Visualizar o status e os detalhes de todas as execuções de um fluxo de trabalho específico](#)
- [Visualização de execuções de um fluxo de trabalho no diagrama do fluxo de trabalho](#)

## Visualização do status e detalhes de uma única execução

Talvez você queira visualizar o status e os detalhes de uma única execução de fluxo de trabalho para conferir se ela foi bem-sucedida, para ver em que momento foi concluída ou para ver quem ou o que a iniciou.

Como visualizar o status e os detalhes de uma única execução

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
  2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Abaixo do nome do fluxo de trabalho, selecione Execuções.
  6. Em Histórico de execução, na coluna ID da execução, escolha uma execução. Por exemplo, Run-95a4d.
  7. Abaixo do nome da execução, faça o seguinte:
    - Visual para ver um diagrama do fluxo de trabalho mostrando as ações e o status da execução do fluxo de trabalho (consulte [Estados de execução do fluxo de trabalho](#)). Essa visualização também mostra o repositório de origem e a ramificação usados durante a execução.
- No diagrama do fluxo de trabalho, selecione uma ação para ver detalhes, como logs, relatórios e saídas geradas pela ação durante a execução. As informações exibidas dependem do tipo de ação selecionado. Para ter mais informações sobre como visualizar logs de compilação ou implantação, consulte [Visualização dos resultados de uma ação de criação](#) ou [Visualização dos logs de implantação](#).
- YAML para ver o arquivo de definição do fluxo de trabalho usado para a execução.
  - Artefatos para ver os artefatos produzidos pela execução do fluxo de trabalho. Para obter mais informações sobre artefatos, consulte [Compartilhar artefatos e arquivos entre ações](#).
  - Relatórios para ver os relatórios de teste e outros tipos de relatórios produzidos pela execução do fluxo de trabalho. Para obter mais informações sobre os relatórios, consulte [Tipos de relatório de qualidade](#).
  - Variáveis para ver as variáveis de saída produzidas pela execução do fluxo de trabalho. Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

 Note

Se o fluxo de trabalho principal da execução foi excluído, uma mensagem indicando esse fato será exibida na parte superior da página de detalhes da execução.

## Visualizar o status e detalhes de todas as execuções em seu projeto

Talvez você queira visualizar o status e detalhes de todas as execuções de fluxo de trabalho em seu projeto, entender quanta atividade de fluxo de trabalho está acontecendo em seu projeto e aprender sobre a integridade geral de seus fluxos de trabalho.

Para visualizar o status e detalhes de todas as execuções em seu projeto

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Em Fluxos de trabalho, selecione Execuções.

Todas as execuções, para todos os fluxos de trabalho, em todas as ramificações, em todos os repositórios do seu projeto, são exibidas.

A página inclui as seguintes colunas:

- ID de execução – Identificador exclusivo da execução. Selecione o link do ID de execução para visualizar informações detalhadas sobre a execução.
- Status – O status de processamento da execução do fluxo de trabalho. Para ter mais informações sobre os status de execução, consulte [Estados de execução do fluxo de trabalho](#).
- Gatilho: a pessoa, a confirmação e a solicitação pull (PR) ou a programação que iniciou a execução do fluxo de trabalho. Para obter mais informações, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Fluxo de trabalho – O nome do fluxo de trabalho para o qual a execução foi iniciada e o repositório de origem e a ramificação em que o arquivo de definição do fluxo de trabalho reside. Talvez seja necessário expandir a largura da coluna para ver essas informações.

 Note

Se essa coluna estiver definida como Não disponível, geralmente é porque o fluxo de trabalho associado foi excluído ou movido.

- Hora de início – Hora em que a execução do fluxo de trabalho foi iniciada.
- Duração – Quanto tempo a execução do fluxo de trabalho levou para ser processada. Durações muito longas ou muito curtas podem indicar problemas.
- Hora de término – Hora em que a execução do fluxo de trabalho foi finalizada.

## Visualizar o status e os detalhes de todas as execuções de um fluxo de trabalho específico

Talvez você queira visualizar o status e os detalhes de todas as execuções associadas a um fluxo de trabalho específico para ver se alguma execução está criando gargalos no fluxo de trabalho ou para ver quais execuções estão atualmente em andamento ou foram concluídas.

Para visualizar o status e detalhes de todas as execuções de um fluxo de trabalho específico

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Abaixo do nome do fluxo de trabalho, selecione Execuções.

As execuções associadas ao fluxo de trabalho selecionado são exibidas.

A página é dividida em duas seções:

- Execuções ativas – Exibe as execuções que estão em andamento. Essas execuções estarão em um dos seguintes estados: Em andamento.
- Histórico de execuções – Exibe as execuções que foram concluídas (ou seja, que não estão em andamento).



Para ter mais informações sobre os status de execução, consulte [Estados de execução do fluxo de trabalho](#).

## Visualização de execuções de um fluxo de trabalho no diagrama do fluxo de trabalho

Você pode visualizar o status de todas as execuções de um fluxo de trabalho conforme elas progridem juntas no fluxo de trabalho. As execuções são exibidas no diagrama do fluxo de trabalho (em vez de em uma exibição em lista). Isso fornece uma representação visual de quais execuções estão sendo processadas por quais ações e quais execuções estão aguardando em uma fila.

Como visualizar o status de várias execuções conforme elas progridem juntas em um fluxo de trabalho

### Note

Esse procedimento será aplicado apenas se o fluxo de trabalho estiver usando o modo de execução em fila ou de substituição. Para obter mais informações, consulte [Configurar o comportamento de enfileiramento das execuções](#).

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

### Note

Verifique se você está vendo uma página de fluxo de trabalho e não uma página de execução.

5. Selecione a guia Estado mais recente no canto superior esquerdo.

Um diagrama do fluxo de trabalho é exibido.

6. Revise o diagrama do fluxo de trabalho. O diagrama mostra todas as execuções que estão em andamento no fluxo de trabalho e as execuções mais recentes que foram concluídas. Mais especificamente:
- As execuções exibidas na parte superior, antes de Origens, estão na fila aguardando o início.
  - As execuções exibidas entre as ações são colocadas em fila e aguardam para serem processadas pela próxima ação.
  - As execuções que aparecem em uma ação 1. estão atualmente sendo processadas pela ação, 2. terminaram de ser processadas pela ação ou 3. não foram processadas pela ação (geralmente porque uma ação anterior falhou).

## Configuração de ações de fluxo de trabalho

Uma ação é o principal componente de um fluxo de trabalho e define uma unidade lógica de trabalho, ou tarefa, a ser realizada durante a execução de um fluxo de trabalho. Normalmente, um fluxo de trabalho inclui várias ações que são executadas de modo sequencial ou paralelo, dependendo de como você as configurou.

### Tópicos

- [Tipos de ação](#)
- [Adição de uma ação a um fluxo de trabalho](#)
- [Remover uma ação de um fluxo de trabalho](#)
- [Desenvolver uma ação personalizada](#)
- [Agrupar ações em grupos de ações](#)
- [Sequenciar ações](#)
- [Compartilhar artefatos e arquivos entre ações](#)
- [Especificação da versão da ação a ser usada](#)
- [Lista das versões de ação disponíveis](#)
- [Visualizar o código-fonte de uma ação](#)
- [Integração com GitHub ações](#)

## Tipos de ação

Em um CodeCatalyst fluxo de trabalho da Amazon, você pode usar os seguintes tipos de ações.

## Tipos de ação

- [CodeCatalyst ações](#)
- [CodeCatalyst Ações do Labs](#)
- [GitHub Ações](#)
- [Ações de terceiros](#)

## CodeCatalyst ações

Uma CodeCatalyst ação é uma ação criada, mantida e totalmente apoiada pela equipe de CodeCatalyst desenvolvimento.

Existem CodeCatalyst ações para criar, testar e implantar aplicativos, bem como para realizar tarefas diversas, como invocar uma função. AWS Lambda

As seguintes CodeCatalyst ações estão disponíveis:

- **Compilar**

Essa ação cria seus artefatos e executa seus testes de unidade em um contêiner do Docker. Para obter mais informações, consulte [Adição da ação de criação](#).

- **Teste**

Essa ação executa testes de integração e sistema em relação à aplicação ou a artefatos. Para obter mais informações, consulte [Adição da ação de teste](#).

- **Publicação do Amazon S3**

Essa ação copia os artefatos da aplicação para um bucket do Amazon S3. Para obter mais informações, consulte [Publicação de arquivos no Amazon S3 com um fluxo de trabalho](#).

- **AWS CDK bootstrap**

Essa ação provisiona os recursos AWS CDK necessários para implantar seu aplicativo CDK. Para obter mais informações, consulte [Inicializando um AWS CDK aplicativo com um fluxo de trabalho](#).

- **AWS CDK implantar**

Essa ação sintetiza e implanta um aplicativo. AWS Cloud Development Kit (AWS CDK) Para obter mais informações, consulte [Implantando um AWS CDK aplicativo com um fluxo de trabalho](#).

- **AWS Lambda invocar**

Essa ação invoca uma AWS Lambda função. Para obter mais informações, consulte [Invocar uma função do Lambda usando um fluxo de trabalho](#).

- GitHub Ações

Essa ação é uma CodeCatalystação que permite que você execute GitHub ações em um CodeCatalyst fluxo de trabalho. Para obter mais informações, consulte [Invocar uma função do Lambda usando um fluxo de trabalho](#).

- Implante a CloudFormation pilha

Essa ação implanta CloudFormation pilhas. Para obter mais informações, consulte [Implantação de uma pilha CloudFormation](#).

- Implantar no Amazon ECS

Essa ação registra uma definição de tarefa do Amazon ECS e a implanta em um serviço do Amazon ECS. Para obter mais informações, consulte [Implantação no Amazon ECS com um fluxo de trabalho](#).

- Implantar no cluster do Kubernetes

Essa ação implanta uma aplicação em um cluster do Kubernetes. Para obter mais informações, consulte [Implantar no Amazon EKS com um fluxo de trabalho](#).

- Renderizar definição de tarefa do Amazon ECS

Essa ação insere um URI de imagem de contêiner em um arquivo JSON de definição de tarefa do Amazon ECS, criando um novo arquivo de definição de tarefa. Para obter mais informações, consulte [Modificação de uma definição de tarefa do Amazon ECS](#).

A documentação CodeCatalyst das ações está disponível neste guia e no readme de cada ação.

Para obter informações sobre as CodeCatalyst ações disponíveis e como adicioná-las a um fluxo de trabalho, consulte [Adição de uma ação a um fluxo de trabalho](#).

## CodeCatalyst Ações do Labs

Uma ação do CodeCatalyst Labs é uma ação que faz parte do Amazon CodeCatalyst Labs, um campo de testes para aplicações experimentais. CodeCatalyst As ações do Labs foram desenvolvidas para mostrar as integrações com AWS os serviços.

As seguintes ações do CodeCatalyst Labs estão disponíveis:

- Implemente AWS Amplify na hospedagem

Esta ação implanta uma aplicação no Amplify Hosting.

- Implemente em AWS App Runner

Essa ação implanta a imagem mais recente em um repositório de imagens de origem no App Runner.

- Implemente na Amazon CloudFront e no Amazon S3

Essa ação implanta um aplicativo no Amazon S3. CloudFront

- Implemente com AWS SAM

Essa ação implanta a aplicação sem servidor com AWS Serverless Application Model (AWS SAM).

- Invalidar o Amazon Cache CloudFront

Essa ação invalida um CloudFront cache para um determinado conjunto de caminhos.

- Webhook de saída

Essa ação permite que os usuários enviem mensagens dentro de um fluxo de trabalho para um servidor web arbitrário usando uma solicitação HTTPS.

- Publicar em AWS CodeArtifact

Essa ação publica pacotes em um CodeArtifact repositório.

- Publicar no Amazon SNS

Essa ação permite que os usuários se integrem ao Amazon SNS criando um tópico, publicando em um tópico ou assinando um tópico.

- Enviar no Amazon ECR

Essa ação compila e publica uma imagem do Docker em um repositório do Amazon Elastic Container Registry (Amazon ECR).

- Digitalize com a Amazon CodeGuru Security

Essa ação cria um arquivo zip de um caminho de código configurado e usa a CodeGuru Segurança para executar uma varredura de código.

- Terraform Community Edition

Essa ação executa a Terraform Community Edition `plan` e as operações `apply`.

A documentação das ações do CodeCatalyst Labs está disponível no readme de cada ação.

Para obter informações sobre como adicionar uma ação do CodeCatalyst Labs a um fluxo de trabalho e visualizar seu readme, consulte [Adição de uma ação a um fluxo de trabalho](#).

## GitHub Ações

Uma GitHub ação é muito parecida com uma [CodeCatalyst ação](#), exceto pelo fato de ter sido desenvolvida para uso com GitHub fluxos de trabalho. Para obter detalhes sobre GitHub ações, consulte a documentação de [GitHub ações](#).

Você pode usar GitHub ações junto com CodeCatalyst ações nativas em um CodeCatalyst fluxo de trabalho.

Para sua conveniência, o CodeCatalyst console fornece acesso a várias GitHub ações populares. Você também pode usar qualquer GitHub Ação listada no [GitHub Marketplace](#) (sujeita a algumas limitações).

A documentação GitHub das ações está disponível no readme de cada ação.

Para obter mais informações, consulte [Integração com GitHub ações](#).

## Ações de terceiros

Uma ação de terceiros é uma ação de autoria de um fornecedor terceirizado e disponibilizada no console. CodeCatalyst Exemplos de ações de terceiros incluem as ações Mend SCA e SonarCloud Scan, de autoria de Mend e Sonar, respectivamente.

A documentação das ações de terceiros está disponível no readme de cada ação. Documentação adicional também pode ser fornecida pelo fornecedor terceirizado.

Para ter informações sobre como adicionar uma ação de terceiro a um fluxo de trabalho e visualizar seu readme, consulte [Adição de uma ação a um fluxo de trabalho](#).

## Adição de uma ação a um fluxo de trabalho

Use as instruções a seguir para adicionar uma ação a um fluxo de trabalho e depois configurá-lo.

Para adicionar e configurar uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.

3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. No canto superior esquerdo, selecione + Ações para exibir o catálogo de Ações.
7. Na lista suspensa, realize uma das seguintes ações.
  - Escolha Amazon CodeCatalyst para visualizar [CodeCatalyst](#), [CodeCatalyst Labs](#) ou ações [de terceiros](#).
    - CodeCatalyst as ações têm um AWS rótulo por.
    - CodeCatalyst As ações do Labs têm um rótulo by CodeCatalyst Labs.
    - As ações de terceiros têm um *vendor* rótulo “by”, onde *vendor* está o nome do fornecedor terceirizado.
  - Escolha GitHubver uma [lista organizada de GitHub ações](#).
8. No catálogo de ações, procure uma ação e siga um dos seguintes procedimentos:
  - Selecione o sinal de adição (+) para adicionar a ação ao seu fluxo de trabalho.
  - Selecione o nome da ação para visualizar seu readme.
9. Configure a ação. Selecione Visual para usar o editor visual ou YAML para usar o editor YAML. Para ter instruções detalhadas, consulte os seguintes links:

Para obter instruções sobre como adicionar [CodeCatalystações](#), consulte:

- [Adição da ação de criação](#)
- [Adição da ação de teste](#)
- [Adição da ação “Implantar no Amazon ECS”](#)
- [Adição da ação “Implantar no cluster do Kubernetes”](#)
- [Adicionando a ação “Deploy CloudFormation stack”](#)
- [Adicionando a ação 'AWS CDK implantar'](#)
- [Adicionando a ação 'AWS CDK bootstrap'](#)
- [Adicionar a ação “Publicação no Amazon S3”](#)
- [Adicionando a ação 'AWS Lambda invocar'](#)
- [Adicionar a ação “Renderizar definição de tarefa do Amazon ECS”](#)

Para obter instruções sobre como adicionar [ações do CodeCatalyst Labs](#), consulte:

- O readme da ação. Você pode encontrar o readme selecionando o nome da ação no catálogo de ações.

Para obter instruções sobre como adicionar [GitHub ações](#), consulte:

- [Integração com GitHub ações](#)

Para ter instruções sobre como adicionar [ações de terceiros](#), consulte:

- O readme da ação. Você pode encontrar o readme selecionando o nome da ação no catálogo de ações.

10. (Opcional) Selecione Validar para garantir que o código YAML seja válido.

11. Selecione Confirmar para confirmar suas alterações.

## Remover uma ação de um fluxo de trabalho

Use as instruções a seguir para remover uma ação de um fluxo de trabalho.

### Visual

Como remover uma ação usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, na ação que você deseja remover, selecione o ícone de reticências e escolha Remover.



8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como remover uma ação usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Encontre a seção do YAML que contém a ação que você deseja remover.

Selecione a seção e pressione a tecla “delete” no teclado.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Desenvolver uma ação personalizada

Você pode desenvolver uma ação personalizada para usar em seus fluxos de trabalho usando o CodeCatalyst Action Development Kit (ADK). Em seguida, você pode publicar a ação no catálogo de CodeCatalyst ações, para que outros CodeCatalyst usuários possam visualizá-la e usá-la em seus fluxos de trabalho.

Como desenvolver, testar e publicar uma ação (tarefas de alto nível)

1. Instale as ferramentas e os pacotes necessários para desenvolver uma ação.
2. Crie um CodeCatalyst repositório para armazenar seu código de ação.
3. Inicialize a ação. Isso estabelece os arquivos de origem exigidos pela ação, inclusive um arquivo de definição de ação (`action.yml`) que você pode atualizar com o próprio código.

4. Inicialize o código de ação para ter as ferramentas e as bibliotecas necessárias para compilar, testar e lançar o projeto de ação.
5. Crie a ação no seu computador local e envie as alterações para o seu CodeCatalyst repositório.
6. Teste a ação com testes de unidade localmente e execute o fluxo de trabalho gerado pelo ADK no CodeCatalyst.
7. Publique a ação no catálogo de CodeCatalyst ações escolhendo o botão Publicar no CodeCatalyst console.

Para obter etapas detalhadas, consulte o [Guia do desenvolvedor do Amazon CodeCatalyst Action Development Kit](#).

## Agrupar ações em grupos de ações

Um grupo de ações contém uma ou mais ações. O agrupamento de ações em grupos ajuda a manter o fluxo de trabalho organizado e também permite configurar dependências entre grupos diferentes.

### Note

Não é possível aninhar grupos de ações em outros grupos de ações ou ações.

### Tópicos

- [Definir um grupo de ações](#)
- [Exemplo: definir dois grupos de ações](#)

## Definir um grupo de ações

Use as instruções a seguir para definir um grupo de CodeCatalyst ação.

### Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

### YAML

Para definir um grupo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em Actions, adicione um código semelhante ao seguinte:

```
Actions:
 action-group-name:
 Actions:
 action-1:
 Identifier: aws/build@v1
 Configuration:
 ...
 action-2:
 Identifier: aws/build@v1
 Configuration:
 ...
```

Para obter outro exemplo, consulte [Exemplo: definir dois grupos de ações](#). Para ter mais informações, consulte a descrição da propriedade `action-group-name` em [Ações](#) de [Definição do YAML do fluxo de trabalho](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Exemplo: definir dois grupos de ações

O exemplo a seguir mostra como definir dois grupos de CodeCatalyst ação da Amazon: BuildAndTest Deploy e. O grupo BuildAndTest inclui duas ações (Build e Test), e o grupo Deploy também inclui duas ações (DeployCloudFormationStack e DeployToECS).

```
Actions:
 BuildAndTest: # Action group 1
 Actions:
```

```
Build:
 Identifier: aws/build@v1
 Configuration:
 ...
Test:
 Identifier: aws/managed-test@v1
 Configuration:
 ...
Deploy: #Action group 2
Actions:
 DeployCloudFormationStack:
 Identifier: aws/cfn-deploy@v1
 Configuration:
 ...
 DeployToECS:
 Identifier: aws/ecs-deploy@v1
 Configuration:
 ...
```

## Sequenciar ações

Por padrão, quando você adiciona ações a um fluxo de trabalho, elas são adicionadas lado a lado ao [editor visual](#). Isso significa que as ações serão executadas paralelamente quando você iniciar a execução de um fluxo de trabalho. Se você quiser que as ações sejam executadas em ordem sequencial (e apareçam verticalmente no editor visual), configure dependências entre elas. Por exemplo, você pode configurar uma ação Test para depender da ação Build para que a ação de teste seja executada após a ação de criação.

É possível configurar dependências entre ações e grupos de ações. Você também pode configurar one-to-many dependências para que uma ação dependa de várias outras para ser iniciada. Consulte as diretrizes em [Configurar dependências entre ações](#) para garantir que sua configuração de dependência esteja em conformidade com a sintaxe YAML do fluxo de trabalho.

### Tópicos

- [Exemplos de como configurar dependências entre ações](#)
- [Configurar dependências entre ações](#)

## Exemplos de como configurar dependências entre ações

Os exemplos a seguir mostram como configurar dependências entre ações e grupos no arquivo de definição do fluxo de trabalho.

## Tópicos

- [Exemplo: configurar uma dependência simples](#)
- [Exemplo: configurar um grupo de ações para depender de uma ação](#)
- [Exemplo: configurar um grupo de ações para depender de outro grupo de ações.](#)
- [Exemplo: configurar um grupo de ações para depender de várias ações.](#)

### Exemplo: configurar uma dependência simples

O exemplo a seguir mostra como configurar uma ação Test para depender da ação Build usando a propriedade DependsOn.

```
Actions:
 Build:
 Identifier: aws/build@v1
 Configuration:
 ...
 Test:
 DependsOn:
 - Build
 Identifier: aws/managed-test@v1
 Configuration:
 ...
```

### Exemplo: configurar um grupo de ações para depender de uma ação

O exemplo a seguir mostra como configurar uma ação DeployGroup para depender da ação FirstAction. Observe que a ação e o grupo de ações estão no mesmo nível.

```
Actions:
 FirstAction: #An action outside an action group
 Identifier: aws/github-actions-runner@v1
 Configuration:
 ...
 DeployGroup: #An action group containing two actions
 DependsOn:
 - FirstAction
 Actions:
 DeployAction1:
 ...
```

```
DeployAction2:
...
```

Exemplo: configurar um grupo de ações para depender de outro grupo de ações.

O exemplo a seguir mostra como configurar um grupo de ações `DeployGroup` para depender do grupo de ações `BuildAndTestGroup`. Observe que os grupos de ações estão no mesmo nível.

```
Actions:
 BuildAndTestGroup: # Action group 1
 Actions:
 BuildAction:
 ...
 TestAction:
 ...
 DeployGroup: #Action group 2
 DependsOn:
 - BuildAndTestGroup
 Actions:
 DeployAction1:
 ...
 DeployAction2:
 ...
```

Exemplo: configurar um grupo de ações para depender de várias ações.

O exemplo a seguir mostra como configurar um grupo de ações `DeployGroup` para depender da ação `FirstAction`, da ação `SecondAction` e do grupo de ações `BuildAndTestGroup`. Observe que `DeployGroup` está no mesmo nível de `FirstAction`, `SecondAction` e `BuildAndTestGroup`.

```
Actions:
 FirstAction: #An action outside an action group
 ...
 SecondAction: #Another action
 ...
 BuildAndTestGroup: #Action group 1
 Actions:
 Build:
 ...
 Test:
 ...
```

```
DeployGroup: #Action group 2
 DependsOn:
 - FirstAction
 - SecondAction
 - BuildAndTestGroup
 Actions:
 DeployAction1:
 ...
 DeployAction2:
 ...
```

## Configurar dependências entre ações

Use as instruções a seguir para configurar dependências entre ações em um fluxo de trabalho.

Ao configurar dependências, siga estas diretrizes:

- Se uma ação estiver dentro de um grupo, ela só pode depender de outras ações dentro do mesmo grupo.
- As ações e os grupos de ações podem depender de outras ações e grupos de ações no mesmo nível na hierarquia YAML, mas não em um nível diferente.

### Visual

Como configurar dependências usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação que dependerá de outra ação.
8. Escolha a guia Entradas.
9. Em Depende de – opcional, faça o seguinte:

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como configurar dependências usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação que dependerá de outra, adicione um código semelhante ao seguinte:

```
action-name:
 DependsOn:
 - action-1
```

Para obter mais exemplos, consulte [Exemplos de como configurar dependências entre ações](#). Para conhecer as diretrizes gerais, consulte [Configurar dependências entre ações](#). Para ter mais informações, consulte a descrição da propriedade DependsOn em [Definição do YAML do fluxo de trabalho](#) para a ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.



## Compartilhar artefatos e arquivos entre ações

Artefato é a saída de uma ação de fluxo de trabalho e geralmente consiste em uma pasta ou um arquivamento de arquivos. Os artefatos são importantes porque permitem que você compartilhe arquivos e informações entre as ações.

Por exemplo, você pode ter uma ação de criação que gera um arquivo `sam-template.yml`, mas deseja que uma ação de implantação a use. Nesse cenário, você usaria um artefato para permitir que a ação de criação compartilhasse o arquivo `sam-template.yml` com a ação de implantação. O código pode ser semelhante a este:

```
Actions:
 BuildAction:
 Identifier: aws/build@v1
 Steps:
 - Run: sam package --output-template-file sam-template.yml
 Outputs:
 Artifacts:
 - Name: MYARTIFACT
 Files:
 - sam-template.yml
 DeployAction:
 Identifier: aws/cfn-deploy@v1
 Inputs:
 Artifacts:
 - MYARTIFACT
 Configuration:
 template: sam-template.yml
```

No código anterior, a ação de criação (`BuildAction`) gera um arquivo `sam-template.yml` e o adiciona a um artefato de saída chamado `MYARTIFACT`. Uma ação de implantação subsequente (`DeployAction`) especifica `MYARTIFACT` como uma entrada, dando acesso ao arquivo `sam-template.yml`.

### Tópicos

- [Posso compartilhar artefatos sem especificá-los como saídas e entradas?](#)
- [Posso compartilhar artefatos entre fluxos de trabalho?](#)
- [Exemplos de artefatos](#)
- [Definir um artefato de saída](#)

- [Definir um artefato de entrada](#)
- [Referência de arquivos em um artefato](#)
- [Baixar artefatos](#)

## Posso compartilhar artefatos sem especificá-los como saídas e entradas?

Sim, você pode compartilhar artefatos entre ações sem especificá-los nas seções `Outputs` e `Inputs` do código de YAML de suas ações. Para fazer isso, você deve ativar o compartilhamento de computação. Para ter mais informações sobre o compartilhamento de computação e como especificar artefatos quando ele está ativado, consulte [Compartilhamento de computação entre ações](#).

### Note

Embora o recurso de compartilhamento de computação permita que você simplifique o código YAML do seu fluxo de trabalho eliminando a necessidade das seções `Outputs` e `Inputs`, o recurso tem limitações que você deve conhecer antes de ativá-lo. Para ter informações sobre essas limitações, consulte [Considerações sobre compartilhamento de computação](#).

## Posso compartilhar artefatos entre fluxos de trabalho?

Não, você não pode compartilhar artefatos entre fluxos de trabalho diferentes. No entanto, pode compartilhar artefatos entre ações dentro do mesmo fluxo de trabalho.

## Exemplos de artefatos

Os exemplos a seguir mostram como gerar, inserir e referenciar artefatos no arquivo de definição de fluxo de trabalho do Amazon CodeCatalyst.

### Tópicos

- [Exemplo: geração de um artefato](#)
- [Exemplo: inserir um artefato gerado por outra ação](#)
- [Exemplo: referência de arquivos em vários artefatos](#)
- [Exemplo: referência de um arquivo em um único artefato](#)
- [Exemplo: referência a um arquivo em um artefato quando uma `WorkflowSource` está presente](#)

- [Exemplo: referência a um arquivo em um artefato quando um grupo de ações está presente](#)

### Exemplo: geração de um artefato

O exemplo a seguir mostra como gerar um artefato que inclui dois arquivos .jar.

```
Actions:
 Build:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ARTIFACT1
 Files:
 - build-output/file1.jar
 - build-output/file2.jar
```

### Exemplo: inserir um artefato gerado por outra ação

O exemplo a seguir mostra como gerar um artefato chamado ARTIFACT4 em BuildActionA, e inseri-lo em BuildActionB.

```
Actions:
 BuildActionA:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ARTIFACT4
 Files:
 - build-output/file1.jar
 - build-output/file2.jar
 BuildActionB:
 Identifier: aws/build@v1
 Inputs:
 Artifacts:
 - ARTIFACT4
 Configuration:
```

### Exemplo: referência de arquivos em vários artefatos

O exemplo a seguir mostra como gerar dois artefatos chamados ART5 e ART6 em BuildActionC e, depois, referenciar dois arquivos chamados file5.txt (no artefato ART5) e file6.txt (no artefato ART6) em BuildActionD (em Steps).

**Note**

Para ter mais informações sobre como referenciar arquivos, consulte [Referência de arquivos em um artefato](#).

**Note**

Embora o exemplo mostre o prefixo `$CATALYST_SOURCE_DIR_ART5` que está sendo usado, você pode omiti-lo. Isso ocorre porque ART5 é a entrada primária. Para saber mais sobre a entrada primária, consulte [Referência de arquivos em um artefato](#).

**Actions:****BuildActionC:**

Identifier: aws/build@v1

**Outputs:****Artifacts:**

- Name: ART5

**Files:**

- build-output/file5.txt

- Name: ART6

**Files:**

- build-output/file6.txt

**BuildActionD:**

Identifier: aws/build@v1

**Inputs:****Artifacts:**

- ART5

- ART6

**Configuration:****Steps:**

- run: cd \$CATALYST\_SOURCE\_DIR\_ART5/build-output && cat file5.txt

- run: cd \$CATALYST\_SOURCE\_DIR\_ART6/build-output && cat file6.txt

Exemplo: referência de um arquivo em um único artefato

O exemplo a seguir mostra como gerar um artefato chamado ART7 em BuildActionE e, depois, referenciar file7.txt (no artefato ART7) em BuildActionF (em Steps).

Observe como a referência não exige o prefixo `$CATALYST_SOURCE_DIR_`*artifact-name* na frente do diretório build-output, como acontecia em [Exemplo: referência de arquivos em vários artefatos](#). Isso ocorre porque há somente um item especificado em Inputs.

 Note

Para ter mais informações sobre como referenciar arquivos, consulte [Referência de arquivos em um artefato](#).

```
Actions:
 BuildActionE:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ART7
 Files:
 - build-output/file7.txt
 BuildActionF:
 Identifier: aws/build@v1
 Inputs:
 Artifacts:
 - ART7
 Configuration:
 Steps:
 - run: cd build-output && cat file7.txt
```

Exemplo: referência a um arquivo em um artefato quando uma WorkflowSource está presente

O exemplo a seguir mostra como gerar um artefato chamado ART8 em BuildActionG e, depois, referenciar file8.txt (no artefato ART8) em BuildActionH (em Steps).

Observe como a referência exige o prefixo `$CATALYST_SOURCE_DIR_`*artifact-name*, como aconteceu em [Exemplo: referência de arquivos em vários artefatos](#). Isso ocorre porque há vários itens especificados em Inputs (uma origem e um artefato), então você precisa do prefixo para indicar onde procurar o arquivo.

 Note

Para ter mais informações sobre como referenciar arquivos, consulte [Referência de arquivos em um artefato](#).

```
Actions:
 BuildActionG:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ART8
 Files:
 - build-output/file8.txt
 BuildActionH:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Artifacts:
 - ART8
 Configuration:
 Steps:
 - run: cd $CATALYST_SOURCE_DIR_ART8/build-output && cat file8.txt
```

Exemplo: referência a um arquivo em um artefato quando um grupo de ações está presente

O exemplo a seguir mostra como gerar um artefato chamado ART9 em ActionGroup1, ActionI, e referenciar file9.txt (no artefato ART9) em ActionJ.

Para ter mais informações sobre como referenciar arquivos, consulte [Referência de arquivos em um artefato](#).

```
Actions:
 ActionGroup1:
 Actions:
 ActionI:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ART9
 Files:
```

```
 - build-output/file9.yml
ActionJ:
 Identifier: aws/cfn-deploy@v1
 Inputs:
 Sources:
 - WorkflowSource
 Artifacts:
 - ART9
 Configuration:
 template: /artifacts/ActionGroup1@ActionJ/ART9/build-output/file9.yml
```

## Definir um artefato de saída

Use as instruções a seguir para definir um artefato que você deseja que uma CodeCatalyst ação da Amazon produza. Esse artefato então fica disponível para uso de outras ações.

### Note

Nem todas as ações são compatíveis com artefatos de saída. Para determinar se sua ação é compatível, siga as instruções do editor visual a seguir e veja se a ação inclui um botão de Artefatos de saída na guia Saídas. Se ela incluir, significa que os artefatos de saída são compatíveis.

## Visual

Para definir um artefato de saída usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação que produzirá o artefato.
8. Escolha a guia Outputs.

9. Em Artefatos, selecione Adicionar artefato.
10. Selecione Adicionar artefato e insira as informações nos campos, da seguinte maneira.

#### Nome do artefato de criação

Especifique o nome de um artefato gerado pela ação. Os nomes de artefato devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (\_). Espaços, hifens (-) e outros caracteres especiais não são permitidos. Não é possível usar aspas para habilitar espaços, hifens e outros caracteres especiais em nomes de artefato de saída.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

#### Arquivos produzidos por criação

Especifique os arquivos CodeCatalyst incluídos no artefato que é gerado pela ação. Esses arquivos são gerados pela ação do fluxo de trabalho quando ela é executada e também estão disponíveis no repositório de origem. Os caminhos de arquivo podem residir em um repositório de origem ou em um artefato de uma ação anterior e são relativos ao repositório de origem ou à raiz do artefato. É possível usar padrões glob para especificar caminhos.

Exemplos:

- Para especificar um único arquivo que esteja na raiz do local de compilação ou do local do repositório de origem, use `my-file.jar`.
- Para especificar um único arquivo em um subdiretório, use `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Para especificar todos os arquivos, use `**/*`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.
- Para especificar todos os arquivos e diretórios em um diretório chamado `directory`, use `directory/**/*`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.
- Para especificar todos os arquivos em um diretório chamado `directory`, mas não em nenhum de seus subdiretórios, use `directory/*`.



**Note**

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outro caractere especial, coloque o caminho entre aspas duplas ("" ). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

**Note**

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou origem encontrá-lo. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como definir um artefato de saída usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação de fluxo de trabalho, adicione um código semelhante ao seguinte:

```
action-name:
 Outputs:
 Artifacts:
 - Name: artifact-name
 Files:
 - file-path-1
 - file-path-2
```

Para obter mais exemplos, consulte [Exemplos de artefatos](#). Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Definir um artefato de entrada

Se você quiser usar um artefato gerado por outra CodeCatalyst ação da Amazon, você deve especificá-lo como uma entrada para a ação atual. Talvez você possa especificar vários artefatos como entrada – isso depende da ação. Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para sua ação.

### Note

Você não pode referenciar artefatos de outros fluxos de trabalho.

Use as instruções a seguir para especificar um artefato de outra ação como entrada para a ação atual.

### Pré-requisito

Antes de começar, certifique-se de ter gerado o artefato da outra ação. Para obter mais informações, consulte [Definir um artefato de saída](#). A saída do artefato o torna disponível para uso de outras ações.

## Visual

Para especificar um artefato como entrada para uma ação (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação onde deseja especificar um artefato como entrada.
8. Selecione Entradas.
9. Em Artefatos – opcional, faça o seguinte:

Especifique artefatos de ações anteriores que você deseja fornecer como entrada para essa ação. Esses artefatos já devem estar definidos como artefatos de saída em ações anteriores.

Se você não especificar nenhum artefato de entrada, deverá especificar pelo menos um repositório de origem em *action-name*/Inputs/Sources.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

### Note

Se a lista suspensa Artefatos – opcional não estiver disponível (editor visual) ou se você receber erros ao validar o YAML (editor YAML), talvez seja porque a ação permite apenas uma entrada. Nesse caso, tente remover a entrada de origem.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como especificar um artefato como entrada para uma ação (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Na ação em que você deseja especificar o artefato como entrada, adicione um código semelhante ao seguinte:

```
action-name:
 Inputs:
 Artifacts:
 - artifact-name
```

Para obter mais exemplos, consulte [Exemplos de artefatos](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Referência de arquivos em um artefato

Se você tiver um arquivo que reside em um artefato e precisar fazer referência a ele em uma das ações do fluxo de trabalho do Amazon CodeCatalyst, conclua o procedimento a seguir.

### Note

Consulte também [Fazer referência a arquivos do repositório de origem](#).

## Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

## YAML

Para referenciar arquivos em um artefato (editor YAML)

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Selecione a opção Editar.
6. Selecione YAML.
7. Na ação em que você deseja fazer referência a um arquivo, adicione um código semelhante ao seguinte:

```
Actions:
 My-action:
 Inputs:
 Sources:
 - WorkflowSource
 Artifacts:
 - artifact-name
 Configuration:
 template: artifact-path/path/to/file.yml
```

No código anterior, substitua:

- *artifact-name* pelo nome do artefato.
- *artifact-path* por um valor da tabela a seguir.

|                                                                      |                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Se você estiver adicionando a referência a...                        | Substitua <b><i>artifact-path</i></b> por...                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">Uma ação de criação</a> ou <a href="#">ação de teste</a> | <code>\$CATALYST_SOURCE_DIR_ <b><i>artifact-name</i></b> /</code>                                                                                                                                                                                                                                                                                                                         |
| Todas as outras ações                                                | <code>\$CATALYST_SOURCE_DIR_ <b><i>artifact-name</i></b> /</code><br><br>ou<br><br><code>/artifacts/ <b><i>current-action-name</i></b> /<b><i>artifact-name</i></b> /</code><br><br>ou<br><br>Se a ação atual estiver dentro de um <a href="#">grupo de ações</a> :<br><br><code>/artifacts/ <b><i>current-action-group@current-action-name</i></b> /<b><i>artifact-name</i></b> /</code> |

Para obter exemplos, consulte [Exemplos de artefatos](#).

 **Note**

Você pode omitir o ***artifact-path*** e apenas especificar o caminho do arquivo em relação ao diretório raiz do artefato se:

- A ação em que você está incluindo a referência inclui apenas um item abaixo de Inputs (por exemplo, inclui um artefato de entrada e nenhuma origem).
- O arquivo que você deseja referenciar reside na entrada primária. A entrada primária é WorkflowSource, ou o primeiro artefato de entrada listado, se não houver WorkflowSource.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Baixar artefatos

Você pode baixar e inspecionar artefatos gerados por suas ações de fluxo de trabalho do Amazon CodeCatalyst para fins de solução de problemas. Há dois tipos de artefato que podem ser baixados:

- Artefatos de origem: um artefato que contém um snapshot do conteúdo do repositório de origem tal como ele existia quando a execução foi iniciada.
- Artefatos do fluxo de trabalho: um artefato definido na propriedade Outputs do arquivo de configuração do fluxo de trabalho.

### Como baixar artefatos gerados pelo fluxo de trabalho

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Abaixo do nome do fluxo de trabalho, selecione Execuções.
6. Em Histórico de execução, na coluna ID da execução, escolha uma execução. Por exemplo, Run-95a4d.
7. Abaixo do nome da execução, selecione Artefatos.
8. Ao lado de um artefato, selecione Baixar. Um arquivo é baixado. Seu nome de arquivo consiste em sete caracteres aleatórios.
9. Extraia o arquivo usando um utilitário de extração de arquivos de sua escolha.

## Especificação da versão da ação a ser usada

Por padrão, quando você adiciona uma ação a um fluxo de trabalho, a Amazon CodeCatalyst adiciona a versão completa ao arquivo de definição do fluxo de trabalho usando o formato:

## *vmajor.minor.patch*

Por exemplo:

```
My-Build-Action:
 Identifier: aws/build@v1.0.0
```

Você pode encurtar a versão completa na propriedade `Identifier` para que o fluxo de trabalho sempre use a versão secundária ou de patch mais recente da ação.

Por exemplo, se você especificar:

```
My-CloudFormation-Action:
 Identifier: aws/cfn-deploy@v1.0
```

... e a versão mais recente do patch for `1.0.4`, então a ação usará `1.0.4`. Se uma versão posterior for lançada, digamos `1.0.5`, a ação usará `1.0.5`. Se uma versão secundária for lançada, digamos `1.1.0`, a ação continuará usando `1.0.5`.

Para receber instruções detalhadas sobre a especificação de versões, consulte um dos seguintes tópicos.

Use as instruções a seguir para indicar qual versão de uma ação você deseja que seu fluxo de trabalho use. Especifique a versão principal ou secundária mais recente ou uma versão de patch específica.

Recomendamos o uso da versão secundária ou de patch mais recente de uma ação.

### Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

### YAML

Para configurar um fluxo de trabalho para usar a versão mais recente de uma ação ou uma versão de patch específica

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.



4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Encontre a ação cuja versão você deseja editar.
8. Encontre a propriedade `Identifier` da ação e defina a versão como uma das seguintes:
  - `action-identifier @v major` — Use essa sintaxe para que o fluxo de trabalho use uma versão principal específica e permita que as versões secundárias e de patch mais recentes sejam escolhidas automaticamente.
  - `identificador de ação @v. major minor` — Use essa sintaxe para que o fluxo de trabalho use uma versão secundária específica e permita que a versão mais recente do patch seja escolhida automaticamente.
  - `identificador de ação @v. major minor. patch` — Use essa sintaxe para que o fluxo de trabalho use uma versão de patch específica.

 Note

Se não tiver certeza de qual versão está disponível, consulte [Lista das versões de ação disponíveis](#).

 Note

Não é possível omitir a versão principal.

9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Lista das versões de ação disponíveis

Use as instruções a seguir para determinar quais versões de uma ação estão disponíveis para você usar em um fluxo de trabalho.

## Visual

Para determinar quais versões de ação estão disponíveis

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. Encontre a ação cujas versões você deseja visualizar:
  - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - b. Selecione o nome de qualquer fluxo de trabalho ou crie um. Para ter informações sobre a criação de um fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).
  - c. Escolha Editar.
  - d. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  - e. Na lista suspensa, escolha Amazon CodeCatalyst para visualizar ações CodeCatalyst, CodeCatalyst Labs e de terceiros, ou escolha GitHub para visualizar ações selecionadas GitHub.
  - f. Pesquise uma ação e selecione seu nome. Não selecione o sinal de mais (+).

Os detalhes sobre a ação são exibidos.

4. Na caixa de diálogo de detalhes da ação, no canto superior direito, selecione a lista suspensa Versões para ver uma lista das versões disponíveis da ação.

## YAML

Não disponível. Selecione “visual” para visualizar as instruções do editor visual.

## Visualizar o código-fonte de uma ação

Visualize o código-fonte de uma ação para garantir que ela não contenha código arriscado, vulnerabilidades de segurança ou outros defeitos.

Use as instruções a seguir para visualizar o código-fonte de uma ação [CodeCatalyst](#), do [CodeCatalyst Labs](#) ou [de terceiros](#).

 Note

Para ver o código-fonte de uma [GitHubação](#), acesse a página da ação no [GitHub Marketplace](#). A página inclui um link para o repositório da ação, onde você pode encontrar o código-fonte da ação.

 Note

Você não pode visualizar o código-fonte das seguintes CodeCatalyst ações: [compilar](#), [testar](#), [GitHub Ações](#).

 Note

AWS não suporta nem garante o código de ação de GitHub ações ou ações de terceiros.

## Como visualizar o código-fonte de uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. Encontre a ação cujo código você deseja visualizar:
  - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - b. Selecione o nome de qualquer fluxo de trabalho ou crie um. Para ter informações sobre a criação de um fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).
  - c. Escolha Editar.
  - d. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  - e. Na lista suspensa, escolha Amazon CodeCatalyst para visualizar CodeCatalyst, CodeCatalyst Labs e ações de terceiros.
  - f. Pesquise uma ação e selecione seu nome. Não selecione o sinal de mais (+).

Os detalhes sobre a ação são exibidos.

4. Na caixa de diálogo de detalhes da ação, na parte inferior, selecione Download.

Uma página é exibida com o bucket do Amazon S3 em que o código-fonte da ação está localizado. Para ter informações sobre o Amazon S3, consulte [O que é o Amazon S3?](#) no Guia do usuário do Amazon Simple Storage Service.

5. Inspecione o código para garantir que ele atenda às suas expectativas de qualidade e segurança.

## Integração com GitHub ações

Uma GitHub ação é muito parecida com uma [CodeCatalyst ação](#), exceto pelo fato de ter sido desenvolvida para uso com GitHub fluxos de trabalho. Para obter detalhes sobre GitHub ações, consulte a documentação de [GitHub ações](#).

Você pode usar GitHub ações junto com CodeCatalyst ações nativas em um CodeCatalyst fluxo de trabalho.

Há duas maneiras de adicionar uma GitHub ação a um CodeCatalyst fluxo de trabalho:

- Você pode selecionar a GitHub Ação em uma lista organizada no CodeCatalyst console. Várias GitHub ações populares estão disponíveis. Para obter mais informações, consulte [Adicionando uma ação com curadoria GitHub](#).
- Se a GitHub Ação que você deseja usar não estiver disponível no CodeCatalyst console, você poderá adicioná-la usando uma ação GitHub Ações.

Uma ação de GitHub ações é uma CodeCatalyst ação que envolve uma GitHub ação e a torna compatível com CodeCatalyst fluxos de trabalho.

Aqui está um exemplo de uma ação de GitHub ações envolvendo a ação [GitHubSuper-Linter](#):

```
Actions:
 GitHubAction:
 Identifier: aws/github-actions-runner@v1
 Configuration:
 Steps:
 - name: Lint Code Base
 uses: github/super-linter@v4
 env:
 VALIDATE_ALL_CODEBASE: "true"
 DEFAULT_BRANCH: main
```

No código anterior, a ação CodeCatalyst GitHub Actions (identificada por `aws/github-actions-runner@v1`) envolve a ação Super-Linter (identificada por `github/super-linter@v4`), fazendo com que ela funcione em um fluxo de trabalho. CodeCatalyst

Para obter mais informações, consulte [Adicionando a GitHub ação 'Ações'](#).

Todas as GitHub ações, curadas ou não, devem ser agrupadas dentro de uma ação GitHub Actions (`aws/github-actions-runner@v1`), conforme mostrado no exemplo anterior. O invólucro é necessário para a ação funcionar bem.

## Tópicos

- [Como as GitHub ações são diferentes das CodeCatalyst ações?](#)
- [GitHub As ações podem interagir com outras CodeCatalyst ações no fluxo de trabalho?](#)
- [Quais GitHub ações posso usar?](#)
- [GitHub Limitações das ações em CodeCatalyst](#)
- [Como adiciono uma GitHub ação \(etapas de alto nível\)?](#)
- [A GitHub ação é executada GitHub?](#)
- [Também posso usar GitHub fluxos de trabalho?](#)
- [Imagem de tempo de execução usada pela GitHub ação 'Ações'](#)
- [Tutorial: código Lint usando uma ação do GitHub](#)
- [Adicionando a GitHub ação 'Ações'](#)
- [Adicionando uma ação com curadoria GitHub](#)
- [Exportação dos parâmetros de saída do GitHub](#)
- [Referência aos parâmetros de saída do GitHub](#)
- [GitHub Ação 'Ações' YAML](#)

## Como as GitHub ações são diferentes das CodeCatalyst ações?

GitHub As ações usadas em um CodeCatalyst fluxo de trabalho não têm o mesmo nível de acesso e integração com AWS CodeCatalyst recursos (como [ambientes](#) e [problemas](#)) que CodeCatalyst as ações.

## GitHub As ações podem interagir com outras CodeCatalyst ações no fluxo de trabalho?

Sim. Por exemplo, GitHub as ações podem usar variáveis produzidas por outras CodeCatalyst ações como entrada e também podem compartilhar parâmetros de saída e artefatos com CodeCatalyst ações. Para obter mais informações, consulte [Exportação dos parâmetros de saída do GitHub](#) e [Referência aos parâmetros de saída do GitHub](#).

## Quais GitHub ações posso usar?

Você pode usar qualquer GitHub ação disponível no CodeCatalyst console e qualquer GitHub ação disponível no [GitHubMarketplace](#). Se você decidir usar uma GitHub Ação do Marketplace, tenha em mente as seguintes [limitações](#).

### GitHub Limitações das ações em CodeCatalyst

- GitHub As ações não podem ser usadas com o tipo de [computação CodeCatalyst Lambda](#).
- GitHub As ações são executadas na imagem Docker do ambiente de execução de [novembro de 2022](#), que inclui ferramentas mais antigas. Para ter mais informações sobre a imagem e as ferramentas, consulte [Especificação de imagens de ambiente de runtime](#).
- GitHub Ações que dependem internamente do [githubcontexto](#) ou em que recursos GitHub específicos de referência não funcionarão. CodeCatalyst Por exemplo, as ações a seguir não funcionarão em CodeCatalyst:
  - Ações que tentam adicionar, alterar ou atualizar GitHub recursos. Os exemplos incluem ações que atualizam pull requests ou criam problemas no GitHub.
  - Quase todas as ações estão listadas em <https://github.com/actions>.
- GitHub As ações que são [ações de contêiner do Docker](#) funcionarão, mas devem ser executadas pelo usuário padrão do Docker (root). Não execute a ação como usuário 1001. (No momento em que este artigo foi escrito, o usuário 1001 trabalhava no GitHub, mas não no CodeCatalyst.) Para obter mais informações, consulte o tópico [USUÁRIO](#) em [Suporte do Dockerfile para GitHub ações](#).

Para obter uma lista das GitHub ações disponíveis por meio do CodeCatalyst console, consulte [Adicionando uma ação com curadoria GitHub](#).

## Como adiciono uma GitHub ação (etapas de alto nível)?

As etapas de alto nível para adicionar uma GitHub ação a um CodeCatalyst fluxo de trabalho são as seguintes:

1. No seu CodeCatalyst projeto, você cria um fluxo de trabalho. No fluxo de trabalho, você define como criar, testar e implantar a aplicação. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#).
2. No fluxo de trabalho, você adiciona uma GitHub ação com curadoria ou adiciona a ação GitHub Ações.
3. Realize uma das seguintes ações:
  - Se você optar por adicionar uma ação selecionada, configure-a. Para obter mais informações, consulte [Adicionando uma ação com curadoria GitHub](#).
  - Se você optar por adicionar uma ação não curada, na ação GitHubAções, cole o código YAML da GitHub Ação. Você pode encontrar esse código na página de detalhes da GitHub Ação escolhida no [GitHubMarketplace](#). Provavelmente, você precisará modificar um pouco o código para que ele funcione CodeCatalyst. Para obter mais informações, consulte [Adicionando a GitHub ação 'Ações'](#).
4. (Opcional) No fluxo de trabalho, adicione outras ações, como as ações de criação e teste. Para obter mais informações, consulte [Compilação, teste e implantação com fluxos de trabalho](#).
5. Você inicia o fluxo de trabalho manual ou automaticamente por meio de um gatilho. O fluxo de trabalho executa a GitHub Ação e quaisquer outras ações no fluxo de trabalho. Para obter mais informações, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

Para saber detalhes das etapas, consulte:

- [Adicionando uma ação com curadoria GitHub](#).
- [Adicionando a GitHub ação 'Ações'](#).

## A GitHub ação é executada GitHub?

Não. A GitHub ação é executada em CodeCatalyst, usando a [imagem CodeCatalyst do ambiente de tempo de execução](#).

## Também posso usar GitHub fluxos de trabalho?

Não.

## Imagem de tempo de execução usada pela GitHub ação 'Ações'

A ação CodeCatalyst GitHub Ações é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Tutorial: código Lint usando uma ação do GitHub

Neste tutorial, você adiciona a [ação do GitHub Super-Linter](#) a um fluxo de trabalho do Amazon CodeCatalyst. A ação Super-Linter inspeciona o código, encontra áreas em que o código tem erros, problemas de formatação e estruturas suspeitas e, depois, envia os resultados ao console do CodeCatalyst. Depois de adicionar o linter ao fluxo de trabalho, você executa o fluxo de trabalho para aplicar o código lint a uma aplicação Node.js de exemplo (`app.js`). Depois, você corrige os problemas relatados e executa o fluxo de trabalho novamente para ver se as correções funcionaram.

### Tip

Pense em usar o Super-Linter para aplicar o código lint a arquivos YAML, como [modelos do CloudFormation](#).

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um repositório de origem](#)
- [Etapa 2: adicionar um arquivo `app.js`](#)
- [Etapa 3: criar um fluxo de trabalho que execute a ação Super-Linter](#)
- [Etapa 4: corrigir problemas encontrados pelo Super-Linter](#)
- [Limpeza](#)

## Pré-requisitos

Antes de começar, você precisa de:

- Um espaço do CodeCatalyst com uma Conta da AWS conectada. Para obter mais informações, consulte [Criar um espaço](#).



- Um projeto vazio no espaço do CodeCatalyst chamado `codecatalyst-linter-project`.  
Selecione a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

### Etapa 1: criar um repositório de origem

Nesta etapa, você cria um repositório de origem no CodeCatalyst. Você usará esse repositório para armazenar o arquivo de origem da aplicação de exemplo, `app.js`, para este tutorial.

Para ter mais informações sobre repositórios de origem, consulte [Criar um repositório de origem](#).

#### Como criar um repositório de origem

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto, `codecatalyst-linter-project`.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha Adicionar repositório e selecione Criar repositório.
5. Em Nome do repositório, insira:

**`codecatalyst-linter-source-repository`**

6. Escolha Criar.

### Etapa 2: adicionar um arquivo `app.js`

Nesta etapa, você vai adicionar um arquivo `app.js` ao repositório de origem. O `app.js` contém um código de função que contém alguns erros que o linter encontrará.

#### Para adicionar o arquivo `app.js`

1. No console do CodeCatalyst, escolha seu projeto, `codecatalyst-linter-project`.
2. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
3. Na lista de repositórios de origem, escolha o seu repositório, `codecatalyst-linter-source-repository`.
4. Em Arquivos, selecione Criar arquivo.

## 5. Na caixa de texto, insira o código a seguir:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;
/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 *
 */
exports.lambdaHandler = async (event, context) => {
 try {
 // const ret = await axios(url);
 response = {
 statusCode: 200,
 'body': JSON.stringify({
 message: 'hello world'
 // location: ret.data.trim()
 })
 }
 } catch (err) {
 console.log(err)
 return err
 }

 return response
}
```

## 6. Em Nome do arquivo, insira `app.js`. Mantenha as outras opções padrão.

## 7. Selecione Confirmar.

Agora você criou um chamado `app.js`.

### Etapa 3: criar um fluxo de trabalho que execute a ação Super-Linter

Nesta etapa, você vai criar um fluxo de trabalho que executa a ação Super-Linter ao enviar código para o repositório de origem. O fluxo de trabalho consiste nos seguintes componentes, que você define em um arquivo YAML:

- Um gatilho – Esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação “GitHub Actions” — No gatilho, a ação GitHub Actions executa a ação Super-Linter, que, por sua vez, inspeciona todos os arquivos no repositório de origem. Se o linter encontrar um problema, a ação do fluxo de trabalho falhará.

Para criar um fluxo de trabalho que execute a ação Super-Linter

1. No console do CodeCatalyst, escolha seu projeto, `codecatalyst-linter-project`.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione Criar fluxo de trabalho.
4. Em Repositório de origem, selecione `codecatalyst-linter-source-repository`.
5. Em Ramificação, selecione `main`.
6. Escolha Criar.
7. Exclua o código de amostra YAML.
8. Adicione o seguinte YAML:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 SuperLintAction:
 Identifier: aws/github-actions-runner@v1
 Configuration:
 Steps:
 github-action-code
```

No código anterior, substitua *github-action-code* pelo código de ação Super-Linter, conforme instruído nas etapas a seguir deste procedimento.

9. Acesse a [página Super-Linter](#) no GitHub Marketplace.
10. Em `steps:` (minúsculas), localize o código e cole-o no fluxo de trabalho do CodeCatalyst em `Steps:` (maiúsculas).

Ajuste o código da ação do GitHub de acordo com os padrões do CodeCatalyst, conforme mostrado no código a seguir.

O fluxo de trabalho do CodeCatalyst agora tem a seguinte aparência:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 SuperLinterAction:
 Identifier: aws/github-actions-runner@v1
 Configuration:
 Steps:
 - name: Lint Code Base
 uses: github/super-linter@v4
 env:
 VALIDATE_ALL_CODEBASE: "true"
 DEFAULT_BRANCH: main
```

11. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
12. Selecione Confirmar, insira uma Mensagem de confirmação, escolha seu repositório do `codecatalyst-linter-source-repository` e selecione Confirmar novamente.

Agora você criou um fluxo de trabalho. A execução de um fluxo de trabalho é iniciada automaticamente devido ao gatilho definido na parte superior do fluxo de trabalho.

Para visualizar a execução do fluxo de trabalho em andamento

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar: `codecatalyst-linter-workflow`.

3. No diagrama do fluxo de trabalho, selecione SuperLinterAction.
4. Aguarde a falha da ação. Essa falha é esperada porque o linter encontrou problemas no código.
5. Deixe o console do CodeCatalyst aberto e vá para [Etapa 4: corrigir problemas encontrados pelo Super-Linter](#).

#### Etapa 4: corrigir problemas encontrados pelo Super-Linter

O Super-Linter deve ter encontrado problemas no código `app.js`, bem como no arquivo `README.md` incluído no seu repositório de origem.

Para corrigir os problemas encontrados pelo linter

1. No console do CodeCatalyst, escolha a guia Logs e, depois, selecione Lint Code Base.

Os logs gerados pela ação Super-Linter são exibidos.

2. Nos logs do Super-Linter, role para baixo até a linha 90, onde você encontra o início dos problemas. Eles são semelhantes a:

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body'
found.
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but
found 4.
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not
found.
```

3. Corrija `app.js` e `README.md` no repositório de origem e confirme as alterações.

#### Tip

Para corrigir o `README.md`, adicione markdown ao bloco de código, assim:

```
```markdown  
Setup examples:  
...  
```
```

Suas alterações iniciam outro fluxo de trabalho executado automaticamente. Aguarde a conclusão do fluxo de trabalho. Se você corrigiu todos os problemas, o fluxo de trabalho deverá ser bem-sucedido.

## Limpeza

Limpe no CodeCatalyst para remover os traços deste tutorial do seu ambiente.

Para limpar no CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Exclua `codecatalyst-linter-source-repository`.
3. Exclua `codecatalyst-linter-workflow`.

Neste tutorial, você aprendeu como adicionar a ação do GitHub Super-Linter a um fluxo de trabalho do CodeCatalyst para aplicar lint a alguns códigos.

## Adicionando a GitHub ação 'Ações'

Uma ação de GitHub ações é uma CodeCatalyst ação que envolve uma GitHub ação e a torna compatível com CodeCatalyst fluxos de trabalho.

Para obter mais informações, consulte [Integração com GitHub ações](#).

Para adicionar a ação GitHub Ações a um fluxo de trabalho, siga estas etapas.

### Tip

Para obter um tutorial que mostra como usar a ação GitHub Ações, consulte [Tutorial: código Lint usando uma ação do GitHub](#).

## Visual

Para adicionar a GitHub ação 'Ações' usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.

3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Pesquise a ação GitHub Ações e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

  - Escolha GitHub Ações. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [GitHub Ação 'Ações' YAML](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a GitHub ação 'Ações' usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione YAML.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha. GitHub
  9. Pesquise a ação GitHub Ações e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Escolha GitHub Ações. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
  10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [GitHub Ação 'Ações' YAML](#).
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

### Definição da GitHub ação “Ações”

A ação GitHub Ações é definida como um conjunto de propriedades YAML dentro do seu arquivo de definição de fluxo de trabalho. Para ter mais informações sobre essas propriedades, consulte [GitHub Ação 'Ações' YAML](#) em [Definição do YAML do fluxo de trabalho](#).

### Adicionando uma ação com curadoria GitHub

Uma GitHub ação com curadoria é uma GitHub ação que é disponibilizada no CodeCatalyst console e serve como um exemplo de como usar uma GitHub ação dentro de um CodeCatalyst fluxo de trabalho.



GitHub As ações selecionadas são agrupadas na [ação de GitHub ações CodeCatalyst](#) de autoria, identificada pelo identificador. `aws/github-actions-runner@v1` Por exemplo, aqui está a aparência da GitHub Ação, [TruffleHog OSS](#), com curadoria:

```
Actions:
 TruffleHogOSS_e8:
 Identifier: aws/github-actions-runner@v1
 Inputs:
 Sources:
 - WorkflowSource # This specifies that the action requires this Workflow as a
source
 Configuration:
 Steps:
 - uses: trufflesecurity/trufflehog@v3.16.0
 with:
 path: ' ' # Required; description: Repository path
 base: ' ' # Required; description: Start scanning from here (usually main
branch).
 head: ' ' # Optional; description: Scan commits until here (usually dev
branch).
 extra_args: ' ' # Optional; description: Extra args to be passed to the
trufflehog cli.
```

No código anterior, a ação CodeCatalyst GitHub Ações (identificada por `aws/github-actions-runner@v1`) envolve a ação TruffleHog OSS (identificada por `trufflesecurity/trufflehog@v3.16.0`), fazendo com que ela funcione em um CodeCatalyst fluxo de trabalho.

Para configurar essa ação, você substituiria as strings vazias em `with:` por seus próprios valores. Por exemplo:

```
Actions:
 TruffleHogOSS_e8:
 Identifier: aws/github-actions-runner@v1
 Inputs:
 Sources:
 - WorkflowSource # This specifies that the action requires this Workflow as a
source
 Configuration:
 Steps:
 - uses: trufflesecurity/trufflehog@v3.16.0
 with:
 path: ./
```

```
base: main # Required; description: Start scanning from here (usually main
branch).
head: HEAD # Optional; description: Scan commits until here (usually dev
branch).
extra_args: '--debug --only-verified' # Optional; description: Extra args
to be passed to the truffleshog cli.
```

Para adicionar uma GitHub ação organizada a um fluxo de trabalho, use o procedimento a seguir. Para obter informações gerais sobre o uso de GitHub ações em um CodeCatalyst fluxo de trabalho, consulte [Integração com GitHub ações](#).

#### Note

Se você não vê sua GitHub Ação na lista de ações selecionadas, você ainda pode adicioná-la ao seu fluxo de trabalho usando a ação GitHub Ações. Para obter mais informações, consulte [Adicionando a GitHub ação 'Ações'](#).

## Visual

Para adicionar uma GitHub ação com curadoria usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Procure ou pesquise uma GitHub ação e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha o nome da GitHub ação. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
  - 10. Nas guias Entradas, Configuração e Saídas, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [GitHub Ação 'Ações' YAML](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) disponível para a ação GitHubAções, conforme aparece nos editores YAML e visual.
- Para obter informações sobre as opções de configuração disponíveis para a GitHub Ação selecionada, consulte sua documentação.
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar uma GitHub ação com curadoria usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Procure ou pesquise uma GitHub ação e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha o nome da GitHub ação. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
  - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
  - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- 10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível para a ação GitHub Ações é fornecida no [GitHub Ação 'Ações' YAML](#).

Para obter informações sobre as opções de configuração disponíveis para a GitHub Ação selecionada, consulte sua documentação.

- 11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Exportação dos parâmetros de saída do GitHub

Você pode usar os [parâmetros de saída do GitHub](#) nos fluxos de trabalho do CodeCatalyst.

### Note

Outra palavra para parâmetro de saída é variável. Como o GitHub usa o termo parâmetro de saída na documentação, também usaremos esse termo.

Use as instruções a seguir para exportar um parâmetro de saída do GitHub de uma ação do GitHub para que ele fique disponível para uso por outras ações do fluxo de trabalho do CodeCatalyst.

Para exportar um parâmetro de saída do GitHub

1. Abra um fluxo de trabalho e selecione Editar. Para obter mais informações, consulte [Criação de um fluxo de trabalho](#).
2. Na ação GitHub Actions que gera o parâmetro de saída que você deseja exportar, adicione uma seção `c0utputs` com uma propriedade `Variables` subjacente semelhante a esta:

```
Actions:
 MyGitHubAction:
 Identifier: aws/github-actions-runner@v1
 Outputs:
 Variables:
 - 'step-id_output-name'
```

Substitua:

- **step-id** pelo valor da propriedade `id`: na seção `steps` da ação do GitHub.
- **output-name** pelo nome do parâmetro de saída do GitHub.

## Exemplo

O exemplo a seguir mostra como exportar um parâmetro de saída do GitHub chamado `SELECTEDCOLOR`.

```
Actions:
 MyGitHubAction:
 Identifier: aws/github-actions-runner@v1
 Outputs:
 Variables:
 - 'random-color-generator_SELECTEDCOLOR'
 Configuration:
 Steps:
 - name: Set selected color
 run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
 id: random-color-generator
```

## Referência aos parâmetros de saída do GitHub

Use as instruções a seguir para referenciar um parâmetro de saída do GitHub.

Para fazer referência a um parâmetro de saída do GitHub

1. Siga as etapas em [Exportação dos parâmetros de saída do GitHub](#).

O parâmetro de saída do GitHub agora está disponível para uso em outras ações.

2. Observe o valor `Variables` do parâmetro de saída. Inclui um sublinhado (`_`).

### 3. Consulte o parâmetro de saída usando a seguinte sintaxe:

```
${action-name.output-name}
```

Substitua:

- *action-name* pelo nome da ação do GitHub do CodeCatalyst que produz o parâmetro de saída (não use name ou id da ação do GitHub).
- *output-name* pelo valor Variables do parâmetro de saída que você anotou anteriormente.

#### Exemplo

```
BuildActionB:
 Identifier: aws/build@v1
 Configuration:
 Steps:
 - Run: echo ${MyGitHubAction.random-color-generator_SELECTEDCOLOR}
```

#### Exemplo com contexto

O exemplo a seguir mostra como definir uma variável SELECTEDCOLOR em GitHubActionA, exibi-la e, depois, referenciá-la em BuildActionB.

```
Actions:
 GitHubActionA:
 Identifier: aws/github-actions-runner@v1
 Configuration:
 Steps:
 - name: Set selected color
 run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
 id: random-color-generator
 Outputs:
 Variables:
 - 'random-color-generator_SELECTEDCOLOR'

 BuildActionB:
 Identifier: aws/build@v1
 Configuration:
 Steps:
```

```
- Run: echo ${GitHubActionA.random-color-generator_SELECTEDCOLOR}
```

## GitHub Ação 'Ações' YAML

A seguir está a definição YAML da ação GitHubAções.

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

Escolha uma propriedade YAML no código a seguir para ver uma descrição dela.

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
action-name:
 Identifier: aws/github-actions-runner@v1
 DependsOn:
 - dependent-action-name-1
 Compute:
 Fleet: fleet-name
 Timeout: timeout-minutes
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
 Inputs:
 Sources:
```

- *source-name-1*
- *source-name-2*

#### Artifacts:

- *artifact-name*

#### Variables:

- Name: *variable-name-1*  
Value: *variable-value-1*
- Name: *variable-name-2*  
Value: *variable-value-2*

#### Outputs:

##### Artifacts:

- Name: *output-artifact-1*

##### Files:

- github-output/artifact-1.jar
- "github-output/build\*"

- Name: *output-artifact-2*

##### Files:

- github-output/artifact-2.1.jar
- github-output/artifact-2.2.jar

#### Variables:

- *variable-name-1*
- *variable-name-2*

#### AutoDiscoverReports:

Enabled: *true | false*

ReportNamePrefix: *AutoDiscovered*

##### IncludePaths:

- *\*\*/\**

##### ExcludePaths:

- *node\_modules/cdk/junit.xml*

#### SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

##### Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

#### Reports:

##### report-name-1:

Format: *format*

##### IncludePaths:

- *\*.xml*

##### ExcludePaths:

- *report2.xml*
- *report3.xml*



SuccessCriteria:PassRate: *percent*LineCoverage: *percent*BranchCoverage: *percent*Vulnerabilities:Severity: *CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL*Number: *whole-number*ConfigurationSteps:

- *github-actions-code*

action-name

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

UI correspondente: guia Configuração/ *action-name*

Identifier

*(action-name/Identifier)*

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Use aws/github-actions-runner@v1 para GitHubações de ações.

UI correspondente: diagrama de fluxo de trabalho//*action-name*aws/ @v1 label github-actions-runner

DependsOn

*(action-name/DependsOn)*

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

Compute

(*action-name*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Fleet

(*action-name*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

Interface de usuário correspondente: guia Configuração/Frota de computação - opcional

Timeout

(*action-name*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

Environment

(*action-name*/Environment)

(Optional)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

 Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs e Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

Name

(*action-name*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

## Connections

(*action-name*/Environment/Connections)

(Optional)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em `Environment`.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

UI correspondente: a configuração tab/Environment/What está pronta *my-environment*? /menu de três pontos/ Mudar função

Name

(*action-name*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

UI correspondente: a configuração tab/Environment/What está pronta *my-environment*? /menu de três pontos/ Mudar função

Role

(*action-name*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome do perfil do IAM que essa ação usa para acessar e operar em serviços da AWS , como o Amazon S3 e o Amazon ECR. Esse perfil deve ser adicionado à conexão da Conta da AWS no espaço. Para adicionar um perfil do IAM a uma conexão de conta, consulte [Adicionar perfis do IAM às conexões da conta](#).

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

 Note

É possível usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

 Warning

Limite as permissões às exigidas pela GitHub ação Ação. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

UI correspondente: a configuração `tab/Environment/What está prontamy-environment?` /menu de três pontos/ Mudar função

Inputs

(*action-name*/Inputs)

(Optional)

A seção `Inputs` define os dados de que uma ação precisa durante a execução de um fluxo de trabalho.

 Note

São permitidas no máximo quatro entradas (uma fonte e três artefatos) por GitHub ação de Ações. As variáveis não são contadas nesse total.

Se você precisar consultar arquivos que residem em entradas diferentes (digamos, uma origem e um artefato), a entrada de origem será a entrada primária e o artefato será a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferirem das primárias. Para obter detalhes, consulte [Exemplo: referência de arquivos em vários artefatos](#).

Interface de usuário correspondente: guia Entradas

Sources

(*action-name*/Inputs/Sources)

(Optional)

Especifique os rótulos que representam os repositórios de origem que serão necessários para a ação. Atualmente, o único rótulo compatível é `WorkflowSource`, que representa o repositório de origem em que o arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma origem, deverá especificar pelo menos um artefato de entrada em *action-name*/Inputs/Artifacts.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

Artifacts - input

(*action-name*/Inputs/Artifacts)

(Optional)

Especifique artefatos de ações anteriores que você deseja fornecer como entrada para essa ação. Esses artefatos já devem estar definidos como artefatos de saída em ações anteriores.

Se você não especificar nenhum artefato de entrada, deverá especificar pelo menos um repositório de origem em *action-name*/Inputs/Sources.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

 Note

Se a lista suspensa Artefatos – opcional não estiver disponível (editor visual) ou se você receber erros ao validar o YAML (editor YAML), talvez seja porque a ação permite apenas uma entrada. Nesse caso, tente remover a entrada de origem.

Interface de usuário correspondente: guia Entradas/Artefatos - opcional

Variables - input

(*action-name*/Inputs/Variables)

(Optional)

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Variáveis - opcional

Outputs

(*action-name*/Outputs)

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

Artifacts - output

(*action-name*/Outputs/Artifacts)

## (Optional)

Especifique o nome de um artefato gerado pela ação. Os nomes de artefato devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (\_). Espaços, hifens (-) e outros caracteres especiais não são permitidos. Não é possível usar aspas para habilitar espaços, hifens e outros caracteres especiais em nomes de artefato de saída.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Saídas/Artefatos

### Name

(*action-name*/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique o nome de um artefato gerado pela ação. Os nomes de artefato devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (\_). Espaços, hifens (-) e outros caracteres especiais não são permitidos. Não é possível usar aspas para habilitar espaços, hifens e outros caracteres especiais em nomes de artefato de saída.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

UI correspondente: artefato de tab/Artifacts/Add saída/Nome do artefato de construção

### Files

(*action-name*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) for incluído)

Especifique os arquivos CodeCatalyst incluídos no artefato que é gerado pela ação. Esses arquivos são gerados pela ação do fluxo de trabalho quando ela é executada e também estão disponíveis no repositório de origem. Os caminhos de arquivo podem residir em um repositório de origem ou em um artefato de uma ação anterior e são relativos ao repositório de origem ou à raiz do artefato. É possível usar padrões glob para especificar caminhos. Exemplos:

- Para especificar um único arquivo que esteja na raiz do local de compilação ou do local do repositório de origem, use `my-file.jar`.



- Para especificar um único arquivo em um subdiretório, use `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Para especificar todos os arquivos, use `"**/*"`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.
- Para especificar todos os arquivos e diretórios em um diretório chamado `directory`, use `"directory/**/*"`. O padrão glob `**` indica que corresponde a qualquer número de subdiretórios.
- Para especificar todos os arquivos em um diretório chamado `directory`, mas não em nenhum de seus subdiretórios, use `"directory/*"`.

 Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outro caractere especial, coloque o caminho entre aspas duplas (""). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

 Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou origem encontrá-lo. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

UI correspondente: produz `tab/Artifacts/Add artefatos/arquivos` produzidos pela compilação

Variables - output

(*action-name*/Outputs/Variables)

(Optional)

Especifique as variáveis que você deseja que a ação exporte para que estejam disponíveis para uso em ações subsequentes.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Saídas/Variáveis/Adicionar variável

variable-name-1

(nome da *action-name* /Outputs/Variables variável-1)

(Optional)

Especifique o nome de uma variável a ser exportada pela ação. Essa variável já deve estar definida na seção Inputs ou Steps da mesma ação.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

UI correspondente: tab/Variables/Add Variável de saída/nome

AutoDiscoverReports

(*action-name*/Outputs/AutoDiscoverReports)

(Optional)

Define a configuração do recurso de descoberta automática.

Quando você ativa a descoberta automática, CodeCatalyst pesquisa todos os dados Inputs passados para a ação, bem como todos os arquivos gerados pela própria ação, procurando relatórios de teste, cobertura de código e análise de composição de software (SCA). Para cada relatório encontrado, ele é CodeCatalyst transformado em um CodeCatalyst relatório. Um CodeCatalyst relatório é um relatório totalmente integrado ao CodeCatalyst serviço e que pode ser visualizado e manipulado por meio do CodeCatalyst console.

 Note

Por padrão, o recurso de descoberta automática inspeciona todos os arquivos. É possível limitar quais arquivos são inspecionados usando as propriedades [IncludePaths](#) ou [ExcludePaths](#).

Interface de usuário correspondente: nenhuma

Enabled

(*action-name*/Outputs/AutoDiscoverReports/Enabled)

(Optional)

Habilite ou desabilite o recurso de descoberta automática.

Os valores válidos são `true` ou `false`.

Se Enabled for omitido, o padrão será `true`.

Interface de usuário correspondente: guia Saídas/Relatórios/Relatórios de descoberta automática

ReportNamePrefix

(*action-name*/Outputs/AutoDiscoverReports/ReportNamePrefix)

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e habilitado)

Especifique um prefixo que seja CodeCatalyst anexado a todos os relatórios encontrados para nomear seus relatórios associados. CodeCatalyst Por exemplo, se você especificar um prefixo deAutoDiscovered, e CodeCatalyst descobrir automaticamente dois relatórios de teste, TestSuiteOne.xml eTestSuiteTwo.xml, os CodeCatalyst relatórios associados serão nomeados e. AutoDiscoveredTestSuiteOne AutoDiscoveredTestSuiteTwo

UI correspondente: saídas tab/Reports/Automatically descubrem relatórios/prefixo do relatório

IncludePaths

(*action-name*/Outputs/AutoDiscoverReports/IncludePaths)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/IncludePaths)

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e habilitado ou se [Reports](#) estiver incluído)

Especifique os arquivos e os caminhos de arquivo CodeCatalyst incluídos na pesquisa de relatórios brutos. Por exemplo, se você especificar"/test/report/\*", CodeCatalyst pesquisa toda a

[imagem de compilação](#) usada pela ação procurando o `/test/report/*` diretório. Quando encontra esse diretório CodeCatalyst, procura relatórios nesse diretório.

 Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outros caracteres especiais, coloque o caminho entre aspas duplas ("). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Se essa propriedade for omitida, o padrão será `**/*`, o que significa que a pesquisa inclui todos os arquivos em todos os caminhos.

 Note

Em relatórios configurados manualmente, `IncludePaths` deverá ser um padrão global que corresponda a um único arquivo.

Interface de usuário correspondente:

- Caminhos de saída' tab/Reports/Automatically discover reports/'Include/exclude Caminhos de inclusão'
- As saídas tab/Reports/Manually configuram relatórios/ 'incluir/excluir caminhos'**report-name-1/** Incluir caminhos

`ExcludePaths`

(**action-name**/Outputs/AutoDiscoverReports/ExcludePaths)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/ExcludePaths)

(Optional)

Especifique os arquivos e os caminhos de arquivo que são CodeCatalyst excluídos ao pesquisar relatórios brutos. Por exemplo, se você especificar `/test/my-reports/**/*`, não CodeCatalyst procurará arquivos no `/test/my-reports/` diretório. Para ignorar todos os arquivos em um diretório, use o padrão de glob `**/*`.

 Note

Se o caminho do arquivo incluir um ou mais asteriscos (\*) ou outros caracteres especiais, coloque o caminho entre aspas duplas (" "). Para ter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Interface de usuário correspondente:

- Caminhos de tab/Reports/Automatically discover reports/'Include/exclude saída'/ Excluir caminhos
- As saídas tab/Reports/Manually configuram relatórios/ /'Include/Excluir caminhos'**report-name-1**/Excluir caminhos

SuccessCriteria

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria)

(Optional)

Especifique os critérios de sucesso para os relatórios de teste, cobertura de código, análise de composição de software (SCA) e análise estática (SA).

Para obter mais informações, consulte [Configuração de critérios de sucesso para relatórios](#).

Interface de usuário correspondente:

- Resultados: tab/Reports/Automatically descubra relatórios/critérios de sucesso
- Saídas tab/Reports/Manually configuram relatórios//Critérios de sucesso **report-name-1**

PassRate

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/PassRate)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria/PassRate)

## (Optional)

Especifique a porcentagem de testes em um relatório de teste que devem ser aprovados para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para ter mais informações sobre os relatórios de teste, consulte [Relatórios de teste](#).

Interface de usuário correspondente:

- tab/Reports/Automatically discover reports/SuccessCritérios de saídas/Taxa de aprovação
- Saídas tab/Reports/Manually configuram relatórios/ /Critérios de **report-name-1** sucesso/ Taxa de aprovação

## LineCoverage

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/LineCoverage)

Ou

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria/LineCoverage)

## (Optional)

Especifique a porcentagem de linhas em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

Interface de usuário correspondente:

- tab/Reports/Automatically discover reports/SuccessCritérios de saídas/Cobertura de linha
- As saídas tab/Reports/Manually configuram relatórios/ /Critérios de **report-name-1** sucesso/ Cobertura de linha

## BranchCoverage

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/BranchCoverage)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/BranchCoverage)

(Optional)

Especifique a porcentagem de filiais em um relatório de cobertura de código que devem ser cobertas para que o CodeCatalyst relatório associado seja marcado como aprovado. Os valores válidos são números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de ramificações são aplicados somente aos relatórios de cobertura de código. Para ter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

Interface de usuário correspondente:

- tab/Reports/Automatically discover reports/SuccessCritérios de saídas/Cobertura da filial
- As saídas tab/Reports/Manually configuram relatórios/ /Critérios de *report-name-1* sucesso/Cobertura de filiais

Vulnerabilities

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities)

Ou

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/Vulnerabilities)

(Optional)

Especifique o número máximo e a gravidade das vulnerabilidades permitidas no relatório SCA para que o CodeCatalyst relatório associado seja marcado como aprovado. Para especificar vulnerabilidades, é necessário especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais grave para o menos grave, são: CRITICAL, HIGH, MEDIUM, LOW e INFORMATIONAL.

Por exemplo, se você escolher HIGH, as vulnerabilidades HIGH e CRITICAL serão contabilizadas.

- O número máximo de vulnerabilidades da gravidade especificada que você deseja permitir. Exceder esse número faz com que o CodeCatalyst relatório seja marcado como falhado. Os valores válidos são números inteiros.

Os critérios de vulnerabilidade são aplicados somente aos relatórios de SCA. Para ter mais informações sobre os relatórios de SCA, consulte [Relatórios de análise de composição de software](#).

Para especificar a gravidade mínima, use a propriedade `Severity`. Para especificar o número máximo de vulnerabilidades, use a propriedade `Number`.

Para ter mais informações sobre os relatórios de SCA, consulte [Tipos de relatório de qualidade](#).

Interface de usuário correspondente:

- `tab/Reports/Automatically discover reports/Success` Critérios de saídas/vulnerabilidades
- As saídas `tab/Reports/Manually configuram relatórios/ /Critérios de sucesso/ report-name-1` Vulnerabilidades

## Reports

(`action-name/Outputs/Reports` )

(Optional)

Uma seção que especifica a configuração dos relatórios de teste.

Interface de usuário correspondente: guia Saídas/Relatórios

`report-name-1`

(nome do `action-name` /Outputs/Reports/ relatório-1)

(Obrigatório se [Reports](#) for incluído)

O nome que você deseja dar ao CodeCatalyst relatório que será gerado a partir de seus relatórios brutos.

UI correspondente: saídas `tab/Reports/Manually configuram relatórios/nome do relatório`

## Format

(`action-name/Outputs/Reports/report-name-1/Format`)

(Obrigatório se [Reports](#) for incluído)

Especifique o formato de arquivo utilizado para os relatórios. Veja a seguir os valores possíveis.

- Para relatórios de teste:
  - Para Cucumber JSON, especifique Cucumber (editor visual) ou CUCUMBERJSON (editor YAML).
  - Para JUnit XML, especifique JUnit(editor visual) ou JUNITXML (editor YAML).



- Para NUnit XML, especifique NUnit(editor visual) ou NUNITXML (editor YAML).
- Para NUnit 3 XML, especifique NUnit3(editor visual) ou NUNIT3XML (editor YAML).
- Para Visual Studio TRX, especifique Visual Studio TRX (editor visual) ou VISUALSTUDIOTRX (editor YAML).
- Para TestNG XML, especifique TestNG (editor visual) ou TESTNGXML (editor YAML).
- Para relatórios de cobertura de código:
  - Para Clover XML, especifique Clover (editor visual) ou CLOVERXML (editor YAML).
  - Para Cobertura XML, especifique Cobertura (editor visual) ou COBERTURAXML (editor YAML).
  - Para JaCoCo XML, especifique JaCoCo(editor visual) ou JACOCOXML (editor YAML).
  - Para SimpleCov JSON gerado por [simplecov](#), [não simplecov-json](#), especifique Simplecov (editor visual) ou (editor YAML). SIMPLECOV
- Para relatórios de análise de composição de software (SCA):
  - Para SARIF, especifique SARIF (editor visual) ou SARIFSCA (editor YAML).

UI correspondente: tab/Reports/Manually configure reports/Add relatório de saídas **report-name-1**//Tipo de relatório e formato do relatório

## Configuration

(**action-name**/Configuration)

(Obrigatório) Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## Steps

(**action-name**/Configuration/Steps)

(Obrigatório)

Especifique seu código de GitHub ação conforme ele aparece na página de detalhes da ação no [GitHub Marketplace](#). Adicione o código seguindo estas diretrizes:

1. Cole o código da steps : seção GitHub Ação na Steps : seção do CodeCatalyst fluxo de trabalho. O código começa com um traço (-) e tem uma aparência semelhante ao exemplo a seguir.

GitHub código para colar:

```
- name: Lint Code Base
 uses: github/super-linter@v4
 env:
 VALIDATE_ALL_CODEBASE: false
 DEFAULT_BRANCH: master
 GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

2. Revise o código que você acabou de colar e modifique-o conforme necessário para que esteja em conformidade com os CodeCatalyst padrões. Por exemplo, com o bloco de código anterior, você pode remover o código e adicioná-lo em negrito. *red italics*

CodeCatalyst fluxo de trabalho yaml:

```
Steps:
 - name: Lint Code Base
 uses: github/super-linter@v4
 env:
 VALIDATE_ALL_CODEBASE: false
 DEFAULT_BRANCH: mastermain
 GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

3. Para obter código adicional incluído na GitHub Ação, mas que não existe dentro da steps : seção, adicione-o ao CodeCatalyst fluxo de trabalho usando o código CodeCatalyst -equivalent. Você pode revisar o [Definição do YAML do fluxo de trabalho](#) para ter uma ideia de como você pode portar seu GitHub código para CodeCatalyst o. As etapas detalhadas da migração estão fora do escopo deste guia.

Aqui está um exemplo de como especificar caminhos de arquivo em uma ação de GitHub ações:

```
Steps:
 - name: Lint Code Base
 uses: github/super-linter@v4
 ...
 - run: cd /sources/WorkflowSource/MyFolder/ && cat file.txt
 - run: cd /artifacts/MyGitHubAction/MyArtifact/MyFolder/ && cat file2.txt
```

Para ter mais informações sobre como especificar caminhos de arquivo, consulte [Fazer referência a arquivos do repositório de origem](#) e [Referência de arquivos em um artefato](#).

UI correspondente: guia de GitHub configuração/ações YAML

## Configuração de imagens de computação e runtime

Em um CodeCatalyst fluxo de trabalho, você pode especificar a imagem do ambiente de computação e tempo de execução CodeCatalyst usada para executar ações do fluxo de trabalho.

Computação se refere ao mecanismo de computação (CPU, memória e sistema operacional) gerenciado e mantido por CodeCatalyst para executar ações de fluxo de trabalho.

### Note

Se a computação for definida como uma propriedade do fluxo de trabalho, ela não poderá ser definida como uma propriedade de nenhuma ação nesse fluxo de trabalho. Da mesma forma, se computação for definida como uma propriedade de qualquer ação, ela não poderá ser definida no fluxo de trabalho.

Uma imagem do ambiente de tempo de execução é um contêiner do Docker no qual são CodeCatalyst executadas ações de fluxo de trabalho. O contêiner do Docker é executado na plataforma de computação escolhida e inclui um sistema operacional e ferramentas extras que uma ação de fluxo de trabalho pode precisar, como o AWS CLI Node.js e .tar.

### Tópicos

- [Tipos de computação](#)
- [Frotas de computação](#)
- [Propriedades da frota sob demanda](#)
- [Propriedades da frota provisionada](#)
- [Criar uma frota provisionada](#)
- [Edição de uma frota provisionada](#)
- [Exclusão de uma frota provisionada](#)
- [Atribuir uma frota ou computação a uma ação](#)
- [Compartilhamento de computação entre ações](#)
- [Especificação de imagens de ambiente de runtime](#)

## Tipos de computação

CodeCatalyst oferece os seguintes tipos de computação:

- Amazon EC2
- AWS Lambda

A Amazon EC2 oferece flexibilidade otimizada durante a execução de ações e o Lambda oferece velocidades otimizadas de inicialização de ações. O Lambda é compatível com execuções de ações de fluxo de trabalho mais rápidas devido a uma menor latência de inicialização. O Lambda permite executar fluxos de trabalho básicos que podem criar, testar e implantar aplicações com tecnologia sem servidor com runtimes comuns. Esses runtimes incluem Node.js, Python, Java, .NET e Go. No entanto, existem alguns casos de uso que o Lambda não suporta e, se afetarem você, use o tipo de computação Amazon: EC2

- O Lambda não é compatível com imagens de ambiente de runtime de um registro especificado.
- O Lambda não é compatível com ferramentas que exijam permissões de raiz. Para ferramentas como yum ou rpm, use o tipo de EC2 computação Amazon ou outras ferramentas que não exijam permissões de root.
- O Lambda não é compatível com compilações ou execuções do Docker. As seguintes ações que usam imagens do Docker não são suportadas: Deploy AWS CloudFormation stack, Deploy to Amazon ECS, Amazon S3 publish, AWS CDK bootstrap, AWS CDK deploy, invoke e Actions. AWS Lambda GitHub Actions As ações baseadas em Docker que estão sendo executadas na ação CodeCatalyst GitHub Actions também não são compatíveis com a computação Lambda. É possível usar alternativas que não exijam permissões de raiz, como o Podman.
- O Lambda não é compatível com a gravação em arquivos externos a /tmp. Ao configurar suas ações de fluxo de trabalho, você pode reconfigurar suas ferramentas para instalação ou gravação de /tmp. Se você tiver uma ação de criação que instale npm, configure-a para instalar em /tmp.
- O Lambda não é compatível com runtimes de mais de 15 minutos.

## Frotas de computação

CodeCatalyst oferece as seguintes frotas de computação:

- Frotas sob demanda
- Frotas provisionadas

Com frotas sob demanda, quando uma ação de fluxo de trabalho é iniciada, o fluxo de trabalho provisiona os recursos necessários. As máquinas são destruídas quando a ação termina. Você só paga pelo número de minutos em que executar suas ações. As frotas sob demanda são totalmente gerenciadas e incluem recursos de escalabilidade automática para lidar com picos de demanda.

CodeCatalyst também oferece frotas provisionadas que contêm máquinas alimentadas pela Amazon EC2 que são mantidas pela. CodeCatalyst Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Com frotas provisionadas, suas máquinas estão sempre funcionando e incorrerão em custos enquanto forem provisionadas.

Para criar, atualizar ou excluir uma frota, você deve ter o perfil de administrador do espaço ou o perfil de administrador do projeto.

## Propriedades da frota sob demanda

CodeCatalyst fornece as seguintes frotas sob demanda:

| Nome                | Sistema operacional | Arquitetura | v CPUs | Memória (GiB) | Espaço em disco | Tipos de computação compatíveis |
|---------------------|---------------------|-------------|--------|---------------|-----------------|---------------------------------|
| Linux.Arm64.Large   | Amazon Linux 2      | Arm64       | 2      | 4             | 64 GB           | Amazon EC2                      |
|                     |                     |             |        |               | 10 GB           | Lambda                          |
| Linux.Arm64.XLarge  | Amazon Linux 2      | Arm64       | 4      | 8             | 128 GB          | Amazon EC2                      |
|                     |                     |             |        |               | 10 GB           | Lambda                          |
| Linux.Arm64.2XLarge | Amazon Linux 2      | Arm64       | 8      | 16            | 128 GB          | Amazon EC2                      |

| Nome                 | Sistema operacional | Arquitetura | v CPUs | Memória (GiB) | Espaço em disco | Tipos de computação compatíveis |
|----------------------|---------------------|-------------|--------|---------------|-----------------|---------------------------------|
| Linux.x86-64.Large   | Amazon Linux 2      | x86-64      | 2      | 4             | 64 GB           | Amazon EC2                      |
|                      |                     |             |        |               | 10 GB           | Lambda                          |
| Linux.x86-64.XLarge  | Amazon Linux 2      | x86-64      | 4      | 8             | 128 GB          | Amazon EC2                      |
|                      |                     |             |        |               | 10 GB           | Lambda                          |
| Linux.x86-64.2XLarge | Amazon Linux 2      | x86-64      | 8      | 16            | 128 GB          | Amazon EC2                      |

#### Note

As especificações para frotas sob demanda variam de acordo com seu nível de faturamento. Para obter mais informações, consulte [Preço](#).

Se nenhuma frota for selecionada, CodeCatalyst usa Linux.x86-64.Large.

## Propriedades da frota provisionada

Uma frota provisionada contém as seguintes propriedades:

### Sistema operacional

O sistema operacional Os seguintes sistemas operacionais estão disponíveis:

- Amazon Linux 2
- Windows Server 2022

**Note**

As frotas do Windows são compatíveis somente na ação de criação. Outras ações atualmente não são compatíveis com o Windows.

## Arquitetura

A arquitetura do processador. As seguintes arquiteturas estão disponíveis:

- x86\_64
- Arm64

## Tipo de máquina

O tipo de máquina para cada instância. Os seguintes tipos de máquina estão disponíveis:

| v CPUs | Memória (GiB) | Espaço em disco | Sistema operacional |
|--------|---------------|-----------------|---------------------|
| 2      | 4             | 64 GB           | Amazon Linux 2      |
| 4      | 8             | 128 GB          | Amazon Linux 2      |
|        |               |                 | Windows Server 2022 |
| 8      | 16            | 128 GB          | Amazon Linux 2      |
|        |               |                 | Windows Server 2022 |

## Capacidade

O número inicial de máquinas alocadas para a frota, que define o número de ações que podem ser executadas paralelamente.

## Modo de escalabilidade

Define o comportamento quando o número de ações excede a capacidade da frota.

## Provisionamento da capacidade adicional sob demanda

Máquinas adicionais são configuradas sob demanda, o que aumenta a escala verticalmente de forma automática em resposta à execução de novas ações e, depois, reduz a escala verticalmente para a capacidade básica ao finalizar as ações. Isso pode gerar custos adicionais, já que você paga por minuto por cada máquina em funcionamento.

Esperar até que a capacidade adicional da frota esteja disponível

As execuções de ação são colocadas em uma fila até que uma máquina esteja disponível. Isso limita os custos adicionais porque nenhuma máquina adicional é alocada.

## Criar uma frota provisionada

Use as instruções a seguir para criar uma frota provisionada.

### Note

As frotas provisionadas serão desativadas após 2 semanas de inatividade. Se usadas novamente, elas serão reativadas automaticamente, mas essa reativação pode causar uma latência.

### Como criar uma frota provisionada

1. No painel de navegação, selecione CI/CD e Computação.
2. Selecione Criar frota provisionada.
3. No campo de texto Nome da frota provisionada, insira um nome para a frota.
4. No menu suspenso Sistema operacional, escolha o sistema operacional.
5. No menu suspenso Tipo de máquina, escolha o tipo de máquina.
6. No campo de texto Capacidade, insira o número máximo de máquinas na frota.
7. No menu suspenso Modo de escalabilidade, escolha o comportamento de sobreposição desejado. Consulte mais informações sobre esses campos em [Propriedades da frota provisionada](#).
8. Escolha Criar.



Depois de criar a frota provisionada, você está pronto para atribuí-la a uma ação. Para obter mais informações, consulte [Atribuir uma frota ou computação a uma ação](#).

## Edição de uma frota provisionada

Use as instruções a seguir para editar uma frota provisionada.

### Note

As frotas provisionadas serão desativadas após 2 semanas de inatividade. Se usadas novamente, elas serão reativadas automaticamente, mas essa reativação pode causar uma latência.

Para editar uma frota provisionada

1. No painel de navegação, selecione CI/CD e Computação.
2. Na lista Frota provisionada, selecione a frota que você quer editar.
3. Escolha Editar.
4. No campo de texto Capacidade, insira o número máximo de máquinas na frota.
5. No menu suspenso Modo de escalabilidade, escolha o comportamento de sobreposição desejado. Consulte mais informações sobre esses campos em [Propriedades da frota provisionada](#).
6. Escolha Salvar.

## Exclusão de uma frota provisionada

Use as instruções a seguir para excluir uma frota provisionada.

Para excluir uma frota provisionada

### Warning

Antes de excluir uma frota provisionada, remova-a de todas as ações excluindo a propriedade `Fleet` do código YAML da ação. Qualquer ação que continue referenciando uma frota provisionada depois que ela for excluída falhará na próxima vez em que a ação for executada.

1. No painel de navegação, selecione CI/CD e Computação.
2. Na lista Frota provisionada, selecione a frota que você quer excluir.
3. Escolha Excluir.
4. Insira **delete** para confirmar a exclusão.
5. Escolha Excluir.

## Atribuir uma frota ou computação a uma ação

Por padrão, as ações de fluxo de trabalho usam a frota `Linux.x86-64.Large` sob demanda com um tipo de EC2 computação da Amazon. Para usar uma frota provisionada ou uma frota sob demanda diferente, como `Linux.x86-64.2XLarge`, siga estas instruções.

### Visual

#### Antes de começar

- Se quiser atribuir uma frota provisionada, primeiro será necessário criá-la. Para obter mais informações, consulte [Criar uma frota provisionada](#).

#### Como atribuir uma frota provisionada ou um tipo de frota diferente a uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, escolha a ação à qual você deseja atribuir a frota provisionada ou o novo tipo de frota.
8. Escolha a guia Configuração.
9. Em Frota de computação, faça o seguinte:

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se `Fleet` for omitido, o padrão será `Linux.x86-64.Large`.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

### Antes de começar

- Se quiser atribuir uma frota provisionada, primeiro será necessário criá-la. Para obter mais informações, consulte [Criar uma frota provisionada](#).

Como atribuir uma frota provisionada ou um tipo de frota diferente a uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Encontre a ação à qual você deseja atribuir a frota provisionada ou o novo tipo de frota.

8. Na ação, adicione uma propriedade Compute e defina Fleet com o nome da frota ou o tipo de frota sob demanda. Para ter mais informações, consulte a descrição da propriedade Fleet em [Ações de criação e de teste YAML](#) para a ação.
9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Compartilhamento de computação entre ações

Por padrão, as ações em um fluxo de trabalho são executadas em instâncias separadas em uma [frota](#). Esse comportamento fornece ações com isolamento e previsibilidade sobre o estado das entradas. O comportamento padrão requer configuração explícita para compartilhar contexto, como arquivos e variáveis, entre as ações.

O compartilhamento de computação é um recurso que permite executar todas as ações em um fluxo de trabalho na mesma instância. Usar o compartilhamento de computação pode proporcionar runtimes de fluxo de trabalho mais rápidos, pois menos tempo é gasto no provisionamento de instâncias. Você também pode compartilhar arquivos (artefatos) entre ações sem configuração adicional do fluxo de trabalho.

Quando um fluxo de trabalho é executado usando o compartilhamento de computação, uma instância na frota padrão ou especificada é reservada durante todas as ações nesse fluxo de trabalho. Quando a execução do fluxo de trabalho é concluída, a reserva da instância é liberada.

### Tópicos

- [Execução de várias ações em computação compartilhada](#)
- [Considerações sobre compartilhamento de computação](#)
- [Ativação do compartilhamento de computação](#)
- [Exemplos](#)

## Execução de várias ações em computação compartilhada

Você pode usar o atributo Compute no YAML de definição no nível do fluxo de trabalho para especificar as propriedades de compartilhamento de frota e computação das ações. Também é possível configurar propriedades de computação usando o editor visual no CodeCatalyst. Para

especificar uma frota, defina o nome de uma frota existente, defina o tipo de computação como EC2 e ative o compartilhamento de computação.

 Note

Só haverá suporte para o compartilhamento de computação se o tipo de computação estiver definido como EC2. Ele não é compatível com o sistema operacional Windows Server 2022. Para ter informações sobre frotas, tipos e propriedades de computação, consulte [Configuração de imagens de computação e runtime](#).

 Note

Se você estiver no nível Gratuito e especificar a frota `Linux.x86-64.XLarge` ou `Linux.x86-64.2XLarge` manualmente no YAML de definição de fluxo de trabalho, a ação ainda será executada na frota padrão (`Linux.x86-64.Large`). Para ter mais informações sobre disponibilidade e preços de computação, consulte a [tabela com as opções de níveis](#).

Quando o compartilhamento de computação está ativado, a pasta que contém a origem do fluxo de trabalho é copiada automaticamente entre as ações. Você não precisa configurar artefatos de saída e referenciá-los como artefatos de entrada em uma definição de fluxo de trabalho (arquivo YAML). Como autor do fluxo de trabalho, você precisa conectar variáveis de ambiente usando entradas e saídas, da mesma forma que faria sem usar o compartilhamento de computação. Se você quiser compartilhar pastas entre ações fora da origem do fluxo de trabalho, considere o armazenamento em cache de arquivos. Para obter mais informações, consulte [Compartilhar artefatos e arquivos entre ações](#) e [Armazenar arquivos em cache entre execuções de fluxo de trabalho](#).

O repositório de origem em que seu arquivo de definição de fluxo de trabalho reside é identificado pelo rótulo `WorkflowSource`. Ao usar o compartilhamento de computação, a origem do fluxo de trabalho é baixada na primeira ação que faz referência a ela e disponibilizada automaticamente para uso em ações subsequentes na execução do fluxo de trabalho. Todas as alterações feitas na pasta que contém a origem do fluxo de trabalho por meio de uma ação, como adicionar, modificar ou remover arquivos, também são visíveis nas ações subsequentes do fluxo de trabalho. Você pode referenciar arquivos que residem na pasta de origem do fluxo de trabalho em qualquer ação do fluxo de trabalho, da mesma forma que poderia fazer sem usar o compartilhamento de computação. Para obter mais informações, consulte [Fazer referência a arquivos do repositório de origem](#).

**Note**

Os fluxos de trabalho de compartilhamento de computação precisam especificar uma sequência estrita de ações, para que ações paralelas não possam ser definidas. Embora os artefatos de saída possam ser configurados em qualquer ação na sequência, não há suporte para os artefatos de entrada.

## Considerações sobre compartilhamento de computação

Você pode executar fluxos de trabalho com compartilhamento de computação para acelerar as execuções de fluxo de trabalho e compartilhar o contexto entre as ações em um fluxo de trabalho que usa a mesma instância. Considere o seguinte para determinar se o uso do compartilhamento de computação é apropriado para seu cenário:

|                                                       | Compartilhamento de computação                                               | Sem compartilhamento de computação                                           |
|-------------------------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Tipo de computação                                    | Amazon EC2                                                                   | Amazon EC2, AWS Lambda                                                       |
| Provisionamento de instância                          | Ações executadas na mesma instância                                          | As ações são executadas em instâncias separadas                              |
| Sistema operacional                                   | Amazon Linux 2                                                               | Amazon Linux 2, Windows Server 2022 (somente ação de criação)                |
| Referência a arquivos                                 | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code> | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code> |
| Estrutura Workflow                                    | As ações só podem ser executadas em sequência                                | As ações podem ser executadas paralelamente                                  |
| Acesso a dados em todas as ações do fluxo de trabalho | Acessar a origem de fluxo de trabalho armazenada em cache (WorkflowSource )  | Acessar saídas de artefatos compartilhados (requer configuração adicional)   |

## Ativação do compartilhamento de computação

Use as instruções a seguir para ativar o compartilhamento de computação para um fluxo de trabalho.

### Visual

Para ativar o compartilhamento de computação usando o editor visual

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. Selecione Propriedades do fluxo de trabalho.
8. No menu suspenso Tipo de computação, selecione EC2.
9. (Opcional) No menu suspenso Frota de computação - opcional, escolha uma frota que você deseja usar para executar ações de fluxo de trabalho. Você pode escolher uma frota sob demanda ou criar e escolher uma frota provisionada. Para ter mais informações, consulte [Criar uma frota provisionada](#) e [Atribuir uma frota ou computação a uma ação](#).
10. Alterne o botão para ativar o compartilhamento de computação e fazer com que as ações no fluxo de trabalho sejam executadas na mesma frota.
11. (Opcional) Escolha o modo de execução do fluxo de trabalho. Para obter mais informações, consulte [Configurar o comportamento de enfileiramento das execuções](#).
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

### YAML

Para ativar o compartilhamento de computação usando o editor YAML

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho.
5. Escolha Editar.

6. Selecione YAML.
7. Ative o compartilhamento de computação definindo o campo `SharedInstance` como `TRUE` e `Type` como `EC2`. Defina `Fleet` como uma frota de computação que você deseja usar para executar ações de fluxo de trabalho. Você pode escolher uma frota sob demanda ou criar e escolher uma frota provisionada. Para ter mais informações, consulte [Criar uma frota provisionada](#) e [Atribuir uma frota ou computação a uma ação](#)

Em um YAML de fluxo de trabalho, adicione um código semelhante ao seguinte:

```
Name: MyWorkflow
SchemaVersion: "1.0"
Compute: # Define compute configuration.
 Type: EC2
 Fleet: MyFleet # Optionally, choose an on-demand or provisioned fleet.
 SharedInstance: true # Turn on compute sharing. Default is False.
Actions:
 BuildFirst:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: ...
 ...
```

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Exemplos

### Tópicos

- [Exemplo: Amazon S3 Publish](#)



## Exemplo: Amazon S3 Publish

Os exemplos de fluxo de trabalho a seguir mostram como realizar a ação do Amazon S3 Publish de duas maneiras: primeiro usando artefatos de entrada e, depois, usando o compartilhamento de computação. Com o compartilhamento de computação, os artefatos de entrada não são necessários, pois você pode acessar a `WorkflowSource` armazenada em cache. Além disso, o artefato de saída na ação de criação não é mais necessário. A ação do S3 Publish está configurada para usar a propriedade `DependsOn` explícita para manter ações sequenciais; a ação de criação deve ser executada para que a ação do S3 Publish seja executada.

- Sem o compartilhamento de computação, você precisa usar artefatos de entrada e compartilhar as saídas com ações subsequentes:

```
Name: S3PublishUsingInputArtifact
SchemaVersion: "1.0"
Actions:
 Build:
 Identifier: aws/build@v1
 Outputs:
 Artifacts:
 - Name: ArtifactToPublish
 Files: [output.zip]
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: ./build.sh # Build script that generates output.zip
 PublishToS3:
 Identifier: aws/s3-publish@v1
 Inputs:
 Artifacts:
 - ArtifactToPublish
 Environment:
 Connections:
 - Role: codecatalyst-deployment-role
 Name: dev-deployment-role
 Name: dev-connection
 Configuration:
 SourcePath: output.zip
 DestinationBucketName: amzn-s3-demo-bucket
```

- Ao usar o compartilhamento de computação definindo `SharedInstance` como `TRUE`, você pode executar várias ações na mesma instância e compartilhar artefatos especificando uma única origem de fluxo de trabalho. Os artefatos de entrada não são obrigatórios e não podem ser especificados:

```
Name: S3PublishUsingComputeSharing
SchemaVersion: "1.0"
Compute:
 Type: EC2
 Fleet: dev-fleet
 SharedInstance: TRUE
Actions:
 Build:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: ./build.sh # Build script that generates output.zip
 PublishToS3:
 Identifier: aws/s3-publish@v1
 DependsOn:
 - Build
 Environment:
 Connections:
 - Role: codecatalyst-deployment-role
 Name: dev-deployment-role
 Name: dev-connection
 Configuration:
 SourcePath: output.zip
 DestinationBucketName: amzn-s3-demo-bucket
```

## Especificação de imagens de ambiente de runtime

Uma imagem do ambiente de tempo de execução é um contêiner do Docker no qual são CodeCatalyst executadas ações de fluxo de trabalho. O contêiner do Docker é executado na

plataforma de computação escolhida e inclui um sistema operacional e ferramentas extras que uma ação de fluxo de trabalho pode precisar, como o AWS CLI Node.js e .tar.

Por padrão, as ações do fluxo de trabalho serão executadas em uma das [imagens ativas](#) fornecidas e mantidas pela CodeCatalyst. Somente ações de criação e teste oferecem suporte a imagens personalizadas. Para obter mais informações, consulte [Atribuir uma imagem do Docker de um ambiente de runtime personalizada a uma ação](#).

## Tópicos

- [Imagens ativas](#)
- [E se uma imagem ativa não incluir as ferramentas de que preciso?](#)
- [Atribuir uma imagem do Docker de um ambiente de runtime personalizada a uma ação](#)
- [Exemplos](#)

## Imagens ativas

Imagens ativas são imagens do ambiente de execução que são totalmente suportadas CodeCatalyst e incluem ferramentas pré-instaladas. Atualmente, existem dois conjuntos de imagens ativas: um lançado em março de 2024 e outro lançado em novembro de 2022.

O uso da imagem de março de 2024 ou novembro de 2022 depende da ação:

- As ações de criação e teste adicionadas a um fluxo de trabalho em ou após 26 de março de 2024 incluirão uma seção `Container` na definição de YAML que especifica explicitamente uma [imagem de março de 2024](#). Se desejar, você pode remover a seção `Container` para voltar à [imagem de novembro de 2022](#).
- As ações de criação e teste que foram adicionadas a um fluxo de trabalho antes de 26 de março de 2024 não incluirão uma seção `Container` na definição de YAML e, consequentemente, usarão uma [imagem de novembro de 2022](#). É possível manter a imagem de novembro de 2022 ou atualizá-la. Para atualizar a imagem, abra a ação no editor visual, selecione a guia Configuração e selecione a imagem de março de 2024 na lista suspensa Imagem do Docker do ambiente de runtime. Essa seleção adicionará uma seção `Container` à definição de YAML da ação que é preenchida com a imagem apropriada de março de 2024.
- Todas as outras ações usarão uma [imagem de novembro de 2022](#) ou uma [imagem de março de 2024](#). Para ter mais informações, consulte a documentação da ação.

Tópicos

- [Imagens de março de 2024](#)
- [Imagens de novembro de 2022](#)

Imagens de março de 2024

As imagens de março de 2024 são as imagens mais recentes fornecidas por CodeCatalyst. Há uma imagem de março de 2024 por combinação computacional. type/fleet

A tabela a seguir mostra as ferramentas instaladas em cada imagem de março de 2024.

Ferramentas de imagem de março de 2024

| Ferramenta         | CodeCatalyst<br>Amazon EC2 para<br>Linux x86_64<br>- CodeCatalystLinux_<br>x86_64:2024_03 | CodeCatalyst<br>Lambda para<br>Linux x86_64<br>- CodeCatalystLinuxL<br>ambda_x86_64:2024_03 | CodeCatalyst<br>Amazon EC2<br>para Linux Arm64<br>- CodeCatalystLinux_<br>Arm64:2024_03 | CodeCatalyst<br>Lambda para<br>Linux Arm64 - CodeCatalystLinux_<br>ambda_Arm64:2024_03 |
|--------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| AWS CLI            | 2.15.17                                                                                   | 2.15.17                                                                                     | 2.15.17                                                                                 | 2.15.17                                                                                |
| AWS CLI do Copilot | 1.32.1                                                                                    | 1.32.1                                                                                      | 1.32.1                                                                                  | 1.32.1                                                                                 |
| Docker             | 24.0.9                                                                                    | N/D                                                                                         | 24.0.9                                                                                  | N/D                                                                                    |
| Docker Compose     | 2.23.3                                                                                    | N/D                                                                                         | 2.23.3                                                                                  | N/D                                                                                    |
| Git                | 2.43.0                                                                                    | 2.43.0                                                                                      | 2.43.0                                                                                  | 2.43.0                                                                                 |
| Go                 | 1.21.5                                                                                    | 1.21.5                                                                                      | 1.21.5                                                                                  | 1.21.5                                                                                 |
| Gradle             | 8.5                                                                                       | 8.5                                                                                         | 8.5                                                                                     | 8.5                                                                                    |
| Java               | Corretto17                                                                                | Corretto17                                                                                  | Corretto17                                                                              | Corretto17                                                                             |
| Maven              | 3.9.6                                                                                     | 3.9.6                                                                                       | 3.9.6                                                                                   | 3.9.6                                                                                  |
| Node.js            | 18.19.0                                                                                   | 18.19.0                                                                                     | 18.19.0                                                                                 | 18.19.0                                                                                |

| Ferramenta | CodeCatalyst<br>Amazon EC2 para<br>Linux x86_64<br>- <b>CodeCatalystLinux_x86_64:2024_03</b> | CodeCatalyst<br>Lambda para<br>Linux x86_64<br>- <b>CodeCatalystLinuxLambda_x86_64:2024_03</b> | CodeCatalyst<br>Amazon EC2<br>para Linux Arm64<br>- <b>CodeCatalystLinux_Arm64:2024_03</b> | CodeCatalyst<br>Lambda para<br>Linux Arm64 - <b>CodeCatalystLinuxLambda_Arm64:2024_03</b> |
|------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| npm        | 10.2.3                                                                                       | 10.2.3                                                                                         | 10.2.3                                                                                     | 10.2.3                                                                                    |
| Python     | 3.9.18                                                                                       | 3.9.18                                                                                         | 3.9.18                                                                                     | 3.9.18                                                                                    |
| Python3    | 3.11.6                                                                                       | 3.11.6                                                                                         | 3.11.6                                                                                     | 3.11.6                                                                                    |
| pip        | 22.3.1                                                                                       | 22.3.1                                                                                         | 22.3.1                                                                                     | 22.3.1                                                                                    |
| .NET       | 8.0.100                                                                                      | 8.0.100                                                                                        | 8.0.100                                                                                    | 8.0.100                                                                                   |

## Imagens de novembro de 2022

Há uma imagem de novembro de 2022 por type/fleet combinação computacional. Também há uma imagem do Windows de novembro de 2022 disponível com a ação de criação, caso você tenha configurado uma [frota provisionada](#).

A tabela a seguir mostra as ferramentas instaladas em cada imagem de novembro de 2022.

## Ferramentas de imagem de novembro de 2022

| Ferramenta         | CodeCatalyst<br>Amazon EC2 para<br>Linux x86_64<br>- <b>CodeCatalystLinux_x86_64:2022_11</b> | CodeCatalyst<br>Lambda para<br>Linux x86_64<br>- <b>CodeCatalystLinuxLambda_x86_64:2022_11</b> | CodeCatalyst<br>Amazon EC2<br>para Linux Arm64<br>- <b>CodeCatalystLinux_Arm64:2022_11</b> | CodeCatalyst<br>Lambda para<br>Linux Arm64 - <b>CodeCatalystLinuxLambda_Arm64:2022_11</b> |
|--------------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| AWS CLI            | 2.15.17                                                                                      | 2.15.17                                                                                        | 2.15.17                                                                                    | 2.15.17                                                                                   |
| AWS CLI do Copilot | 0.6.0                                                                                        | 0.6.0                                                                                          | N/D                                                                                        | N/D                                                                                       |

| Ferramenta     | CodeCatalyst<br>Amazon EC2 para<br>Linux x86_64<br>- CodeCatalystLinux_<br>x86_64:2022_11 | CodeCatalyst<br>Lambda para<br>Linux x86_64<br>- CodeCatalystLinuxL<br>ambda_x86<br>_64:2022_11 | CodeCatalyst<br>Amazon EC2<br>para Linux Arm64<br>- CodeCatalystLinux_<br>Arm64:2022_11 | CodeCatalyst<br>Lambda para<br>Linux Arm64 - CodeCatalystLinux_<br>ambda_Ar<br>64:2022_11 |
|----------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Docker         | 23.01                                                                                     | N/D                                                                                             | 23.0.1                                                                                  | N/D                                                                                       |
| Docker Compose | 2.16.0                                                                                    | N/D                                                                                             | 2.16.0                                                                                  | N/D                                                                                       |
| Git            | 2.40.0                                                                                    | 2.40.0                                                                                          | 2.39.2                                                                                  | 2.39.2                                                                                    |
| Go             | 1.20.2                                                                                    | 1.20.2                                                                                          | 1.20.1                                                                                  | 1.20.1                                                                                    |
| Gradle         | 8.0.2                                                                                     | 8.0.2                                                                                           | 8.0.1                                                                                   | 8.0.1                                                                                     |
| Java           | Corretto17                                                                                | Corretto17                                                                                      | Corretto17                                                                              | Corretto17                                                                                |
| Maven          | 3.9.4                                                                                     | 3.9.4                                                                                           | 3.9.0                                                                                   | 3.9.0                                                                                     |
| Node.js        | 16.20.2                                                                                   | 16.20.2                                                                                         | 16.19.1                                                                                 | 16.14.2                                                                                   |
| npm            | 8.19.4                                                                                    | 8.19.4                                                                                          | 8.19.3                                                                                  | 8.5.0                                                                                     |
| Python         | 3.9.15                                                                                    | 2.7.18                                                                                          | 3.11.2                                                                                  | 2.7.18                                                                                    |
| Python3        | N/D                                                                                       | 3.9.15                                                                                          | N/D                                                                                     | 3.11.2                                                                                    |
| pip            | 22.2.2                                                                                    | 22.2.2                                                                                          | 23.0.1                                                                                  | 23.0.1                                                                                    |
| .NET           | 6.0.407                                                                                   | 6.0.407                                                                                         | 6.0.406                                                                                 | 6.0.406                                                                                   |

E se uma imagem ativa não incluir as ferramentas de que preciso?

Se nenhuma das [imagens ativas](#) fornecidas pelo CodeCatalyst incluir as ferramentas necessárias, você tem algumas opções:

- Você pode fornecer uma imagem do Docker de ambiente de runtime personalizada que inclua as ferramentas necessárias. Para obter mais informações, consulte [Atribuir uma imagem do Docker de um ambiente de runtime personalizada a uma ação](#).

 Note

Se você quiser fornecer uma imagem do Docker de ambiente de runtime personalizada, a imagem personalizada deverá ter o Git instalado nela.

- Você pode fazer com que a ação de criação ou teste do fluxo de trabalho instale as ferramentas necessárias.

Por exemplo, você pode incluir as seguintes instruções na seção Steps do código YAML da ação de criação ou teste:

```
Configuration:
 Steps:
 - Run: ./setup-script
```

A *setup-script* instrução então executaria o seguinte script para instalar o gerenciador de pacotes Node (npm):

```
#!/usr/bin/env bash
echo "Setting up environment"

touch ~/.bashrc
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install v16.1.0
source ~/.bashrc
```

Para ter mais informações sobre o YAML da ação de criação, consulte [Ações de criação e de teste YAML](#).

## Atribuir uma imagem do Docker de um ambiente de runtime personalizada a uma ação

Se você não quiser usar uma [imagem ativa](#) fornecida pela CodeCatalyst, você pode fornecer uma imagem Docker de ambiente de tempo de execução personalizada. Se você quiser fornecer uma

imagem personalizada, ela deverá ter o Git instalado. A imagem pode residir no Docker Hub, no Amazon Elastic Container Registry ou em qualquer repositório público.

Para saber como criar uma imagem do Docker personalizada, consulte [Criar contêineres em uma aplicação](#) na documentação do Docker.

Use as instruções a seguir para atribuir a imagem do Docker de ambiente de runtime personalizada a uma ação. Depois de especificar uma imagem, CodeCatalyst implante-a na sua plataforma de computação quando a ação começar.

#### Note

As ações a seguir não oferecem suporte a imagens personalizadas do Docker do ambiente de tempo de execução: Deploy CloudFormation stack, Deploy to ECS e GitHub Actions. As imagens do Docker do ambiente de tempo de execução personalizado também não oferecem suporte ao tipo de computação Lambda.

## Visual

Para atribuir uma imagem do Docker de ambiente de runtime personalizada usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione Visual.
6. No diagrama do fluxo de trabalho, escolha a ação que usará a imagem do Docker de ambiente de runtime personalizada.
7. Escolha a guia Configuração.
8. Na parte inferior, preencha os seguintes campos.

Imagem do Docker de ambiente de runtime - opcional

Especifique o registro em que a imagem está armazenada. Os valores válidos são:



- CODECATALYST (editor YAML)

A imagem é armazenada no CodeCatalyst registro.

- Docker Hub (editor visual) ou DockerHub (editor YAML)

A imagem é armazenada no registro de imagens do Docker Hub.

- Outro registro (editor visual) ou Other (editor YAML)

A imagem é armazenada em um registro de imagem personalizado. Qualquer registro disponível publicamente pode ser usado.

- Amazon Elastic Container Registry (editor visual) ou ECR (editor YAML)

A imagem é armazenada em um repositório de imagens do Amazon Elastic Container Registry. Para usar uma imagem em um repositório do Amazon ECR, essa ação precisa acessar o Amazon ECR. Para habilitar esse acesso, é necessário criar um [perfil do IAM](#) que inclua as permissões e a política de confiança personalizada a seguir. (Se desejar, você poderá modificar um perfil existente para incluir as permissões e a política.)

O perfil do IAM deve incluir as seguintes permissões na política:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

O perfil do IAM deve incluir a seguinte política de confiança personalizada:

Para ter mais informações sobre como criar perfis do IAM, consulte [Criar um perfil usando políticas de confiança personalizadas \(console\)](#) no Guia do usuário do IAM.

Depois que você criar o perfil, deverá atribuí-lo à ação por meio de um ambiente. Para obter mais informações, consulte [Associação de um ambiente a uma ação](#).

URL da imagem do ECR, imagem do Docker Hub ou URL da imagem

Especifique um dos seguintes:

- Se você estiver usando um registro CODECATALYST, defina a imagem como uma das seguintes [imagens ativas](#):
  - CodeCatalystLinux\_x86\_64:2024\_03
  - CodeCatalystLinux\_x86\_64:2022\_11
  - CodeCatalystLinux\_Arm64:2024\_03
  - CodeCatalystLinux\_Arm64:2022\_11
  - CodeCatalystLinuxLambda\_x86\_64:2024\_03
  - CodeCatalystLinuxLambda\_x86\_64:2022\_11
  - CodeCatalystLinuxLambda\_Arm64:2024\_03
  - CodeCatalystLinuxLambda\_Arm64:2022\_11
  - CodeCatalystWindows\_x86\_64:2022\_11
- Se você estiver usando um registro do Docker Hub, defina a imagem como o nome da imagem do Docker Hub e a tag opcional.

Exemplo: postgres:latest

- Se estiver usando um registro do Amazon ECR, defina a imagem como o URI do registro do Amazon ECR.

Exemplo: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo

- Se estiver usando um registro personalizado, defina a imagem como o valor esperado pelo registro personalizado.

9. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para atribuir uma imagem do Docker de ambiente de runtime personalizada usando o editor YAML

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

2. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
3. Escolha Editar.
4. Selecione YAML.
5. Encontre a ação à qual você quer atribuir uma imagem do Docker do ambiente de runtime.
6. Na ação, adicione uma seção `Container` e as propriedades subjacentes `Registry` e `Image`. Para ter mais informações, consulte a descrição das propriedades `Container`, `Registry` e `Image` da [Ações](#) para sua ação.
7. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
8. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Exemplos

Os exemplos a seguir mostram como atribuir uma imagem do Docker de ambiente de runtime personalizada a uma ação no arquivo de definição do fluxo de trabalho.

### Tópicos

- [Exemplo: uso de uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o Amazon ECR](#)
- [Exemplo: uso de uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o Docker Hub](#)

Exemplo: uso de uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o Amazon ECR

O exemplo a seguir mostra como usar uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o [Amazon ECR](#).

```
Configuration:
 Container:
 Registry: ECR
 Image: public.ecr.aws/amazonlinux/amazonlinux:2023
```

Exemplo: uso de uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o Docker Hub

O exemplo a seguir mostra como usar uma imagem do Docker de ambiente de runtime personalizada para adicionar suporte ao Node.js 18 com o [Docker Hub](#).

```
Configuration:
 Container:
 Registry: DockerHub
 Image: node:18.18.2
```

## Conectar repositórios de origem aos fluxos de trabalho

Origem, também chamada de origem de entrada, é um repositório de origem ao qual uma [ação de fluxo de trabalho](#) se conecta para receber os arquivos necessários com o objetivo de realizar as operações. Por exemplo, uma ação de fluxo de trabalho pode se conectar a um repositório de origem para receber arquivos de origem a aplicação com o objetivo de criar uma aplicação.

CodeCatalyst os fluxos de trabalho oferecem suporte às seguintes fontes:

- CodeCatalyst repositórios de origem — Para obter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).
- GitHub repositórios, repositórios Bitbucket e repositórios de GitLab projetos — Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#)

### Tópicos

- [Especificar o repositório de origem de um arquivo de fluxo de trabalho](#)
- [Especificar o repositório de origem da ação do fluxo de trabalho](#)
- [Fazer referência a arquivos do repositório de origem](#)
- [variáveis BranchName " e CommitId "](#)

## Especificar o repositório de origem de um arquivo de fluxo de trabalho

Use as instruções a seguir para especificar o repositório de CodeCatalyst origem no qual você deseja armazenar seu arquivo de definição de fluxo de trabalho. Se você preferir especificar um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto, consulte em vez disso. [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#)

O repositório de origem em que seu arquivo de definição de fluxo de trabalho reside é identificado pelo rótulo `WorkflowSource`.

 Note

Especifique o repositório de origem em que seu arquivo de definição de fluxo de trabalho reside ao confirmar seu arquivo de definição de fluxo de trabalho pela primeira vez. Depois dessa confirmação, o repositório e o arquivo de definição do fluxo de trabalho são vinculados permanentemente. A única maneira de alterar o repositório após a confirmação inicial é recriar o fluxo de trabalho em um repositório diferente.

Como especificar o repositório de origem que armazenará o arquivo de definição do fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione Criar fluxo de trabalho para criar o fluxo de trabalho. Para obter mais informações, consulte [Criação de um fluxo de trabalho](#).

Durante o processo de criação do fluxo de trabalho, você pode especificar o CodeCatalyst repositório, a ramificação e a pasta em que deseja armazenar o arquivo de definição do fluxo de trabalho.

## Especificar o repositório de origem da ação do fluxo de trabalho

Use as instruções a seguir para especificar um repositório de origem a ser usado com uma ação de fluxo de trabalho. Na startup, a ação empacota os arquivos no repositório de origem configurado em um artefato, faz download do artefato na [imagem do Docker do ambiente de runtime](#) em que a ação está sendo executada e, depois, conclui seu processamento usando os arquivos baixados.

 Note

Atualmente, em uma ação de fluxo de trabalho, você só pode especificar um repositório de origem, que é o repositório de origem em que o arquivo de definição do fluxo de trabalho reside (no diretório `.codecatalyst/workflows/` ou em um de seus subdiretórios). Esse repositório de origem é representado pelo rótulo `WorkflowSource`.

## Visual

Como especificar o repositório de origem que uma ação usará (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação onde deseja especificar a origem.
8. Selecione Entradas.
9. Em Origens – opcional, faça o seguinte:

Especifique os rótulos que representam os repositórios de origem que serão necessários para a ação. Atualmente, o único rótulo compatível é WorkflowSource, que representa o repositório de origem em que o arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma origem, deverá especificar pelo menos um artefato de entrada em *action-name*/Inputs/Artifacts.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como especificar o repositório de origem que uma ação usará (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação, adicione um código semelhante ao seguinte:

```
action-name:
 Inputs:
 Sources:
 - WorkflowSource
```

Para ter mais informações, consulte a descrição da propriedade Sources em [Definição do YAML do fluxo de trabalho](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Fazer referência a arquivos do repositório de origem

Se você tiver arquivos que residem em um repositório de origem e precisar se referir a esses arquivos em uma das ações do fluxo de trabalho, conclua o procedimento a seguir.

### Note

Consulte também [Referência de arquivos em um artefato](#).

Como fazer referência a um arquivo armazenado em um repositório de origem

- Na ação em que você deseja fazer referência a um arquivo, adicione um código semelhante ao seguinte:

```
Actions:
 My-action:
 Inputs:
 Sources:
```

```
- WorkflowSource
Configuration:
Steps:
- run: cd my-app && cat file1.jar
```

No código anterior, a ação procura no diretório my-app na raiz do repositório de origem WorkflowSource para encontrar e exibir o arquivo file1.jar.

## variáveis BranchName " e CommitId "

A CodeCatalyst fonte produz BranchName e define CommitId variáveis quando seu fluxo de trabalho é executado. Elas são conhecidas como variáveis predefinidas. Veja a tabela a seguir para ter informações sobre essas variáveis.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave      | Valor                                                                                                                                                                                                                                                                                |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CommitId   | <p>O ID de confirmação que representa o estado do repositório no momento em que a execução do fluxo de trabalho foi iniciada.</p> <p>Exemplo: example3819261db00a3ab59468c8b</p> <p>Consulte também: <a href="#">Exemplo: fazer referência à variável predefinida "CommitId"</a></p> |
| BranchName | <p>O nome da ramificação em que a execução do fluxo de trabalho foi iniciada.</p> <p>Exemplos: main, feature/branch , test-LiJuan</p> <p>Consulte também: <a href="#">Exemplo: fazer referência à variável predefinida "BranchName"</a></p>                                          |



# Conectar repositórios de pacotes a fluxos de trabalho

Um pacote é um pacote que inclui o software e os metadados necessários para instalar o software e resolver quaisquer dependências. CodeCatalyst suporta o formato de pacote npm.

Um pacote consiste em:

- Um nome (por exemplo, webpack é o nome de um pacote npm conhecido)
- Um [namespace](#) opcional (por exemplo, @types em @types/node)
- Um conjunto de [versões](#) (por exemplo, 1.0.0, 1.0.1, 1.0.2)
- Metadados em nível de pacote (por exemplo, tags npm dist)

Em CodeCatalyst, você pode publicar e consumir pacotes de repositórios de CodeCatalyst pacotes em seus fluxos de trabalho. Você pode configurar uma ação de compilação ou teste com um repositório de CodeCatalyst pacotes para configurar automaticamente o cliente npm de uma ação para enviar e extrair pacotes do repositório especificado.

Para ter mais informações sobre pacotes, consulte [Publique e compartilhe pacotes de software no CodeCatalyst](#).

## Note

Atualmente, as ações de compilação e teste oferecem suporte a repositórios de CodeCatalyst pacotes.

## Tópicos

- [Tutorial: extrair de um repositório de pacotes](#)
- [Especificação de repositórios de CodeCatalyst pacotes em fluxos de trabalho](#)
- [Usar tokens de autorização em ações de fluxo de trabalho](#)
- [Exemplos: repositórios de pacotes em fluxos de trabalho](#)

## Tutorial: extrair de um repositório de pacotes

Neste tutorial, você aprende a criar um fluxo de trabalho que executa uma aplicação cujas dependências são extraídas de um [repositório de pacotes do CodeCatalyst](#). A aplicação é um

Node.js simples que imprime uma mensagem “Hello World” nos logs do CodeCatalyst. A aplicação tem uma única dependência: o pacote npm [lodash](#). O pacote `lodash` é usado para transformar uma string `hello-world` em `Hello World`. Você usará a versão 4.17.20 desse pacote.

Depois de configurar a aplicação e o fluxo de trabalho, você configura o CodeCatalyst para impedir que versões adicionais do `lodash` sejam importadas para o repositório de pacotes do CodeCatalyst a partir do registro externo público ([npmjs.com](#)). Depois, você testa se versões adicionais do `lodash` foram bloqueadas.

Ao final deste tutorial, você deve ter uma boa compreensão de como um fluxo de trabalho interage com repositórios de pacotes, dentro e fora do CodeCatalyst, para recuperar pacotes. Você também deve entender as interações nos bastidores que ocorrem entre o npm, seu repositório de pacotes, seu fluxo de trabalho e o arquivo `package.json` da aplicação.

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um repositório de origem](#)
- [Etapa 2: criar os repositórios de pacotes do CodeCatalyst e do gateway](#)
- [Etapa 3: criar a aplicação “Hello World”](#)
- [Etapa 4: criar um fluxo de trabalho que execute o “Hello World”](#)
- [Etapa 5: verificar o fluxo de trabalho](#)
- [Etapa 6: bloquear importações de npmjs.com](#)
- [Etapa 7: testar o recurso de bloqueio](#)
- [Limpeza](#)

## Pré-requisitos

Antes de começar:

- Você precisa de um espaço do CodeCatalyst. Para obter mais informações, consulte [Criar um espaço](#).
- No espaço do CodeCatalyst, você precisa de um projeto vazio chamado:

```
codecatalyst-package-project
```

Use a opção Começar do zero para criar esse projeto.

Para obter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

## Etapa 1: criar um repositório de origem

Nesta etapa, você cria um repositório de origem no CodeCatalyst. Esse repositório armazena os arquivos de origem do tutorial, como os arquivos `index.js` e `package.json`.

Para ter mais informações sobre repositórios de origem, consulte [Criar um repositório de origem](#).

Como criar um repositório de origem

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Navegue até o projeto, `codecatalyst-package-project`.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha Adicionar repositório e selecione Criar repositório.
5. Em Nome do repositório, insira:

**hello-world-app**

6. Escolha Criar.

## Etapa 2: criar os repositórios de pacotes do CodeCatalyst e do gateway

Nesta etapa, você cria um repositório de pacotes no projeto do CodeCatalyst e o conecta a um repositório de gateway, também no projeto do CodeCatalyst. Posteriormente, você importa a dependência do tutorial, `lodash`, de `npmjs.com` para os dois repositórios.

O repositório de gateway é a “cola” que conecta seu repositório de pacotes no CodeCatalyst ao público `npmjs.com`.

Para ter mais informações sobre repositórios de pacote, consulte [Publique e compartilhe pacotes de software no CodeCatalyst](#).

### Note

Este tutorial usa os termos repositório de pacotes do CodeCatalyst e repositório de gateway para se referir aos dois repositórios que você cria no CodeCatalyst no procedimento a seguir.

Para criar repositórios de pacotes do CodeCatalyst e do gateway

1. No painel de navegação, selecione Packages (pacotes).
2. Escolha Criar repositório de pacotes.
3. Em Nome do repositório, insira:

**codecatalyst-package-repository**

4. Escolha + Selecionar repositórios upstream.
5. Escolha Repositórios de gateway.
6. Na caixa npm-public-registry-gateway, selecione Criar.
7. Escolha Selecionar.
8. Escolha Criar.

O CodeCatalyst cria um repositório de pacotes chamado `codecatalyst-package-repository` que está conectado a um repositório de gateway. O repositório de gateway está conectado ao registro `npmjs.com`.

### Etapa 3: criar a aplicação “Hello World”

Nesta etapa, você cria uma aplicação Node.js “Hello World” e importa a dependência (lodash) para os repositórios de pacotes do gateway e do CodeCatalyst.

Para criar a aplicação, você precisa de uma máquina de desenvolvimento com o Node.js e o cliente npm associado instalado.

Este tutorial pressupõe que você usará um Ambiente de Desenvolvimento do CodeCatalyst como máquina de desenvolvimento. Embora você não precise usar um Ambiente de Desenvolvimento do CodeCatalyst, ele é recomendado porque fornece um ambiente de trabalho limpo, tem o Node.js e o npm pré-instalados e é fácil de excluir quando você terminar o tutorial. Para ter mais informações sobre Ambientes de Desenvolvimento do CodeCatalyst, consulte [Criar um Ambiente de Desenvolvimento](#).

Use as instruções a seguir para iniciar um Ambiente de Desenvolvimento do CodeCatalyst e usá-lo para criar a aplicação “Hello World”.

## Como iniciar um novo Ambiente de Desenvolvimento do CodeCatalyst

1. No painel de navegação, escolha Código e Ambientes de Desenvolvimento.
2. Na parte superior, escolha Criar ambiente de desenvolvimento e AWS Cloud9 (no navegador).
3. Verifique se Repositório está definido como `hello-world-app` e Ramificação existente está definido como `main`. Escolha Criar.

O Ambiente de Desenvolvimento é iniciado em uma nova guia do navegador e seu repositório (`hello-world-app`) é clonado nela.

4. Deixe as duas guias do navegador do CodeCatalyst abertas e vá para o próximo procedimento.

## Como criar a aplicação Node.js “Hello World”

1. Acesse o Ambiente de Desenvolvimento.
2. No prompt do terminal, mude para o diretório raiz do repositório de origem `hello-world-app`:

```
cd hello-world-app
```

3. Inicialize um projeto Node.js:

```
npm init -y
```

A inicialização cria um arquivo `package.json` no diretório raiz do `hello-world-app`.

4. Conecte o cliente npm no Ambiente de Desenvolvimento ao repositório de pacotes do CodeCatalyst:

1. Mude para o console do CodeCatalyst.
2. No painel de navegação, selecione Packages (pacotes).
3. Selecione `codecatalyst-package-repository`.
4. Selecione Conectar ao repositório.
5. Selecione Criar token. Um token de acesso pessoal (PAT) é criado para você.
6. Selecione Copiar para copiar os comandos.
7. Mude para o Ambiente de Desenvolvimento.
8. Verifique se você está no diretório `hello-world-app`.

## 9. Cole os comandos. Eles são semelhantes a:

Tutorial: extrair de um repositório de pacotes

```
npm set registry=https://packages.us-west-2.codecatalyst.aws/npm/ExampleCompany/
codecatalyst-package-project/codecatalyst-package-repository/ --location project
npm set //packages.us-west-2.codecatalyst.aws/npm/ExampleCompany/codecatalyst-
package-project/hello-world-app/:_authToken=username:token-secret
```

5. Importe a versão 4.17.20 de lodash:

```
npm install lodash@v4.17.20 --save --save-exact
```

O npm procura a versão 4.17.20 de lodash nos seguintes locais, na seguinte ordem:

- No Ambiente de Desenvolvimento. Ele não consegue encontrá-lo aqui.
- No repositório de pacotes do CodeCatalyst. Ele não consegue encontrá-lo aqui.
- No repositório de gateway. Ele não consegue encontrá-lo aqui.
- Em npmjs.com. Ele o encontra aqui.

O npm importa lodash para o repositório de gateway, o repositório de pacotes do CodeCatalyst e o Ambiente de Desenvolvimento.

 Note

Se você não tivesse conectado o cliente npm ao repositório de pacotes do CodeCatalyst na etapa 4, o npm teria retirado lodash diretamente do npmjs.com e não teria importado o pacote para nenhum dos repositórios.

O npm também atualiza seu arquivo `package.json` com a dependência lodash e cria um diretório `node_modules` contendo lodash e todas as dependências.

6. Teste se lodash foi importado para seu Ambiente de Desenvolvimento. Insira:

```
npm list
```

A mensagem a seguir é exibida, indicando uma importação bem-sucedida:

```
`-- lodash@4.17.20
```

7. (Opcional) Abra `hello-world-app/package.json` e verifique se as linhas em **negrito vermelho** foram adicionadas:

```
{
 "name": "hello-world-app",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
 "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "dependencies": {
 "lodash": "4.17.20"
 }
}
```

8. Em `/hello-world-app`, crie um arquivo chamado `index.js` com o conteúdo a seguir:

 Tip

Você pode usar a navegação lateral no Ambiente de Desenvolvimento para criar esse arquivo.

```
// Importing lodash library
const _ = require('lodash');

// Input string
const inputString = 'hello-world';

// Transforming the string using lodash
const transformedString = _.startCase(inputString.replace('-', ' '));

// Outputting the transformed string to the console
console.log(transformedString);
```

Para testar se “lodash” foi importado para os repositórios de pacotes do gateway e do CodeCatalyst

1. Mude para o console do CodeCatalyst.
2. No painel de navegação, selecione Packages (pacotes).
3. Selecione npm-public-registry-gateway.
4. Certifique-se de que lodash seja exibido. A coluna Versão mais recente indica 4.17.20.
5. Repita esse procedimento para o codecatalyst-package-repository. Talvez seja necessário atualizar a janela do navegador para ver o pacote importado.

Para testar o “Hello World” em seu Ambiente de Desenvolvimento

1. Mude para o Ambiente de Desenvolvimento.
2. Verifique se você ainda está no diretório hello-world-app e execute a aplicação:

```
node index.js
```

Uma mensagem Hello World é exibida. O Node.js executou a aplicação usando o pacote lodash que você baixou para o Ambiente de Desenvolvimento em uma etapa anterior.

Como ignorar o diretório “node\_modules” e confirmar “Hello World”

1. Ignore o diretório node\_modules. Insira:

```
echo "node_modules/" >> .gitignore
```

É uma prática recomendada evitar confirmar esse diretório. Além disso, a confirmação desse diretório interferirá nas etapas posteriores neste tutorial.

2. Adicione, confirme e envie:

```
git add .
git commit -m "add the Hello World application"
git push
```

Os arquivos da aplicação e do projeto “Hello World” são adicionados ao repositório de origem.



## Etapa 4: criar um fluxo de trabalho que execute o “Hello World”

Nesta etapa, crie um fluxo de trabalho que executará a aplicação “Hello World” usando a dependência `lodash`. O fluxo de trabalho inclui uma única ação ou tarefa chamada `RunHelloWorldApp`. A ação `RunHelloWorldApp` inclui os seguintes comandos e seções notáveis:

- **Packages**

Esta seção indica o nome do repositório de pacotes do CodeCatalyst ao qual a ação deve se conectar durante a execução de `npm install`.

- **- Run: `npm install`**

Esse comando instrui o `npm` a instalar as dependências especificadas no arquivo `package.json`. A única dependência especificada no arquivo `package.json` é `lodash`. O `npm` procura `lodash` nos seguintes locais:

- Na imagem do Docker que executa a ação. Ele não consegue encontrá-lo aqui.
- No repositório de pacotes do CodeCatalyst. Ele o encontra aqui.

Depois que o `npm` encontra `lodash`, ele o importa para a imagem do Docker que executa a ação.

- **- Run: `npm list`**

Esse comando imprime qual versão do `lodash` foi baixada para a imagem do Docker que executa a ação.

- **- Run: `node index.js`**

Esse comando executa a aplicação “Hello World” usando a dependência especificada no arquivo `package.json`.

Observe que a ação `RunHelloWorldApp` é uma ação de criação, conforme indicado pelo identificador `aws/build@v1` próximo à parte superior do fluxo de trabalho. Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).

Use as instruções a seguir para criar um fluxo de trabalho que extraia a dependência `lodash` do repositório de pacotes do CodeCatalyst e, depois, execute a aplicação “Hello World”.

## Como criar um fluxo de trabalho

1. Mude para o console do CodeCatalyst.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione Criar fluxo de trabalho.
4. Em Repositório de origem, selecione hello-world-app.
5. Em Ramificação, selecione main.

O arquivo de definição do fluxo de trabalho será criado no repositório de origem e na ramificação escolhidos.

6. Escolha Criar.
7. Selecione YAML na parte superior.
8. Exclua o código de amostra YAML.
9. Adicione o seguinte código YAML:

```
Name: codecatalyst-package-workflow
SchemaVersion: "1.0"

Required - Define action configurations.
Actions:
 RunHelloWorldApp:
 # Identifies the action. Do not modify this value.
 Identifier: aws/build@v1
 Compute:
 Type: Lambda
 Inputs:
 Sources:
 - WorkflowSource # This specifies your source repository.
 Configuration:
 Steps:
 - Run: npm install
 - Run: npm list
 - Run: node index.js
 Container: # This specifies the Docker image that runs the action.
 Registry: CODECATALYST
 Image: CodeCatalystLinuxLambda_x86_64:2024_03
 Packages:
 NpmConfiguration:
 PackageRegistries:
```

- PackagesRepository: *codecatalyst-package-repository*

No código anterior, substitua *codecatalyst-package-repository* pelo nome do repositório de pacotes do CodeCatalyst que você criou em [Etapa 2: criar os repositórios de pacotes do CodeCatalyst e do gateway](#).

Para ter informações sobre as propriedades nesse arquivo, consulte a [Ações de criação e de teste YAML](#).

10. (Opcional) Selecione Validar para garantir que o código YAML seja válido antes de confirmar.
11. Selecione Confirmar.
12. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
  - a. Em Nome do arquivo do fluxo de trabalho, mantenha o padrão, `codecatalyst-package-workflow`.
  - b. Em Confirmar mensagem, insira:

**add initial workflow file**

- c. Em Repositório, selecione hello-world-app.
- d. Em Nome da ramificação, selecione principal.
- e. Selecione Confirmar.

Agora você criou um fluxo de trabalho.

Para executar o fluxo de trabalho

1. Ao lado do fluxo de trabalho que você acabou de criar (`codecatalyst-package-workflow`), selecione Ações e, depois, selecione Executar.

A execução do fluxo de trabalho é iniciada.

2. Na notificação verde na parte superior, à direita, escolha o link para a execução. O link é semelhante a View Run-1234.

Um diagrama do fluxo de trabalho é exibido, mostrando quem iniciou a execução e a ação RunHelloWorldApp.

3. Escolha a caixa RunHelloWorldApp para observar o progresso da ação.

- Quando a execução terminar, vá para [Etapa 5: verificar o fluxo de trabalho](#).

## Etapa 5: verificar o fluxo de trabalho

Nesta etapa, você verifica se o fluxo de trabalho executou a aplicação “Hello World” com a dependência `lodash`.

Como verificar se a aplicação “Hello World” foi executada usando a dependência

- No diagrama do fluxo de trabalho, escolha a caixa `RunHelloWorldApp`.

Uma lista de mensagens de log é exibida.

- Expanda a mensagem de log `node index.js`.

A seguinte mensagem é exibida:

```
[Container] 2024/04/24 21:15:41.545650 Running command node index.js
Hello World
```

A aparência de `Hello Word` (em vez de `hello-world`) indica que a dependência `lodash` foi usada.

- Expanda o log `npm list`.

Uma mensagem semelhante à seguinte é exibida:

```
lodash@4.17.20
```

Essa mensagem indica que a versão 4.17.20 de `lodash` foi baixada para a imagem do Docker que executa a ação do fluxo de trabalho.

## Etapa 6: bloquear importações de `npmjs.com`

Agora que a versão 4.17.20 de `lodash` está presente nos repositórios de pacotes do gateway e do CodeCatalyst, você pode bloquear importações de outras versões. O bloqueio impede que você importe acidentalmente versões posteriores (ou anteriores) do `lodash`, que podem conter código malicioso. Para obter mais informações, consulte [Editar controles de origem do pacote](#) e [Ataques de substituição de dependências](#).

Use as instruções a seguir para bloquear as importações de `lodash` do seu repositório de gateway. Quando você bloqueia pacotes no gateway, eles também são bloqueados em locais posteriores.

Para bloquear importações para o repositório de gateway

1. No painel de navegação, selecione Packages (pacotes).
2. Selecione `npm-publish-registry-gateway`.
3. Selecione `lodash`.
4. Na parte superior, selecione Controles de origem.
5. Em Upstream, selecione Bloquear.
6. Escolha Salvar.

Agora você bloqueou as importações para seu repositório de gateway (e repositórios e computadores downstream) em `npmjs.com`.

## Etapa 7: testar o recurso de bloqueio

Nesta seção, você verifica se o bloqueio configurado em [Etapa 6: bloquear importações de `npmjs.com`](#) está funcionando. Você começa configurando o “Hello World” para solicitar a versão 4.17.21 de `lodash` em vez da disponível no seu repositório de gateway, que é a 4.17.20. Depois, você confere se a aplicação não pode extrair a versão 4.17.21 do `npmjs.com`, indicando um bloqueio bem-sucedido. Como teste final, você desbloqueia as importações para o repositório do gateway e confere se a aplicação consegue extrair a versão 4.17.21 de `lodash`.

Use o conjunto de procedimentos a seguir para testar o recurso de bloqueio.

Antes de começar

1. Mude para o Ambiente de Desenvolvimento.
2. Extraia o arquivo `codecatalyst-package-workflow.yaml` que você criou usando o console do CodeCatalyst anteriormente:

```
git pull
```

Como configurar “Hello World” para solicitar a versão 4.17.21 de “`lodash`”

1. Abra o `/hello-world-app/package.json`.

2. Altere a versão de `lodash` para 4.17.21 conforme mostrado em **negrito vermelho**:

```
{
 "name": "hello-world-app",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
 "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "dependencies": {
 "lodash": "4.17.21"
 }
}
```

Agora há uma incompatibilidade entre a versão no arquivo `package.json` (4.17.21) e a versão nos repositórios de pacotes do gateway e do CodeCatalyst (4.17.20).

3. Adicione, confirme e envie:

```
git add .
git commit -m "update package.json to use lodash 4.17.21"
git push
```

Como testar se o “Hello World” não pode receber a versão 4.17.21 de “lodash”

1. Execute o fluxo de trabalho com a incompatibilidade de versão:
  1. Mude para o console do CodeCatalyst.
  2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  3. Ao lado de `codecatalyst-package-workflow`, selecione Ação e Executar.

O npm examina em `package.json` as dependências e vê que a versão 4.17.21 de `lodash` é exigida pelo “Hello World”. O npm procura a dependência nos seguintes locais, na seguinte ordem:

- Na imagem do Docker que executa a ação. Ele não consegue encontrá-lo aqui.

- No repositório de pacotes do CodeCatalyst. Ele não consegue encontrá-lo aqui.
- No repositório de gateway. Ele não consegue encontrá-lo aqui.
- Em npmjs.com. Ele o encontra aqui.

Depois que o npm encontra a versão 4.17.21 em npmjs.com, ele tenta importá-la para o repositório do gateway, mas como você configura o gateway para bloquear as importações de lodash, a importação não ocorre.

Como a importação não ocorre, o fluxo de trabalho falha.

## 2. Verifique se o fluxo de trabalho falhou:

1. Na notificação verde na parte superior, à direita, escolha o link para a execução. O link é semelhante a `View Run-2345`.
2. No diagrama do fluxo de trabalho, escolha a caixa `RunHelloWorldApp`.
3. Expanda a mensagem de log `npm install`.

A seguinte mensagem é exibida:

```
[Container] 2024/04/25 17:20:34.995591 Running command npm install
npm ERR! code ETARGET
npm ERR! notarget No matching version found for lodash@4.17.21.
npm ERR! notarget In most cases you or one of your dependencies are requesting
npm ERR! notarget a package version that doesn't exist.

npm ERR! A complete log of this run can be found in: /tmp/.npm/
_logs/2024-05-08T22_03_26_493Z-debug-0.log
```

O erro indica que a versão 4.17.21 não foi encontrada. Isso é esperado porque você a bloqueou.

Para desbloquear importações de npmjs.com

1. No painel de navegação, selecione Packages (pacotes).
2. Selecione npm-publish-registry-gateway.
3. Selecione lodash.
4. Na parte superior, selecione Controles de origem.
5. Em Upstream, selecione Permitir.

## 6. Escolha Salvar.

Agora você desbloqueou as importações de `lodash`.

Seu fluxo de trabalho agora pode importar a versão 4.17.21 de `lodash`.

Para testar se as importações de `npmjs.com` estão desbloqueadas

1. Execute o fluxo de trabalho novamente. Desta vez, o fluxo de trabalho deve ser bem-sucedido porque a importação de 4.17.21 agora deve funcionar. Para executar o fluxo de trabalho novamente:

1. Selecione CI/CD e, depois, selecione Fluxos de trabalho.
2. Ao lado de `codecatalyst-package-workflow`, selecione Ações e Executar.
3. Na notificação verde na parte superior, à direita, escolha o link para a execução. O link é semelhante a `View Run-3456`.

Um diagrama do fluxo de trabalho é exibido, mostrando quem iniciou a execução e a ação `RunHelloWorldApp`.

4. Escolha a caixa `RunHelloWorldApp` para observar o progresso da ação.
5. Expanda a mensagem de log `npm list` e verifique se uma mensagem semelhante à seguinte é exibida:

```
lodash@4.17.21
```

Essa mensagem indica que a versão 4.17.21 de `lodash` foi baixada.

2. Verifique se a versão 4.17.21 foi importada para seus repositórios do CodeCatalyst e de gateway:
  1. No painel de navegação, selecione Packages (pacotes).
  2. Selecione `npm-public-registry-gateway`.
  3. Encontre `lodash` e verifique se a versão é 4.17.21.



 Note

Embora a versão 4.17.20 não esteja listada nesta página, você pode encontrá-la escolhendo `lodash` e Versões na parte superior.

4. Repita essas etapas para verificar se a versão 4.17.21 foi importada para o `codecatalyst-package-repository`.

## Limpeza

Limpe os arquivos e serviços usados neste tutorial para evitar cobranças por eles.

Para limpar os pacotes (tutorial)

1. Exclua o `codecatalyst-package-project`:
  - a. No console do CodeCatalyst, navegue até o projeto `codecatalyst-package-project` se você ainda não estiver lá.
  - b. No painel de navegação, escolha Configurações do projeto.
  - c. Selecione Excluir projeto, insira **delete** e selecione Excluir projeto.

O CodeCatalyst exclui todos os recursos do projeto, incluindo os repositórios de origem, gateway e pacotes do CodeCatalyst. O Ambiente de Desenvolvimento também é excluído.

2. Exclua o token PAT:
  - a. Escolha seu nome de usuário à direita e selecione Minhas configurações.
  - b. Em Tokens de acesso pessoal, escolha o token que você criou neste tutorial e selecione Excluir.

Neste tutorial, você aprendeu a criar um fluxo de trabalho que executa uma aplicação que extrai as dependências de um repositório de pacotes do CodeCatalyst. Você também aprendeu a bloquear e desbloquear a entrada de pacotes nos repositórios de pacotes do gateway e do CodeCatalyst.

## Especificação de repositórios de CodeCatalyst pacotes em fluxos de trabalho

Em CodeCatalyst, você pode adicionar um repositório de CodeCatalyst pacotes às suas ações de criação e teste em seu fluxo de trabalho. Seu repositório de pacotes deve ser configurado com um formato de pacote, como npm. Você também pode optar por incluir uma sequência de escopos para o repositório de pacotes selecionado.

Use as instruções a seguir para especificar uma configuração de pacote a ser usada com uma ação de fluxo de trabalho.

### Visual

Como especificar a configuração do pacote que uma ação usará (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação Criar ou Testar com a qual você deseja configurar com um repositório de pacotes.
8. Selecione Pacotes.
9. No menu suspenso Adicionar configuração, selecione a configuração de pacote que você deseja usar com as ações do fluxo de trabalho.
10. Selecione Adicionar repositório de pacotes.
11. No menu suspenso Package repository, especifique o nome do repositório de CodeCatalyst pacotes que você deseja que a ação use.

Para ter mais informações sobre repositórios de pacote, consulte [Repositórios de pacotes](#).

12. (Opcional) Em Escopos – opcional, especifique uma sequência de escopos que deseja definir no registro do pacote.

Na definição de escopos, o repositório de pacotes especificado é configurado como o registro para todos os escopos listados. Se um pacote com o escopo for solicitado por meio do cliente npm, ele usará esse repositório em vez do padrão. Cada nome de escopo deve ter o prefixo “@”.

Se Scopes for omitido, o repositório de pacotes especificado será configurado como o registro padrão para todos os pacotes usados pela ação.

Para ter mais informações sobre escopos, consulte [Namespaces de pacote](#) e [Pacotes com escopo definido](#).

13. Escolha Adicionar.
14. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
15. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Como especificar a configuração do pacote que uma ação usará (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação Criar ou Testar, adicione um código semelhante ao seguinte:

```
action-name:
 Configuration:
 Packages:
 NpmConfiguration:
 PackageRegistries:
 - PackagesRepository: package-repository
 Scopes:
```

- "@scope"

Para ter mais informações, consulte a descrição da propriedade Packages em [Ações de criação e de teste YAML](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Usar tokens de autorização em ações de fluxo de trabalho

Você pode usar um token fornecido pela ação do fluxo de trabalho para configurar manualmente um gerenciador de pacotes para se autenticar com repositórios de CodeCatalyst pacotes. CodeCatalyst disponibiliza esse token como uma variável de ambiente para você referenciar em suas ações.

| Variável de ambiente                  | Valor                                            |
|---------------------------------------|--------------------------------------------------|
| CATALYST_MACHINE_RESOURCE_NAME        | A identidade do usuário do token de autorização. |
| CATALYST_PACKAGES_AUTHORIZATION_TOKEN | O valor do token de autorização.                 |

### Note

Observe que essas variáveis de ambiente só serão preenchidas se você tiver configurado sua ação para exportar o token de autorização.

Use as instruções a seguir para usar um token de autorização com uma ação de fluxo de trabalho.

### Visual

Como usar um token de autorização exportado com uma ação (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.

3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação Criar ou Testar com a qual você deseja configurar com um repositório de pacotes.
8. Selecione Pacotes.
9. Ative o token de autorização de exportação.

## YAML

Como usar um token de autorização exportado com uma ação (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação Criar ou Testar, adicione um código semelhante ao seguinte:

```
Actions:
 action-name:
 Packages:
 ExportAuthorizationToken: true
```

Você pode referenciar as variáveis de ambiente `$CATALYST_MACHINE_RESOURCE_NAME` e `$CATALYST_PACKAGES_AUTHORIZATION_TOKEN` na seção Steps do seu YAML. Para obter mais informações, consulte [Exemplo: configuração manual pip para autenticação com CodeCatalyst](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Exemplos: repositórios de pacotes em fluxos de trabalho

Os exemplos a seguir mostram como referenciar pacotes no arquivo de definição de fluxo de trabalho.

### Tópicos

- [Exemplo: definição de pacotes com NpmConfiguration](#)
- [Exemplo: substituição do registro padrão](#)
- [Exemplo: substituição de escopos em seu registro de pacotes](#)
- [Exemplo: configuração manual pip para autenticação com CodeCatalyst](#)

### Exemplo: definição de pacotes com **NpmConfiguration**

O exemplo a seguir mostra como definir um pacote com NpmConfiguration no arquivo de definição do fluxo de trabalho.

```
Actions:
 Build:
 Identifier: aws/build-beta@v1
 Configuration:
 Packages:
 NpmConfiguration:
 PackageRegistries:
 - PackagesRepository: main-repo
 - PackagesRepository: scoped-repo
 Scopes:
 - "@scope1"
```

Este exemplo configura o cliente npm da seguinte forma:

```
default: main-repo
@scope1: scoped-repo
```

Neste exemplo, há dois repositórios definidos. O registro padrão é definido como `main-repo` sem um escopo. O escopo `@scope1` está configurado em `PackageRegistries` para `scoped-repo`.

### Exemplo: substituição do registro padrão

O exemplo a seguir mostra como substituir o registro padrão.

```
NpmConfiguration:
 PackageRegistries:
 - PackagesRepository: my-repo-1
 - PackagesRepository: my-repo-2
 - PackagesRepository: my-repo-3
```

Este exemplo configura o cliente npm da seguinte forma:

```
default: my-repo-3
```

Se você especificar vários repositórios padrão, o último repositório terá prioridade. Neste exemplo, o último repositório listado é `my-repo-3`, o que significa que o npm se conectará a `my-repo-3`. Isso substitui os repositórios `my-repo-1` e `my-repo-2`.

### Exemplo: substituição de escopos em seu registro de pacotes

O exemplo a seguir mostra como substituir um escopo em seu registro de pacotes.

```
NpmConfiguration:
 PackageRegistries:
 - PackagesRepository: my-default-repo
 - PackagesRepository: my-repo-1
 Scopes:
 - "@scope1"
 - "@scope2"
 - PackagesRepository: my-repo-2
 Scopes:
 - "@scope2"
```

Este exemplo configura o cliente npm da seguinte forma:

```
default: my-default-repo
@scope1: my-repo-1
```

```
@scope2: my-repo-2
```

Se você incluir escopos substitutos, o último repositório terá prioridade. Neste exemplo, a última vez em que o escopo `@scope2` foi configurado em `PackageRegistries` é para `my-repo-2`. Isso substitui o escopo `@scope2` configurado para `my-repo-1`.

## Exemplo: configuração manual **pip** para autenticação com CodeCatalyst

O exemplo a seguir mostra como referenciar variáveis de ambiente de CodeCatalyst autorização em uma ação de criação.

```
Actions:
 Build:
 Identifier: aws/build@v1.0.0
 Configuration:
 Steps:
 - Run: pip config set global.index-url https://$CATALYST_MACHINE_RESOURCE_NAME:
$CATALYST_PACKAGES_AUTHORIZATION_TOKEN@codecatalyst.aws/pypi/my-space/my-project/my-
repo/simple/
 Packages:
 ExportAuthorizationToken: true
```

## Invocar uma função do Lambda usando um fluxo de trabalho

Esta seção descreve como invocar uma AWS Lambda função usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar a ação `Invocação do AWS Lambda` ao seu fluxo de trabalho. A ação `Invocação do AWS Lambda` invoca a função do Lambda que você especifica.

Além de invocar sua função, a ação `Invocação do AWS Lambda` também converte cada chave de nível superior na carga útil de resposta recebida da função do Lambda em uma [variável de saída do fluxo de trabalho](#). Essas variáveis podem ser referenciadas em ações subsequentes do fluxo de trabalho. Se você não quiser que todas as chaves de nível superior sejam convertidas em variáveis, use filtros para especificar as chaves exatas. Para ter mais informações, consulte a descrição da propriedade [ResponseFilters](#) na [YAML da ação “Invocação do AWS Lambda”](#).

### Tópicos

- [Quando usar essa ação](#)
- [Imagem de tempo de execução usada pela AWS Lambda ação 'invocar'](#)
- [Exemplo: invocar uma função do Lambda](#)



- [Adicionando a ação 'AWS Lambda invocar'](#)
- [Variáveis de “Invocação do AWS Lambda ”](#)
- [YAML da ação “Invocação do AWS Lambda ”](#)

## Quando usar essa ação

Use essa ação se quiser adicionar funcionalidade ao seu fluxo de trabalho que é encapsulada e executada por uma função do Lambda.

Por exemplo, talvez você queira que seu fluxo de trabalho envie uma notificação `Build started` para um canal do Slack antes de começar a criar a aplicação. Nesse caso, seu fluxo de trabalho incluiria uma ação `Invocação do AWS Lambda` para invocar um Lambda para enviar a notificação do Slack e uma [ação de criação](#) para criar a aplicação.

Como outro exemplo, você talvez queira que o fluxo de trabalho realize uma verificação de vulnerabilidade na aplicação antes de ser implantada. Nesse caso, você usaria uma ação de criação para criar a aplicação, uma ação `Invocação do AWS Lambda` para invocar um Lambda para verificar vulnerabilidades e uma ação de implantação para implantar a aplicação verificada.

## Imagem de tempo de execução usada pela AWS Lambda ação 'invocar'

A ação `Invocação do AWS Lambda` é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Exemplo: invocar uma função do Lambda

O exemplo de fluxo de trabalho a seguir inclui a ação `Invocação do AWS Lambda`, junto com uma ação de implantação. O fluxo de trabalho envia uma notificação do Slack indicando que uma implantação foi iniciada e, em seguida, implanta um aplicativo AWS usando um CloudFormation modelo. O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação `Invocação do AWS Lambda (LambdaNotify)` — No gatilho, essa ação invoca a função do `Lambda Notify-Start` na conta da AWS e região especificadas (`my-aws-account` e

us-west-2). Na invocação, a função do Lambda envia uma notificação do Slack indicando que a implantação foi iniciada.

- Uma ação Deploy CloudFormation stack (Deploy) — Ao concluir a ação de AWS Lambda invocação, a ação Deploy CloudFormation stack executa o template (`cfn-template.yml`) para implantar sua pilha de aplicativos. Para obter mais informações sobre a ação Deploy CloudFormation stack, consulte [Implantação de uma pilha CloudFormation](#).

#### Note

O exemplo de fluxo de trabalho a seguir serve para fins ilustrativos e não funcionará sem configuração adicional.

#### Note

No código YAML a seguir, você pode omitir as seções `Connections:` se quiser. Se você omitir essas seções, deverá garantir que a função especificada no campo Função padrão do IAM em seu ambiente inclua as permissões e as políticas de confiança exigidas pelas ações de AWS Lambda invocação e implantação da CloudFormation pilha. Para ter mais informações sobre como configurar um ambiente com um perfil do IAM padrão, consulte [Criar um ambiente](#). Para obter mais informações sobre as permissões e as políticas de confiança exigidas pelas ações AWS Lambda invoke e Deploy CloudFormation stack, consulte a descrição da Role propriedade em e. [YAML da ação “Invocação do AWS Lambda”](#) [Ação “Implantar CloudFormation pilha” YAML](#)

```
Name: codecatalyst-lambda-invoke-workflow
```

```
SchemaVersion: 1.0
```

```
Triggers:
```

```
- Type: PUSH
```

```
 Branches:
```

```
 - main
```

```
Actions:
```

```
 LambdaNotify:
```

```
 Identifier: aws/lambda-invoke@v1
```

```
 Environment:
```

```
 Name: my-production-environment
```

```
Connections:
 - Name: my-aws-account
 Role: codecatalyst-lambda-invoke-role
Inputs:
 Sources:
 - WorkflowSource
Configuration:
 Function: Notify-Start
 AWSRegion: us-west-2

Deploy:
 Identifier: aws/cfn-deploy@v1
 Environment:
 Name: my-production-environment
 Connections:
 - Name: my-aws-account
 Role: codecatalyst-deploy-role
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 name: my-application-stack
 region: us-west-2
 role-arn: arn:aws:iam::111122223333:role/StackRole
 template: ./cfn-template.yml
 capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
```

## Adicionando a ação 'AWS Lambda invocar'

Use as instruções a seguir para adicionar a ação **Invocação do AWS Lambda** ao seu fluxo de trabalho.

### Pré-requisito

Antes de começar, certifique-se de que sua AWS Lambda função e a função de execução do Lambda associada estejam prontas e disponíveis no. AWS Para ter mais informações, consulte [Perfil de execução do Lambda](#) no Guia do desenvolvedor do AWS Lambda .

### Visual

Para adicionar a ação 'AWS Lambda invocar' usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione Visual.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Invocação do AWS Lambda e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione Invocação do AWS Lambda . A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
  10. Nas guias Entradas, Configuração e Saídas, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [YAML da ação “Invocação do AWS Lambda”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação 'AWS Lambda invocar' usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.

3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione YAML.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Procure a ação Invocação do AWS Lambda e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione Invocação do AWS Lambda . A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).
    - Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [YAML da ação “Invocação do AWS Lambda”](#).
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Variáveis de “Invocação do AWS Lambda ”

Por padrão, a ação Invocação do AWS Lambda produz uma variável por chave de nível superior na carga útil de resposta do Lambda.

Por exemplo, se a carga útil de resposta for semelhante a esta:

```
responsePayload = {
 "name": "Saanvi",
```

```
"location": "Seattle",
"department": {
 "company": "Amazon",
 "team": "AWS"
}
}
```

... a ação vai gerar as seguintes variáveis.

| Chave      | Valor                                |
|------------|--------------------------------------|
| name       | Saanvi                               |
| location   | Seattle                              |
| department | {"company": "Amazon", "team": "AWS"} |

#### Note

Você pode alterar quais variáveis são geradas usando a propriedade YAML `ResponseFilters`. Para obter mais informações, consulte o [ResponseFilters](#) no [YAML da ação “Invocação do AWS Lambda”](#).

As variáveis produzidas e definidas pela ação 'AWS Lambda invoke' em tempo de execução são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

## YAML da ação “Invocação do AWS Lambda”

Confira a seguir a definição de YAML da ação Invocação do AWS Lambda . Para saber como usar essa ação, consulte [Invocar uma função do Lambda usando um fluxo de trabalho](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

 Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```
The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
LambdaInvoke_nn:
 Identifier: aws/lambda-invoke@v1
 DependsOn:
 - dependent-action
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - request-payload
 Variables:
 - Name: variable-name-1
 Value: variable-value-1
 - Name: variable-name-2
 Value: variable-value-2
 Environment:
 Name: environment-name
 Connections:
 - Name: account-connection-name
 Role: iam-role-name
 Configuration:
 Function: my-function/function-arn
 AWSRegion: us-west-2
```

```
Specify RequestPayload or RequestPayloadFile, but not both.
RequestPayload: '{"firstname": "Li", lastname: "Jean", "company": "ExampleCo",
"team": "Development"}'
RequestPayloadFile: my/request-payload.json
ContinueOnError: true/false
LogType: Tail/None
ResponseFilters: '{"name": ".name", "company": ".department.company"}'
Outputs:
 Artifacts:
 - Name: lambda_artifacts
 Files:
 - "lambda-response.json"
```

## LambdaInvoke

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: Lambda\_Invoke\_Action\_Workflow\_nn.

Interface de usuário correspondente: guia Configuração/Nome da ação

### Identifier

(*LambdaInvoke*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: aws/lambda-invoke@v1.

Interface de usuário correspondente: diagrama de fluxo de trabalho/rótuloLambdaInvoke\_nn/aws/lambda-invoke@v1

### DependsOn

(*LambdaInvoke*/DependsOn)



(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

## Compute

(*LambdaInvoke*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

## Type

(*LambdaInvoke*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

Interface de usuário correspondente: guia Configuração/Tipo de computação

## Fleet

(*LambdaInvoke*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

Interface de usuário correspondente: guia Configuração/Frota de computação

## Timeout

(*LambdaInvoke*/Timeout)

(Obrigatório)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Inputs

(*LambdaInvoke*/Inputs)

(Obrigatório)

A seção `Inputs` define os dados que a ação Invocação do AWS Lambda precisa durante a execução de um fluxo de trabalho.

**Note**

Somente uma entrada (uma origem ou um artefato) é permitida por ação Invocação do AWS Lambda . As variáveis não são contadas nesse total.

Interface de usuário correspondente: guia Entradas

## Sources

(*LambdaInvoke*/Inputs/Sources)

(Obrigatório se [RequestPayloadFile](#) for fornecido)

Se você quiser transmitir um arquivo JSON de carga útil de solicitação para a ação Invocação do AWS Lambda , e esse arquivo de carga útil estiver armazenado em um repositório de origem, especifique o rótulo desse repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se o arquivo de carga útil de solicitação não estiver em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para ter mais informações sobre o arquivo de carga útil, consulte [RequestPayloadFile](#).

**Note**

Em vez de especificar um arquivo de carga útil, você pode adicionar o código JSON da carga útil diretamente à ação usando a propriedade `RequestPayload`. Para obter mais informações, consulte [RequestPayload](#).

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Obrigatório se [RequestPayloadFile](#) for fornecido)

Se você quiser transmitir um arquivo JSON de carga útil de solicitação para a ação Invocação do AWS Lambda, e esse arquivo de carga estiver contido em um [artefato de saída](#) de uma ação anterior, especifique esse artefato aqui.

Para ter mais informações sobre o arquivo de carga útil, consulte [RequestPayloadFile](#).

 Note

Em vez de especificar um arquivo de carga útil, você pode adicionar o código JSON da carga útil diretamente à ação usando a propriedade `RequestPayload`. Para obter mais informações, consulte [RequestPayload](#).

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

## Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Optional)

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Variáveis - opcional

## Environment

(*LambdaInvoke*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação. A ação se conecta à Conta da AWS Amazon VPC opcional especificada no ambiente escolhido. A ação usa a função padrão do IAM

especificada no ambiente para se conectar ao e usa a Conta da AWS função do IAM especificada na [conexão da Amazon VPC](#) para se conectar à Amazon VPC.

#### Note

Se o perfil do IAM padrão não tiver as permissões exigidas pela ação, você poderá configurar a ação para usar um perfil diferente. Para obter mais informações, consulte [Alteração do perfil do IAM de uma ação](#).

Para ter mais informações sobre ambientes, consulte [Implantação em e Contas da AWS VPCs](#) e [Criar um ambiente](#).

Interface de usuário correspondente: guia Configuração/Ambiente

### Name

(*LambdaInvoke*/Environment/Name)

(Obrigatório se [Environment](#) for incluído)

Especifique o nome de um ambiente existente que deseja associar à ação.

Interface de usuário correspondente: guia Configuração/Ambiente

### Connections

(*LambdaInvoke*/Environment/Connections)

(Opcional nas versões mais recentes da ação; obrigatório nas versões mais antigas)

Especifique a conexão da conta a ser associada à ação. É possível especificar no máximo uma conexão de conta em Environment.

Se você não especificar uma conexão de conta:

- A ação usa a Conta da AWS conexão e a função padrão do IAM especificadas no ambiente no CodeCatalyst console. Para ter informações sobre como adicionar uma conexão de conta e um perfil do IAM padrão ao ambiente, consulte [Criar um ambiente](#).
- O perfil do IAM padrão deve incluir as políticas e as permissões exigidas pela ação. Para determinar quais são essas políticas e permissões, consulte a descrição da propriedade Perfil na documentação de definição de YAML da ação.

Para ter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para ter informações sobre como adicionar uma conexão de conta a um ambiente, consulte [Criar um ambiente](#).

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

## Name

(*LambdaInvoke*/Environment/Connections/Name)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da conexão da conta.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração tab/Environment/What está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' Environment/account/role AWS '/' conexão de conta

## Role

(*LambdaInvoke*/Environment/Connections/Role)

(Obrigatório se [Connections](#) for incluído)

Especifique o nome da função do IAM que a ação de AWS Lambda invocação usa para acessar AWS e invocar sua função Lambda. Certifique-se de ter [adicionado a função ao seu CodeCatalyst espaço](#) e de que a função inclua as seguintes políticas.

Se você não especificar uma função do IAM, a ação usará a função padrão do IAM listada no [ambiente](#) no CodeCatalyst console. Se você usar o perfil padrão no ambiente, verifique se ele tem as políticas a seguir.

- A política de permissões a seguir:

 Warning

Limite as permissões às exibidas na política a seguir. Usar um perfil com permissões mais amplas pode representar um risco de segurança.

- A política de confiança personalizada a seguir:

 Note

Você pode usar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` com essa ação, se desejar. Para obter mais informações sobre essa função, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*spaceName\* para a conta e o espaço](#). Entenda que o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` tem permissões de acesso completas, o que pode representar um risco de segurança. Recomendamos que você use esse perfil apenas em tutoriais e em cenários em que a segurança seja menos preocupante.

Interface de usuário correspondente: uma das seguintes, dependendo da versão da ação:

- (Versões mais recentes) A configuração `tab/Environment/What` está pronta? *my-environment* / menu de três pontos/ Mudar função
- (Versões mais antigas) Guia de configuração/' / Função Environment/account/role

## Configuration

(*LambdaInvoke*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## Function

(*LambdaInvoke*/Configuration/Function)

## (Obrigatório)

Especifique a AWS Lambda função que essa ação invocará. Você pode especificar o nome da função ou o nome do recurso da Amazon (ARN). É possível encontrar o nome ou ARN no console do Lambda.

### Note

A AWS conta na qual a função Lambda reside pode ser diferente da conta especificada em `Connections`:

Interface de usuário correspondente: guia Configuração/Função

## AWSRegion

(*LambdaInvoke*/Configuration/AWSRegion)

## (Obrigatório)

Especifique a AWS região em que sua AWS Lambda função reside. Para ver uma lista de códigos de região, consulte [Endpoints regionais](#) na Referência geral da AWS.

Interface de usuário correspondente: guia Configuração/Bucket de destino - opcional

## RequestPayload

(*LambdaInvoke*/Configuration/RequestPayload)

## (Optional)

Se você quiser transmitir uma carga útil de solicitação para a ação Invocação do AWS Lambda , especifique a carga útil da solicitação aqui, no formato JSON.

Exemplo da carga útil de solicitação:

```
'{ "key": "value" }'
```

Se você não quiser transmitir uma carga útil de solicitação para sua função do Lambda, omita essa propriedade.



 Note

Você pode especificar `RequestPayload` ou `RequestPayloadFile`, mas não os dois.

Para ter mais informações sobre a carga útil da solicitação, consulte o tópico [Invocar](#) na Referência da API do AWS Lambda .

Interface de usuário correspondente: guia Configuração/Carga útil de solicitação - opcional

## RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Se você quiser transmitir uma carga útil de solicitação para a ação Invocação do AWS Lambda , especifique o caminho até essa carga útil aqui. O arquivo estar no formato JSON.

O arquivo de carga útil da solicitação pode residir em um repositório de origem ou em um artefato de uma ação anterior. O caminho do arquivo é relativo ao repositório de origem ou à raiz do artefato.

Se você não quiser transmitir uma carga útil de solicitação para sua função do Lambda, omita essa propriedade.

 Note

Você pode especificar `RequestPayload` ou `RequestPayloadFile`, mas não os dois.

Para ter mais informações sobre o arquivo de carga útil da solicitação, consulte o tópico [Invocar](#) na Referência da API do AWS Lambda .

Interface de usuário correspondente: guia Configuração/Arquivo de carga útil de solicitação - opcional

## ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Especifique se você deseja marcar a ação Invocação do AWS Lambda como bem-sucedida, mesmo que a função invocada do AWS Lambda falhe. Considere definir essa propriedade como `true` para permitir que ações subsequentes em seu fluxo de trabalho sejam iniciadas apesar da falha do Lambda.

O padrão é falhar na ação se a função do Lambda falhar (“desativada” no editor visual ou `false` no editor YAML).

Interface de usuário correspondente: guia Configuração/Continuar com erro

## LogType

(*LambdaInvoke*/Configuration/LogType)

(Optional)

Especifique se você deseja incluir logs de erros na resposta da função do Lambda depois que ela for invocada. Você pode ver esses registros na guia Logs da ação de invocação do Lambda no console. CodeCatalyst Os valores possíveis são:

- `Tail` - logs de retorno
- `None` - logs não de retorno

O padrão é `Tail`.

Para ter mais informações sobre o tipo de log, consulte o tópico [Invocar](#) na Referência da API do AWS Lambda .

Para obter mais informações sobre como visualizar logs, consulte [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Interface de usuário correspondente: guia Configuração/Tipo de log

## ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Optional)

Especifique quais chaves na carga útil de resposta do Lambda você deseja converter em variáveis de saída. Em seguida, você pode referenciar as variáveis de saída em ações subsequentes no fluxo de trabalho. Para obter mais informações sobre variáveis em CodeCatalyst, consulte [Uso de variáveis em fluxos de trabalho](#).

Por exemplo, se a carga útil de resposta for semelhante a esta:

```
responsePayload = {
 "name": "Saanvi",
 "location": "Seattle",
 "department": {
 "company": "Amazon",
 "team": "AWS"
 }
}
```

... e os filtros de resposta forem semelhantes a estes:

```
Configuration:
 ...
 ResponseFilters: '{"name": ".name", "company": ".department.company"}'
```

... a ação gera as seguintes variáveis de saída:

| Chave   | Valor  |
|---------|--------|
| name    | Saanvi |
| company | Amazon |

Depois, você pode referenciar as variáveis name e company em ações subsequentes.

Se você não especificar nenhuma chave em `ResponseFilters`, a ação converterá cada chave de nível superior na resposta do Lambda em uma variável de saída. Para obter mais informações, consulte [Variáveis de “Invocação do AWS Lambda”](#).

Considere usar filtros de resposta para limitar as variáveis de saída geradas somente às que você realmente deseja usar.

Interface de usuário correspondente: guia Configuração/Filtros de resposta - opcional

## Outputs

(*LambdaInvoke*/Outputs)

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

## Artifacts

(*LambdaInvoke*/Outputs/Artifacts)

(Optional)

Especifique os artefatos gerados pela ação. Você pode referenciar esses artefatos como entrada em outras ações.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Saídas/Artefatos/Nome do artefato de criação

## Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Optional)

Especifique o nome do artefato que conterá a carga útil de resposta do Lambda que é retornada pela função do Lambda. O valor padrão é `lambda_artifacts`. Se você não especificar um artefato, a carga útil da resposta do Lambda poderá ser visualizada nos registros da ação, que estão disponíveis na guia Registros da ação no console. CodeCatalyst Para obter mais informações sobre como visualizar logs, consulte [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Interface de usuário correspondente: guia Saídas/Artefatos/Nome do artefato de criação

## Files

(*LambdaInvoke*/Outputs/Artifacts/Files)

(Optional)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `lambda-response.json` para que o arquivo de carga útil de resposta do Lambda seja incluído.

Interface de usuário correspondente: guia Saídas/Artefatos/Arquivos produzidos pela criação

## Modificação de uma definição de tarefa do Amazon ECS

Esta seção descreve como atualizar o `image` campo em um arquivo de [definição de tarefa do Amazon Elastic Container Service \(Amazon ECS\) usando](#) um fluxo de trabalho. CodeCatalyst Para fazer isso, você deve adicionar a ação Renderizar definição de tarefa do Amazon ECS ao seu fluxo de trabalho. Essa ação atualiza o campo de imagem no arquivo de definição de tarefa com um nome de imagem do Docker fornecido pelo seu fluxo de trabalho no runtime.

### Note

Você também pode usar esta ação para atualizar o campo `environment` da definição de tarefa com variáveis de ambiente.

### Tópicos

- [Quando usar essa ação](#)
- [Como a ação “Renderizar definição de tarefa do Amazon ECS” funciona](#)
- [Imagem de runtime usada pela ação “Renderizar definição de tarefa do Amazon ECS”](#)
- [Exemplo: modificar uma definição de tarefa do Amazon ECS](#)
- [Adicionar a ação “Renderizar definição de tarefa do Amazon ECS”](#)
- [Visualização do arquivo de definição de tarefa atualizado](#)
- [Variáveis de “Renderizar definição de tarefa do Amazon ECS”](#)
- [Ação YAML “Renderizar definição de tarefa do Amazon ECS”](#)

## Quando usar essa ação

Use essa ação se você tiver um fluxo de trabalho que cria e marca uma imagem do Docker com conteúdo dinâmico, como um ID de confirmação ou carimbo de data/hora.

Não use essa ação se o arquivo de definição de tarefa tiver um valor de imagem que permanece sempre o mesmo. Nesse caso, você pode inserir manualmente o nome da sua imagem no arquivo de definição de tarefa.

## Como a ação “Renderizar definição de tarefa do Amazon ECS” funciona

Você deve usar a ação Renderizar definição de tarefa do Amazon ECS com as ações Compilar e Implantar no Amazon ECS em seu fluxo de trabalho. Juntas, essas ações funcionam da seguinte forma:

1. A ação de criação cria sua imagem do Docker e a marca com um nome, um ID de confirmação, um carimbo de data/hora ou outro conteúdo dinâmico. Por exemplo, sua ação de criação pode ser semelhante a esta:

```
MyECSWorkflow
 Actions:
 BuildAction:
 Identifier: aws/build@v1
 ...
 Configuration:
 Steps:
 # Build, tag, and push the Docker image...
 - Run: docker build -t MyDockerImage:${WorkflowSource.CommitId} .
 ...
```

No código anterior, a diretiva `docker build -t` indica compilar a imagem do Docker e marcá-la com o ID de confirmação no runtime da ação. O nome da imagem gerada pode ser semelhante a este:

`MyDockerImage:a37bd7e`

2. A ação Renderizar definição de tarefa do Amazon ECS adiciona o nome da imagem gerada dinamicamente, `MyDockerImage:a37bd7e`, ao seu arquivo de definição de tarefa, da seguinte forma:

```
{
 "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
execution-role",
 "containerDefinitions": [
 {
 "name": "codecatalyst-ecs-container",
```

```

 "image": MyDockerImage:a37bd7e,
 "essential": true,
 ...
 "portMappings": [
 {
 "hostPort": 80,
 "protocol": "tcp",
 "containerPort": 80
 }
]
 },
],
 ...
}

```

Opcionalmente, você também pode fazer com que a ação Renderizar definição de tarefa do Amazon ECS adicione variáveis de ambiente à definição de tarefa, assim:

```

{
 "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-
role",
 "containerDefinitions": [
 {
 "name": "codecatalyst-ecs-container",
 "image": MyDockerImage:a37bd7e,
 ...
 "environment": [
 {
 "name": "ECS_LOGLEVEL",
 "value": "info"
 }
]
 }
],
 ...
}

```

Para ter mais informações sobre as variáveis de ambiente, consulte [Especificar variáveis de ambiente](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

3. A ação Implantar no Amazon ECS registra o arquivo de definição de tarefa atualizado no Amazon ECS. O registro do arquivo de definição de tarefa atualizado implanta a nova imagem, `MyDockerImage:a37bd7e` no Amazon ECS.

## Imagem de runtime usada pela ação “Renderizar definição de tarefa do Amazon ECS”

A ação Renderizar definição de tarefa do Amazon ECS é executada em uma [imagem de novembro de 2022](#). Para obter mais informações, consulte [Imagens ativas](#).

## Exemplo: modificar uma definição de tarefa do Amazon ECS

Veja a seguir um exemplo de um fluxo de trabalho completo que inclui a ação Renderizar definição de tarefa do Amazon ECS, junto com ações de compilação e implantação. O objetivo do fluxo de trabalho é compilar e implantar uma imagem do Docker em seu cluster do Amazon ECS. O fluxo de trabalho consiste nos seguintes blocos de compilação que são executados sequencialmente:

- Um gatilho: esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).
- Uma ação de criação (`BuildDocker`) – No gatilho, a ação compila a imagem do Docker usando o Dockerfile, a marca com um ID de confirmação e envia a imagem para o Amazon ECR. Para ter mais informações sobre a ação de criação, consulte [Criação com fluxos de trabalho](#).
- Uma ação Renderizar definição de tarefa do Amazon ECS (`RenderTaskDef`): na realização da ação de criação, essa ação atualiza um `taskdef.json` existente, localizado na raiz do repositório de origem, com um valor de campo `image` que inclui o ID de confirmação correto. O arquivo atualizado é salvo com um novo nome de arquivo (`task-definition-random-string.json`) e, depois, cria um artefato de saída que contém esse arquivo. A ação de renderização também gera uma variável chamada `task-definition` e a define como o nome do novo arquivo de definição de tarefa. O artefato e a variável serão usados na ação de implantação, que será a seguir.
- Uma ação Implantar no Amazon ECS (`DeployToECS`) – Ao concluir a ação Renderizar definição de tarefa do Amazon ECS, a ação Implantar no Amazon ECS procura o artefato de saída gerado pela ação de renderização (`TaskDefArtifact`), encontra o arquivo `task-definition-random-string.json` dentro dele e o registra no seu serviço do Amazon ECS. Em seguida, o serviço do Amazon ECS segue as instruções no arquivo `task-`



`definition-random-string.json` para executar as tarefas do Amazon ECS – e os contêineres de imagem do Docker associados – dentro do seu cluster do Amazon ECS.

Name: codecatalyst-ecs-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH

Branches:

- main

Actions:

BuildDocker:

Identifier: aws/build@v1

Environment:

Name: codecatalyst-ecs-environment

Connections:

- Name: codecatalyst-account-connection

- Role: codecatalyst-ecs-build-role

Inputs:

Variables:

- Name: REPOSITORY\_URI

- Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-

repo

- Name: IMAGE\_TAG

- Value: \${WorkflowSource.CommitId}

Configuration:

Steps:

#pre\_build:

- Run: echo Logging in to Amazon ECR...

- Run: aws --version

- Run: aws ecr get-login-password --region us-east-2 | docker login --username

AWS --password-stdin 111122223333.dkr.ecr.us-east-2.amazonaws.com

#build:

- Run: echo Build started on `date`

- Run: echo Building the Docker image...

- Run: docker build -t \$REPOSITORY\_URI:latest .

- Run: docker tag \$REPOSITORY\_URI:latest \$REPOSITORY\_URI:\$IMAGE\_TAG

#post\_build:

- Run: echo Build completed on `date`

- Run: echo Pushing the Docker images...

- Run: docker push \$REPOSITORY\_URI:latest

- Run: docker push \$REPOSITORY\_URI:\$IMAGE\_TAG

```
RenderTaskDef:
 DependsOn:
 - BuildDocker
 Identifier: aws/ecs-render-task-definition@v1
 Inputs:
 Variables:
 - Name: REPOSITORY_URI
 Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
 - Name: IMAGE_TAG
 Value: ${WorkflowSource.CommitId}
 Configuration:
 task-definition: taskdef.json
 container-definition-name: codecatalyst-ecs-container
 image: $REPOSITORY_URI:$IMAGE_TAG
 # The output artifact contains the updated task definition file.
 # The new file is prefixed with 'task-definition'.
 # The output variable is set to the name of the updated task definition file.
 Outputs:
 Artifacts:
 - Name: TaskDefArtifact
 Files:
 - "task-definition*"
 Variables:
 - task-definition

DeployToECS:
 Identifier: aws/ecs-deploy@v1
 Environment:
 Name: codecatalyst-ecs-environment
 Connections:
 - Name: codecatalyst-account-connection
 Role: codecatalyst-ecs-deploy-role
 #Input artifact contains the updated task definition file.
 Inputs:
 Sources: []
 Artifacts:
 - TaskDefArtifact
 Configuration:
 region: us-east-2
 cluster: codecatalyst-ecs-cluster
 service: codecatalyst-ecs-service
```

```
task-definition: ${RenderTaskDef.task-definition}
```

## Adicionar a ação “Renderizar definição de tarefa do Amazon ECS”

Use as instruções a seguir para adicionar a ação Renderizar definição de tarefa do Amazon ECS ao seu fluxo de trabalho.

### Pré-requisito

Antes de começar, verifique se você tem um fluxo de trabalho que inclua uma ação de criação que gere dinamicamente uma imagem do Docker. Consulte o [fluxo de trabalho de exemplo](#) anterior para saber detalhes.

### Visual

Como adicionar a ação “Renderizar definição de tarefa do Amazon ECS” usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Renderizar definição de tarefa do Amazon ECS e faça o seguinte:
  - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Selecione a opção Renderizar definição de tarefa do Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
  - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).

- Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para ver uma descrição de cada campo, consulte [Ação YAML “Renderizar definição de tarefa do Amazon ECS”](#). Essa referência fornece informações detalhadas sobre cada campo (e o valor da propriedade YAML correspondente) conforme elas aparecem nos editores YAML e visual.
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para adicionar a ação “Renderizar definição de tarefa do Amazon ECS” usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  2. Selecione o projeto.
  3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  5. Escolha Editar.
  6. Selecione YAML.
  7. No canto superior esquerdo, selecione + Ações para abrir o catálogo de ações.
  8. Na lista suspensa, escolha Amazon. CodeCatalyst
  9. Pesquise a ação Renderizar definição de tarefa do Amazon ECS e faça o seguinte:
    - Selecione o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Selecione a opção Renderizar definição de tarefa do Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
    - (Opcional) Selecione Visualizar origem para [visualizar o código-fonte da ação](#).

- Selecione Adicionar ao fluxo de trabalho para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Modifique as propriedades no código YAML de acordo com as suas necessidades. Uma explicação de cada propriedade disponível é fornecida em [Ação YAML “Renderizar definição de tarefa do Amazon ECS”](#).
  11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
  12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Próximas etapas

Depois de adicionar a ação de renderização, adicione a ação Implantar no Amazon ECS ao seu fluxo de trabalho seguindo as instruções em [Implantação no Amazon ECS com um fluxo de trabalho](#). Ao adicionar a ação de implantação, faça o seguinte:

1. Na guia Entradas da ação de implantação, em Artefatos - opcional, selecione o artefato que foi gerado pela ação de renderização. Ele contém o arquivo de definição de tarefa atualizado.

Para obter mais informações sobre artefatos, consulte [Compartilhar artefatos e arquivos entre ações](#).

2. Na guia Configuração da ação de implantação, no campo Definição da tarefa, especifique a seguinte variável de ação: `${action-name.task-definition}` onde *action-name* está o nome da sua ação de renderização, por exemplo, `RenderTaskDef`. A ação de renderização define essa variável com o novo nome do arquivo de definição da tarefa.

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

Para ter mais informações sobre como configurar a ação de implantação, consulte o [exemplo de fluxo de trabalho](#) anterior.

## Visualização do arquivo de definição de tarefa atualizado

Você pode visualizar o nome e o conteúdo do arquivo de definição de tarefa atualizado.

Para visualizar o nome do arquivo de definição de tarefa atualizado, depois que a ação Renderizar definição de tarefa do Amazon ECS o tiver processado.

1. Encontre a execução que inclui uma ação de renderização concluída:
  - a. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  - b. Selecione o projeto.
  - c. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - d. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  - e. Escolha uma execução que inclua a ação de renderização concluída.
2. No diagrama do fluxo de trabalho, selecione a ação de renderização.
3. Selecione Saídas.
4. Selecione Variáveis.
5. O nome do arquivo de definição de tarefa é exibido. Ele é semelhante a `task-definition--259-0a2r7gx1TF5X-.json`.

Para visualizar o conteúdo do arquivo de definição de tarefa atualizado

1. Encontre a execução que inclui uma ação de renderização concluída:
  - a. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
  - b. Selecione o projeto.
  - c. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - d. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  - e. Escolha uma execução que inclua a ação de renderização concluída.
2. Na execução do fluxo de trabalho, na parte superior, ao lado de Visual e YAML, selecione Saídas do fluxo de trabalho.
3. Na seção Artefatos, selecione Baixar ao lado do artefato que contém o arquivo de definição de tarefa atualizado. Esse artefato terá uma coluna Produzido por definida com o nome da sua ação de renderização.

4. Abra o arquivo .zip para visualizar o arquivo .json de definição de tarefa.

## Variáveis de “Renderizar definição de tarefa do Amazon ECS”

A ação Renderizar definição de tarefa do Amazon ECS produz e define as seguintes variáveis em runtime. Elas são conhecidas como variáveis predefinidas.

Para ter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usar variáveis predefinidas](#).

| Chave           | Valor                                                                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| task-definition | <p>O nome dado ao arquivo de definição de tarefa que foi atualizado pela ação Renderizar definição de tarefa do Amazon ECS.</p> <p>O nome segue o formato <code>task-definition-<i>random-string</i>.json</code>.</p> <p>Exemplo: <code>task-definition--259-0a2r7gx1TF5Xr.json</code></p> |

## Ação YAML “Renderizar definição de tarefa do Amazon ECS”

Veja a seguir a definição YAML da ação Renderizar definição de tarefa do Amazon ECS. Para saber como usar essa ação, consulte [Modificação de uma definição de tarefa do Amazon ECS](#).

Essa definição de ação existe como uma seção dentro de um arquivo de definição de fluxo de trabalho mais amplo. Para obter mais informações sobre esse arquivo, consulte [Definição do YAML do fluxo de trabalho](#).

### Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

```

The workflow definition starts here.
See Propriedades de nível superior for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

The action definition starts here.
ECSRenderTaskDefinition_nn:
 Identifier: aws/ecs-render-task-definition@v1
 DependsOn:
 - build-action
 Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 Timeout: timeout-minutes
 Inputs:
 # Specify a source or an artifact, but not both.
 Sources:
 - source-name-1
 Artifacts:
 - task-definition-artifact
 Variables:
 - Name: variable-name-1
 Value: variable-value-1
 - Name: variable-name-2
 Value: variable-value-2
 Configuration
 task-definition: task-definition-path
 container-definition-name: container-definition-name
 image: docker-image-name
 environment-variables:
 - variable-name-1=variable-value-1
 - variable-name-2=variable-value-2
 Outputs:
 Artifacts:
 - Name: TaskDefArtifact
 Files: "task-definition*"
 Variables:
 - task-definition

```



## ECSRenderTaskDefinition

(Obrigatório)

Especifique o nome da ação. Todos os nomes de ação devem ser exclusivos no fluxo de trabalho. Os nomes de ação são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de ação.

Padrão: `ECSRenderTaskDefinition_nn`.

Interface de usuário correspondente: guia Configuração/Nome da ação

### Identifier

(*ECSRenderTaskDefinition*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para obter mais informações, consulte [Especificação da versão da ação a ser usada](#).

Padrão: `aws/ecs-render-task-definition@v1`.

UI correspondente: diagrama de fluxo de trabalho/ ECSRenderTaskDefinition \_nn/ aws/ @v1 label  
ecs-render-task-definition

### DependsOn

(*ECSRenderTaskDefinition*/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito para que essa ação seja executada.

Para ter mais informações sobre a funcionalidade “Depende de”, consulte [Sequenciar ações](#).

Interface de usuário correspondente: guia Entradas/Depende de - opcional

### Compute

(*ECSRenderTaskDefinition*/Compute)

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

## Type

(*ECSRenderTaskDefinition*/Compute/Type)

(Obrigatório se [Compute](#) for incluído)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

Interface de usuário correspondente: guia Configuração/Tipo de computação

## Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`.

Interface de usuário correspondente: guia Configuração/Frota de computação

## Timeout

(*ECSRenderTaskDefinition*/Timeout)

(Optional)

Especifique a quantidade de tempo em minutos (editor YAML) ou horas e minutos (editor visual) que a ação pode ser executada antes de CodeCatalyst finalizar a ação. O mínimo é de cinco minutos e o máximo está descrito em [Cotas para fluxos de trabalho em CodeCatalyst](#). O tempo limite padrão é igual ao tempo limite máximo.

Interface de usuário correspondente: guia Configuração/Tempo limite - opcional

## Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Optional)

A seção Inputs define os dados que ECSRenderTaskDefinition precisa durante a execução de um fluxo de trabalho.

### Note

Somente uma entrada (uma origem ou um artefato) é permitida por ação Renderizar definição de tarefa do Amazon ECS. As variáveis não são contadas nesse total.

Interface de usuário correspondente: guia Entradas

## Sources

(*ECSRenderTaskDefinition*/Inputs/Sources)

(Obrigatório se o arquivo de definição de tarefas estiver armazenado em um repositório de origem)

Se seu arquivo de definição de tarefa estiver armazenado em um repositório de origem, especifique o rótulo do repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seu arquivo de definição de tarefa não estiver em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectar repositórios de origem aos fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Origens - opcional

## Artifacts - input

(*ECSRenderTaskDefinition*/Inputs/Artifacts)

(Obrigatório se o arquivo de definição de tarefa estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se o arquivo de definição de tarefa que você deseja implantar estiver em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se seu arquivo de definição de tarefa não estiver em um artefato, ele deverá residir em seu repositório de origem.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Configuração/Artefatos - opcional

## Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Obrigatório)

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Entradas/Variáveis - opcional

## Configuration

(*ECSRenderTaskDefinition*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

Interface de usuário correspondente: guia Configuração

## task-definition

(*ECSRenderTaskDefinition*/Configuration/task-definition)

(Obrigatório)

Especifique o caminho para um arquivo de definição de tarefa existente. Se o arquivo residir em seu repositório de origem, o caminho é relativo à pasta raiz do repositório de origem. Se o arquivo residir em um artefato de uma ação anterior do fluxo de trabalho, o caminho é relativo à pasta raiz do artefato. Para ter mais informações sobre os arquivos de definição de tarefa, consulte [Definições de tarefa](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Definição de tarefa

## container-definition-name

(*ECSRenderTaskDefinition*/Configuration/container-definition-name)

(Obrigatório)

Especifique o nome do contêiner em que sua imagem do Docker será executada. Você pode encontrar esse nome em `containerDefinitions`, campo `name` em seu arquivo de definição de tarefa. Para ter mais informações, consulte [Nome](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Nome do contêiner

## image

(*ECSRenderTaskDefinition*/Configuration/image)

## (Obrigatório)

Especifique o nome da imagem do Docker que você deseja que a ação Renderizar definição de tarefa do Amazon ECS adicione ao seu arquivo de definição de tarefas. A ação adiciona esse nome a `containerDefinitions`, campo `image` em seu arquivo de definição de tarefa. Se um valor já existir no campo `image`, a ação o substituirá. Você pode incluir variáveis no nome da imagem.

Exemplos:

Se você especificar `MyDockerImage:${WorkflowSource.CommitId}`, a ação será adicionada `MyDockerImage:commit-id` ao arquivo de definição da tarefa, onde `commit-id` está uma ID de confirmação gerada em tempo de execução pelo fluxo de trabalho.

Se você especificar `my-ecr-repo/image-repo:${date +%m-%d-%y-%H-%m-%s}`, a ação adicionará `my-ecr-repo/image-repo:date +%m-%d-%y-%H-%m-%s` ao arquivo de definição da tarefa, onde `my-ecr-repo` é o URI de um Amazon Elastic Container Registry (ECR) e `date +%m-%d-%y-%H-%m-%s` é um timestamp no formato month-day-year-hour-minute-second gerado em tempo de execução pelo fluxo de trabalho.

Para ter mais informações sobre o campo `image`, consulte [Imagem](#) no Guia do desenvolvedor do Amazon Elastic Container Service. Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

Interface de usuário correspondente: guia Configuração/Nome da imagem

## environment-variables

(*ECSRenderTaskDefinition*/Configuration/environment-variables)

## (Obrigatório)

Especifique as variáveis de ambiente que você deseja que a ação Renderizar definição de tarefa do Amazon ECS adicione ao seu arquivo de definição de tarefa. A ação adiciona as variáveis a `containerDefinitions`, campo `environment` em seu arquivo de definição de tarefa. Se as variáveis já existirem no arquivo, a ação substituirá os valores das variáveis existentes e adicionará quaisquer novas variáveis. Para ter mais informações sobre as variáveis de ambiente do Amazon ECS, consulte [Especificar variáveis de ambiente](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Interface de usuário correspondente: guia Configuração/Variáveis de ambiente – opcional

## Outputs

(*ECSRenderTaskDefinition*/Outputs)

(Obrigatório)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

Interface de usuário correspondente: guia Saídas

## Artifacts

(*ECSRenderTaskDefinition*/Outputs/Artifacts)

(Obrigatório)

Especifique os artefatos gerados pela ação. Você pode referenciar esses artefatos como entrada em outras ações.

Para ter mais informações sobre artefatos, inclusive exemplos, consulte [Compartilhar artefatos e arquivos entre ações](#).

Interface de usuário correspondente: guia Saídas/Artefatos

## Name

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Name)

(Obrigatório)

Especifique o nome do artefato que conterá o arquivo de definição de tarefa atualizado. O valor padrão é `MyTaskDefinitionArtifact`. Em seguida, você deve especificar esse artefato como entrada na ação Implantar no Amazon ECS. Para entender como adicionar esse artefato como entrada para a ação Implantar no Amazon ECS, consulte [Exemplo: modificar uma definição de tarefa do Amazon ECS](#).

Interface de usuário correspondente: guia Saídas/Artefatos/Nome

## Files

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Files)

## (Obrigatório)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `task-definition-`\* para que o arquivo de definição de tarefa atualizado, que começa com `task-definition-`, seja incluído.

Interface de usuário correspondente: guia Saídas/Artefatos/Arquivos

## Variables

(*ECSRenderTaskDefinition*/Outputs/Variables)

## (Obrigatório)

Especifique o nome de uma variável a ser definida pela ação de renderização. A ação de renderização definirá o valor dessa variável como o nome do arquivo de definição de tarefa atualizado (por exemplo, `task-definition-random-string.json`). Em seguida, você deve especificar essa variável na propriedade Definição de tarefa (editor visual) ou `task-definition` (editor yaml) da ação Implantar no Amazon ECS. Para entender como adicionar essa variável na ação Implantar no Amazon ECS, consulte [Exemplo: modificar uma definição de tarefa do Amazon ECS](#).

Padrão: `task-definition`

Interface de usuário correspondente: guia Saídas/Variáveis/Nome

# Uso de variáveis em fluxos de trabalho

Uma variável é um par de valores-chave que contém informações que você pode consultar em seu fluxo de trabalho da Amazon CodeCatalyst. A parte do valor da variável é substituída por um valor real quando o fluxo de trabalho é executado.

Há dois tipos de variáveis que podem ser usadas no fluxo de trabalho:

- Variáveis definidas pelo usuário – Pares chave-valor que você define.
- Variáveis predefinidas – Pares de chave-valor que são emitidos automaticamente por um fluxo de trabalho. Você não precisa defini-las.

Para obter mais informações sobre fluxos de trabalho, consulte [Compilação, teste e implantação com fluxos de trabalho](#).



 Note

CodeCatalyst também suporta [parâmetros GitHub de saída](#), que se comportam como variáveis e podem ser referenciados em outras ações. Para ter mais informações, consulte [Exportação dos parâmetros de saída do GitHub](#) e [Referência aos parâmetros de saída do GitHub](#)

## Tópicos

- [Usar variáveis definidas pelo usuário](#)
- [Usar variáveis predefinidas](#)

## Usar variáveis definidas pelo usuário

As variáveis definidas pelo usuário são pares chave-valor que você define. Existem dois tipos:

- Variáveis de texto simples, ou simplesmente variáveis – Pares de chave-valor que você define em texto simples no arquivo de definição do fluxo de trabalho.
- Segredos — Esses são pares de valores-chave que você define em uma página separada de segredos do console da Amazon CodeCatalyst . A chave (nome) é um rótulo público e o valor contém as informações que você deseja manter privadas. Especifique apenas a chave no arquivo de definição do fluxo de trabalho. Use segredos no lugar de senhas e outras informações confidenciais no arquivo de definição do fluxo de trabalho.

 Note

Resumindo, este guia usa o termo variável para variável de texto simples.

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

## Tópicos

- [Exemplos de variáveis](#)
- [Definição de uma variável](#)
- [Definição de um segredo](#)

- [Exportação de uma variável para que outras ações possam usá-la](#)
- [Referência a uma variável na ação que a define](#)
- [Referência a uma saída de variável por outra ação](#)
- [Referência a um segredo](#)

## Exemplos de variáveis

Os exemplos a seguir mostram como definir e referenciar variáveis no arquivo de definição de fluxo de trabalho.

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

### Exemplos

- [Exemplo: definição de uma variável usando a propriedade de entradas](#)
- [Exemplo: definição de uma variável usando a propriedade de etapas](#)
- [Exemplo: exportação de uma variável usando a propriedade de saídas](#)
- [Exemplo: referência de uma variável definida na mesma ação](#)
- [Exemplo: referência de uma variável definida em outra ação](#)
- [Exemplo: referenciar um segredo](#)

### Exemplo: definição de uma variável usando a propriedade de entradas

O exemplo a seguir mostra como definir duas variáveis, VAR1 e VAR2, em uma seção Inputs.

```
Actions:
 Build:
 Identifier: aws/build@v1
 Inputs:
 Variables:
 - Name: VAR1
 Value: "My variable 1"
 - Name: VAR2
 Value: "My variable 2"
```

### Exemplo: definição de uma variável usando a propriedade de etapas

O exemplo a seguir mostra como definir explicitamente uma variável DATE na seção Steps.

```
Actions:
 Build:
 Identifier: aws/build@v1
 Configuration:
 Steps:
 - Run: DATE=$(date +%m-%d-%y)
```

Exemplo: exportação de uma variável usando a propriedade de saídas

O exemplo a seguir mostra como definir duas variáveis, `REPOSITORY-URI` e `TIMESTAMP`, e exportá-las usando a seção `Outputs`.

```
Actions:
 Build:
 Identifier: aws/build@v1
 Inputs:
 Variables:
 - Name: REPOSITORY-URI
 Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
 Configuration:
 Steps:
 - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
 Outputs:
 Variables:
 - REPOSITORY-URI
 - TIMESTAMP
```

Exemplo: referência de uma variável definida na mesma ação

O exemplo a seguir mostra como especificar uma variável `VAR1` em `MyBuildAction` e, depois, referenciá-la na mesma ação usando `$VAR1`.

```
Actions:
 MyBuildAction:
 Identifier: aws/build@v1
 Inputs:
 Variables:
 - Name: VAR1
 Value: my-value
 Configuration:
```

**Steps:**

- Run: \$VAR1

Exemplo: referência de uma variável definida em outra ação

O exemplo a seguir mostra como especificar uma variável `TIMESTAMP` em `BuildActionA`, exportá-la usando a propriedade `Outputs` e, depois, referenciá-la em `BuildActionB` usando `${BuildActionA.TIMESTAMP}`.

**Actions:****BuildActionA:**

Identifier: aws/build@v1

**Configuration:****Steps:**

- Run: `TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)`

**Outputs:****Variables:**

- `TIMESTAMP`

**BuildActionB:**

Identifier: aws/build@v1

**Configuration:****Steps:**

- Run: `docker build -t my-ecr-repo/image-repo:latest .`
- Run: `docker tag my-ecr-repo/image-repo:${BuildActionA.TIMESTAMP}`

```
Specifying just '$TIMESTAMP' here will not work
because TIMESTAMP is not a variable
in the BuildActionB action.
```

Exemplo: referenciar um segredo

Os exemplos a seguir mostram como referenciar um segredo de `my-password`. `my-password` é a chave do segredo. A chave secreta e o valor da senha correspondente devem ser especificados na página Segredos do CodeCatalyst console antes de serem usados no arquivo de definição do fluxo de trabalho. Para obter mais informações, consulte [Mascarar dados usando segredos](#).

**Actions:****BuildActionA:**

Identifier: aws/build@v1

**Configuration:****Steps:**

```
- Run: curl -u LiJuan:${Secrets.my-password} https://example.com
```

## Definição de uma variável

É possível definir variáveis de duas maneiras:

- Na seção Inputs de uma ação de fluxo de trabalho. Consulte [Para definir uma variável na seção “Entradas”](#)
- Na seção Steps de uma ação de fluxo de trabalho. Consulte [Para definir uma variável na seção “Etapas”](#)

### Note

O Steps método só funciona com as ações de CodeCatalyst compilação, teste e GitHub ações, porque essas são as únicas ações que incluem uma Steps seção.

Para obter exemplos, consulte [Exemplos de variáveis](#).

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

## Visual

Para definir uma variável na seção “Entradas” (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação onde deseja configurar a variável.
8. Selecione Entradas.
9. Em Variáveis – opcional, selecione Adicionar variável e, depois, faça o seguinte:

Especifique uma sequência de name/value pares que defina as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes de variável são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos espaços. Não é possível usar aspas para habilitar caracteres especiais e espaços nos nomes de variável.

Para ter mais informações sobre variáveis, inclusive exemplos, consulte [Uso de variáveis em fluxos de trabalho](#).

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para definir uma variável na seção “Entradas” (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação de fluxo de trabalho, adicione um código semelhante ao seguinte:

```
action-name:
 Inputs:
 Variables:
 - Name: variable-name
 Value: variable-value
```

Para obter mais exemplos, consulte [Exemplos de variáveis](#). Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Visual

Para definir uma variável na seção “Etapas” (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione a ação onde deseja configurar a variável.
8. Escolher configuração.
9. Nos comandos do Shell ou no GitHubActions YAML, o que estiver disponível, defina uma variável na açãoSteps, explícita ou implicitamente.
  - Para definir a variável explicitamente, inclua-a em um comando bash diretamente na seção Steps.
  - Para definir uma variável implicitamente, especifique-a em um arquivo referenciado na seção Steps da ação.

Para obter exemplos, consulte [Exemplos de variáveis](#). Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para a ação.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para definir uma variável na seção “Etapas” (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Em uma ação de fluxo de trabalho, defina uma variável na seção Steps da ação, explícita ou implicitamente.
  - Para definir a variável explicitamente, inclua-a em um comando bash diretamente na seção Steps.
  - Para definir uma variável implicitamente, especifique-a em um arquivo referenciado na seção Steps da ação.

Para obter exemplos, consulte [Exemplos de variáveis](#). Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para a ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Definição de um segredo

Você define um segredo na página Segredos do CodeCatalyst console. Para obter mais informações, consulte [Mascarar dados usando segredos](#).

Por exemplo, você pode definir um segredo parecido com este:

- Nome (chave): **my-password**
- Valor: **^\*H3#!b9**

Depois que o segredo for definido, você poderá especificar a chave do segredo (**my-password**) no arquivo de definição do fluxo de trabalho. Para obter um exemplo de como fazer isso, consulte [Exemplo: referenciar um segredo](#).



## Exportação de uma variável para que outras ações possam usá-la

Use as instruções a seguir para exportar uma variável de uma ação para que você possa referenciá-la em outras ações.

Antes de exportar uma variável, verifique o seguinte:

- Se você só precisar referenciar a variável dentro da ação em que ela está definida, não será necessário exportá-la.
- Nem todas as ações são compatíveis com a exportação de variáveis. Para determinar se a ação é compatível com esse recurso, siga as instruções do editor visual a seguir e veja se a ação inclui um botão Variáveis na guia Saídas. Se sim, a exportação de variáveis é compatível.
- Para exportar uma variável de uma GitHub Ação, consulte [Exportação dos parâmetros de saída do GitHub](#).

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

### Pré-requisito

Certifique-se de ter definido a variável que você deseja exportar. Para obter mais informações, consulte [Definição de uma variável](#).

### Visual

Para exportar uma variável (editor visual)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama de fluxo de trabalho, selecione a ação da qual deseja exportar a variável.
8. Selecione Saídas.
9. Em Variáveis – opcional, selecione Adicionar variável e, depois, faça o seguinte:

Especifique o nome de uma variável a ser exportada pela ação. Essa variável já deve estar definida na seção Inputs ou Steps da mesma ação.

10. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## YAML

Para exportar uma variável (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Na ação da qual você deseja exportar a variável, adicione um código semelhante ao seguinte:

```
action-name:
 Outputs:
 Variables:
 - Name: variable-name
```

Para obter mais exemplos, consulte [Exemplos de variáveis](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Referência a uma variável na ação que a define

Use as instruções a seguir para referenciar uma variável na ação que a define.

 Note

Para referenciar uma variável gerada por uma GitHub Ação, consulte [Referência aos parâmetros de saída do GitHub](#).

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

## Pré-requisito

Certifique-se de ter definido a variável que você deseja referenciar. Para obter mais informações, consulte [Definição de uma variável](#).

## Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

## YAML

Para referenciar uma variável na ação que a define

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.
7. Na CodeCatalyst ação que define a variável à qual você deseja se referir, adicione a variável usando a seguinte sintaxe bash:

```
$variable-name
```

Por exemplo:

```
MyAction:
 Configuration:
 Steps:
```

- Run: `$variable-name`

Para obter mais exemplos, consulte [Exemplos de variáveis](#). Para ter mais informações, consulte as informações de referência da ação em [Definição do YAML do fluxo de trabalho](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Referência a uma saída de variável por outra ação

Use as instruções a seguir para referenciar saídas de variáveis por outras ações.

### Note

Para referenciar uma saída variável de uma GitHub Ação, consulte [Referência aos parâmetros de saída do GitHub](#).

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

### Pré-requisito

Certifique-se de ter exportado a variável que você deseja referenciar. Para obter mais informações, consulte [Exportação de uma variável para que outras ações possam usá-la](#).

### Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

### YAML

Para referenciar uma saída de variável por outra ação (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

5. Escolha Editar.
6. Selecione YAML.
7. Na CodeCatalyst ação, adicione uma referência à variável usando a seguinte sintaxe:

```
${action-group-name.action-name.variable-name}
```

Substitua:

- *action-group-name* com o nome do grupo de ações que contém a ação que gera a variável.

 Note

Você pode omitir *action-group-name* se não houver um grupo de ações ou se a variável for produzida por uma ação no mesmo grupo de ações.

- *action-name* com o nome da ação que gera a variável.
- *variable-name* com o nome da variável.

Por exemplo:

```
MySecondAction:
 Configuration:
 Steps:
 - Run: ${MyFirstAction.TIMESTAMP}
```

Para obter mais exemplos, consulte [Exemplos de variáveis](#). Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Referência a um segredo

Para receber instruções sobre como fazer referência a um segredo no arquivo de definição do fluxo de trabalho, consulte [Usar um segredo](#).

Para ver um exemplo, consulte [Exemplo: referenciar um segredo](#).

## Usar variáveis predefinidas

Variáveis predefinidas são pares de chave-valor emitidos automaticamente por um fluxo de trabalho e disponibilizados para você usar em ações de fluxo de trabalho.

Para ter mais informações sobre variáveis, consulte [Uso de variáveis em fluxos de trabalho](#).

### Tópicos

- [Exemplos de referência de variáveis predefinidas](#)
- [Referência a uma variável predefinida](#)
- [Determinação de quais variáveis predefinidas seu fluxo de trabalho emite](#)
- [Lista de variáveis predefinidas](#)

## Exemplos de referência de variáveis predefinidas

Os exemplos a seguir mostram como fazer referência a variáveis predefinidas no arquivo de definição de fluxo de trabalho.

Para ter mais informações sobre variáveis predefinidas, consulte [Usar variáveis predefinidas](#).

### Exemplos

- [Exemplo: fazer referência à variável predefinida "CommitId"](#)
- [Exemplo: fazer referência à variável predefinida "BranchName"](#)

### Exemplo: fazer referência à variável predefinida "CommitId"

O exemplo a seguir mostra como fazer referência à variável predefinida `CommitId` na ação `MyBuildAction`. A variável `CommitId` é gerada automaticamente pelo CodeCatalyst. Para obter mais informações, consulte [Lista de variáveis predefinidas](#).

Embora o exemplo mostre a variável que está sendo usada na ação de criação, você pode usar `CommitId` em qualquer ação.

```
MyBuildAction:
```

```
Identifier: aws/build@v1
Inputs:
 Sources:
 - WorkflowSource
Configuration:
 Steps:
 #Build Docker image and tag it with a commit ID
 - Run: docker build -t image-repo/my-docker-image:latest .
 - Run: docker tag image-repo/my-docker-image:${WorkflowSource.CommitId}
```

Exemplo: fazer referência à variável predefinida “BranchName”

O exemplo a seguir mostra como fazer referência à variável predefinida BranchName na ação CDKDeploy. A variável BranchName é gerada automaticamente pelo CodeCatalyst. Para obter mais informações, consulte [Lista de variáveis predefinidas](#).

Embora o exemplo mostre a variável que está sendo usada na ação implantar AWS CDK, você pode usar BranchName em qualquer ação.

```
CDKDeploy:
 Identifier: aws/cdk-deploy@v2
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 StackName: app-stack-${WorkflowSource.BranchName}
```

## Referência a uma variável predefinida

Você pode referenciar variáveis predefinidas em qualquer ação dentro de um fluxo de trabalho do Amazon CodeCatalyst.

Use as instruções a seguir para referenciar uma variável predefinida em um fluxo de trabalho.

Para ter mais informações sobre variáveis predefinidas, consulte [Usar variáveis predefinidas](#).

### Pré-requisito

Determine o nome da variável predefinida que você deseja referenciar, como CommitId. Para obter mais informações, consulte [Determinação de quais variáveis predefinidas seu fluxo de trabalho emite](#).

## Visual

Não disponível. Escolha YAML para visualizar as instruções YAML.

## YAML

Como fazer referência a uma variável predefinida (editor YAML)

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Selecione a opção Editar.
6. Selecione YAML.
7. Em uma ação do CodeCatalyst, adicione a referência de variável predefinida usando a seguinte sintaxe:

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```

Substitua:

- *action-group-name* pelo nome do grupo de ação.

### Note

Você pode omitir *action-group-name* se não houver um grupo de ações ou se a variável for produzida por uma ação no mesmo grupo de ações.

- *action-name-or-WorkflowSource* por:

O nome da ação que gera a variável.

ou

WorkflowSource, se a variável for BranchName ou CommitId.

- *variable-name* pelo nome da variável.



Por exemplo:

```
MySecondAction:
 Configuration:
 Steps:
 - Run: echo ${MyFirstECSAction.cluster}
```

Outro exemplo:

```
MySecondAction:
 Configuration:
 Steps:
 - Run: echo ${WorkflowSource.CommitId}
```

Para obter mais exemplos de , consulte [Exemplos de referência de variáveis predefinidas](#).  
Para ter mais informações, consulte [Definição do YAML do fluxo de trabalho](#) para sua ação.

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Determinação de quais variáveis predefinidas seu fluxo de trabalho emite

Use o seguinte procedimento para determinar quais variáveis predefinidas um fluxo de trabalho emite quando executado. Em seguida, você pode referenciar essas variáveis no mesmo fluxo de trabalho.

Para ter mais informações sobre variáveis predefinidas, consulte [Usar variáveis predefinidas](#).

Como determinar as variáveis predefinidas que seu fluxo de trabalho emite

- Execute um destes procedimentos:
  - Execute o fluxo de trabalho uma vez. Após a conclusão da execução, as variáveis emitidas pelo fluxo de trabalho são exibidas na guia Variáveis da página de detalhes da execução. Para obter mais informações, consulte [Visualização do status e detalhes de execução do fluxo de trabalho](#).
  - Consulte [Lista de variáveis predefinidas](#). Essa referência lista o nome da variável (chave) e o valor de cada variável predefinida.

**Note**

O tamanho total máximo das variáveis de um fluxo de trabalho está listado em [Cotas para fluxos de trabalho em CodeCatalyst](#). Se o tamanho total exceder o máximo, a ação seguinte poderá falhar.

## Lista de variáveis predefinidas

Consulte as seções a seguir para visualizar as variáveis predefinidas produzidas automaticamente pelas ações do CodeCatalyst como parte da execução de um fluxo de trabalho.

Para ter mais informações sobre variáveis predefinidas, consulte [Usar variáveis predefinidas](#).

**Note**

Essa lista inclui somente variáveis predefinidas emitidas pela origem do CodeCatalyst e pelas [ações do CodeCatalyst](#). Se você estiver usando outros tipos de ações, como ações do GitHub ou do CodeCatalyst Labs, consulte [Determinação de quais variáveis predefinidas seu fluxo de trabalho emite](#).

## Lista

**Note**

Nem todas as ações do CodeCatalyst produzem variáveis predefinidas. Se a ação não estiver na lista, ela não produzirá variáveis.

- [variáveis BranchName " e CommitId "](#)
- [Variáveis de " CloudFormation pilha de implantação"](#)
- [Variáveis de "Implantar no Amazon ECS"](#)
- [Variáveis de "Implantar no cluster do Kubernetes"](#)
- [Variáveis de "Implantação do AWS CDK "](#)
- [variáveis de "Inicialização do AWS CDK "](#)

- [Variáveis de “Invocação do AWS Lambda ”](#)
- [Variáveis de “Renderizar definição de tarefa do Amazon ECS”](#)

## Mascarar dados usando segredos

Pode haver momentos em que você precise usar dados confidenciais, como credenciais de autenticação, em seus fluxos de trabalho. Armazenar esses valores em texto simples em qualquer lugar do seu repositório deve ser evitado, pois qualquer pessoa com acesso ao repositório que contém o segredo pode vê-los. Da mesma forma, esses valores não devem ser usados diretamente em nenhuma definição de fluxo de trabalho porque estarão visíveis como arquivos no seu repositório. Com CodeCatalyst, você pode proteger esses valores adicionando um segredo ao seu projeto e, em seguida, referenciando o segredo no seu arquivo de definição de fluxo de trabalho. Observe que você pode ter, no máximo, cinco segredos por ação.

### Note

Os segredos só podem ser usados para substituir senhas e informações confidenciais no arquivo de definição do fluxo de trabalho.

### Tópicos

- [Criar um segredo](#)
- [Editar um segredo](#)
- [Usar um segredo](#)
- [Excluir um segredo](#)

## Criar um segredo

Use o procedimento a seguir para criar um segredo. O segredo contém as informações confidenciais que você deseja ocultar da visualização.

### Note

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

## Como criar um segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Segredos.
3. Escolha Criar segredo.
4. Insira as seguintes informações:

Nome

Insira um nome para o segredo.

Valor

Insira o valor do segredo. Essas são as informações confidenciais que você deseja ocultar da visualização. Por padrão, o valor não é exibido. Para exibir o valor, selecione Mostrar valor.

Descrição

(Opcional) Insira uma descrição para o seu segredo.

5. Escolha Criar.

## Editar um segredo

Utilize o seguinte procedimento para editar um segredo.

### Como editar um segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Segredos.
3. Na lista de segredos, selecione o segredo que você quer editar.
4. Escolha Editar.
5. Edite as seguintes propriedades:

Valor

Insira o valor do segredo. Esse é o valor que você deseja ocultar da visualização. Por padrão, o valor não é exibido.

## Descrição

(Opcional) Insira uma descrição para o seu segredo.

6. Escolha Salvar.

## Usar um segredo

Para usar um segredo em uma ação de fluxo de trabalho, obtenha o identificador de referência do segredo e use-o na ação do fluxo de trabalho.

### Tópicos

- [Receber o identificador de um segredo](#)
- [Fazer referência a um segredo em um fluxo de trabalho](#)

## Receber o identificador de um segredo

Use o procedimento a seguir para obter o identificador de referência do segredo. Esse identificador será adicionado ao seu fluxo de trabalho.

Como receber o identificador de referência do segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Segredos.
3. Na lista de segredos, selecione o segredo que você deseja usar.
4. Na coluna ID de referência, copie o identificador do segredo. Veja a seguir a sintaxe do ID de referência:

```
${Secrets.<name>}
```

## Fazer referência a um segredo em um fluxo de trabalho

Use o procedimento a seguir para fazer referência a um segredo em um fluxo de trabalho.

Como fazer referência a um segredo

1. No painel de navegação, escolha CI/CD e Fluxos de trabalho.

2. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
3. Escolha Editar.
4. Selecione YAML.
5. Modifique o YAML para usar o identificador do segredo. Por exemplo, para usar um nome de usuário e uma senha armazenados como segredos com o comando `curl`, use um comando Run semelhante ao seguinte:

```
- Run: curl -u <username-secret-identifier>:<password-secret-identifier> https://example.com
```

6. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
7. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

## Excluir um segredo

Use o procedimento a seguir para excluir um segredo e o identificador de referência do segredo.

### Note

Antes de excluir um segredo, recomendamos que você remova o identificador de referência do segredo de todas as ações do fluxo de trabalho. Se você excluir o segredo sem excluir o identificador de referência, a ação falhará na próxima vez em que for executada.

Como excluir o identificador de referência de um segredo de um fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
3. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
4. Escolha Editar.
5. Selecione YAML.

6. Pesquise no fluxo de trabalho a seguinte string:

```
${Secrets.
```

Todos os identificadores de referência de todos os segredos serão encontrados.

7. Exclua o identificador de referência do segredo selecionado ou substitua-o por um valor de texto simples.
8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Para excluir um segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e Segredos.
3. Na lista de segredos, selecione o segredo que você quer excluir.
4. Escolha Excluir.
5. Insira **delete** para confirmar a exclusão.
6. Escolha Excluir.

## Visualização do status de um fluxo de trabalho

Talvez você queira ver o status de um fluxo de trabalho para ver se há algum problema de configuração do fluxo de trabalho que você precise resolver ou para solucionar problemas de execuções que não iniciam. CodeCatalyst avalia o status do fluxo de trabalho sempre que você cria ou atualiza o [arquivo de definição de fluxo de trabalho subjacente do fluxo de trabalho](#).

### Note

Você também pode visualizar o status de execução do fluxo de trabalho, que é diferente do status do fluxo de trabalho. Para obter mais informações, consulte [Visualização do status e detalhes de execução do fluxo de trabalho](#).

Para ver uma lista de possíveis estados de fluxo de trabalho, consulte [Estados do fluxo de trabalho em CodeCatalyst](#).

Como visualizar o status de um fluxo de trabalho

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.

O status é exibido com o fluxo de trabalho na lista.

5. (Opcional) Selecione o nome do fluxo de trabalho e encontre o campo Definição do fluxo de trabalho. Ele mostra o status do fluxo de trabalho.

## Cotas para fluxos de trabalho em CodeCatalyst

A tabela a seguir descreve cotas e limites para fluxos de trabalho na Amazon. CodeCatalyst

Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

|                                                                                         |        |
|-----------------------------------------------------------------------------------------|--------|
| Número máximo de fluxos de trabalho por espaço                                          | 800    |
| Tamanho máximo do arquivo de definição do fluxo de trabalho                             | 256 KB |
| Número máximo de arquivos de fluxo de trabalho processados em um único evento de origem | 50     |
| Número máximo de arquivos processados em um único evento de origem                      | 4.000  |
| Número máximo de frotas ativas por espaço                                               | 10     |



|                                                                                                |                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Número máximo de instâncias de computação ativas por frota                                     | 20                                                                                                                                                                                                                                       |
| Número máximo de artefatos de entrada por ação                                                 | 10                                                                                                                                                                                                                                       |
| Número máximo de artefatos de saída por ação                                                   | 10                                                                                                                                                                                                                                       |
| Tamanho total máximo das variáveis de saída de uma única ação                                  | 120 KB                                                                                                                                                                                                                                   |
| Comprimento máximo do valor de uma variável de saída                                           | 500 caracteres ou mais, dependendo da ação que emite o valor. <div> <b>Note</b><br/>Os valores podem ser truncados se excederem o limite da ação.</div> |
| Número máximo de dias para manter os artefatos gerados durante a execução de fluxo de trabalho | 30                                                                                                                                                                                                                                       |
| Número máximo de relatórios por ação                                                           | 50                                                                                                                                                                                                                                       |
| Número máximo de casos de teste por relatório de teste                                         | 20.000                                                                                                                                                                                                                                   |
| Número máximo de arquivos por relatório de cobertura de código                                 | 20.000                                                                                                                                                                                                                                   |
| Número máximo de descobertas da análise de composição de software por relatório                | 20.000                                                                                                                                                                                                                                   |
| Número máximo de arquivos por relatório de análise estática                                    | 20.000                                                                                                                                                                                                                                   |
| Número máximo de execuções simultâneas de fluxo de trabalho por espaço                         | 100                                                                                                                                                                                                                                      |

|                                                                     |                                                                                                                                      |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Número máximo de ações por fluxo de trabalho                        | 50                                                                                                                                   |
| Número máximo de ações em execução simultânea por fluxo de trabalho | 50                                                                                                                                   |
| Número máximo de ações em execução simultânea por espaço            | 200                                                                                                                                  |
| Quantidade máxima de tempo que uma ação pode ser executada          | <p>Para as ações de criação e teste, o tempo limite é de 8 horas.</p> <p>Para todas as outras ações, o tempo limite é de 1 hora.</p> |
| Número máximo de ambientes associados a uma Conta da AWS por espaço | 5.000                                                                                                                                |
| Número máximo de segredos por ação                                  | 5                                                                                                                                    |
| Número máximo de segredos por espaço                                | 500.000                                                                                                                              |

## Estados de execução do fluxo de trabalho

Uma execução de fluxo de trabalho pode estar em um dos seguintes estados:

- Bem-sucedido – A execução do fluxo de trabalho foi processada com êxito.
- Com falha – Uma ou mais ações na execução do fluxo de trabalho falharam.
- Em andamento – A execução do fluxo de trabalho está sendo processada.
- Parado – Uma pessoa parou a execução do fluxo de trabalho enquanto ele estava em andamento.
- Parando – A execução do fluxo de trabalho está sendo interrompida no momento.
- Cancelado — A execução do fluxo de trabalho foi cancelada CodeCatalyst porque o fluxo de trabalho associado foi excluído ou atualizado enquanto a execução estava em andamento.
- Substituído – Ocorre somente se você tiver configurado o [modo de execução de substituição](#). A execução do fluxo de trabalho foi cancelada CodeCatalyst porque uma execução posterior do fluxo de trabalho a substituiu.

# Estados do fluxo de trabalho em CodeCatalyst

Um fluxo de trabalho pode ter um dos seguintes estados:

- Válido – O fluxo de trabalho é executável e pode ser ativado por [gatilhos](#).

Para que um fluxo de trabalho seja marcado como válido, ambas as seguintes condições devem ser verdadeiras:

- O arquivo de definição do fluxo de trabalho do deve ser válido.
- O fluxo de trabalho não deve ter gatilhos, gatilhos de envio ou gatilho de extração executados usando os arquivos na ramificação atual. Para obter mais informações, consulte [Diretrizes para o uso de gatilhos e ramificações](#).
- Inválido – O arquivo de definição do fluxo de trabalho é inválido. O fluxo de trabalho não pode ser executado manual ou automaticamente por meio de gatilhos. Os fluxos de trabalho que não são válidos aparecem com uma definição de fluxo de trabalho com mensagem de **n** erros (ou similar) no CodeCatalyst console.

Para que um fluxo de trabalho seja marcado como inválido, as seguintes condições devem ser verdadeiras:

- O arquivo de definição do fluxo de trabalho deve estar configurado incorretamente.

Para corrigir um arquivo de definição de fluxo de trabalho configurado incorretamente, consulte [Como faço para corrigir os erros “A definição do fluxo de trabalho tem n erros”?](#).

- Inativo – A definição do fluxo de trabalho é válida, mas não pode ser executada manual ou automaticamente por meio de gatilhos.

Para que um fluxo de trabalho seja marcado como inativo, ambas as seguintes condições devem ser verdadeiras:

- O arquivo de definição do fluxo de trabalho do deve ser válido.
- O arquivo de definição do fluxo de trabalho deve incluir um gatilho de envio que especifique uma ramificação diferente daquela em que o arquivo de definição do fluxo de trabalho está atualmente. Para obter mais informações, consulte [Diretrizes para o uso de gatilhos e ramificações](#).

Para mudar um fluxo de trabalho de Inativo para Ativo, consulte [Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?](#).

**Note**

Se houver muitos fluxos de trabalho no estado Inativo, filtre-os para a visualização. Para filtrar fluxos de trabalho inativos, selecione o campo Filtrar fluxos de trabalho na parte superior da página Fluxos de trabalho, selecione Status, Status != Não igual e selecione INATIVO.

**Note**

Se o fluxo de trabalho especificar um recurso que você removerá posteriormente (por exemplo, um repositório de pacotes), CodeCatalyst não detectará essa alteração e continuará marcando o fluxo de trabalho como válido. Esses tipos de problemas serão detectados quando o fluxo de trabalho for executado.

## Definição do YAML do fluxo de trabalho

Veja a seguir a documentação de referência do arquivo de definição do fluxo de trabalho.

Arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve o fluxo de trabalho. Por padrão, o arquivo é armazenado em uma pasta `~/.codecatalyst/workflows/` na raiz do [repositório de origem](#). O arquivo pode ter uma extensão `.yaml` ou `.yml`, e a extensão deve estar em minúsculas.

Para criar e editar o arquivo de definição do fluxo de trabalho, você pode usar um editor como o vim ou usar o editor visual ou o editor YAML do CodeCatalyst console. Para obter mais informações, consulte [Usando os editores visual e YAML do CodeCatalyst console](#).

**Note**

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface, use Ctrl+F. O elemento será listado com a propriedade YAML associada.

## Tópicos

- [Exemplo de um arquivo de definição de fluxo de trabalho](#)
- [Diretrizes e convenções de sintaxe](#)
- [Propriedades de nível superior](#)

## Exemplo de um arquivo de definição de fluxo de trabalho

Veja a seguir um exemplo de um arquivo de definição de fluxo de trabalho simples. Inclui algumas propriedades de nível superior, uma seção Triggers e uma seção Actions com duas ações: Build e Test. Para obter mais informações, consulte [Sobre o arquivo de definição do fluxo de trabalho](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
 - Type: PUSH
 Branches:
 - main
Actions:
 Build:
 Identifier: aws/build@v1
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: docker build -t MyApp:latest .
 Test:
 Identifier: aws/managed-test@v1
 DependsOn:
 - Build
 Inputs:
 Sources:
 - WorkflowSource
 Configuration:
 Steps:
 - Run: npm install
 - Run: npm run test
```

## Diretrizes e convenções de sintaxe

Esta seção descreve as regras de sintaxe para o arquivo de definição do fluxo de trabalho, bem como as convenções de nomenclatura usadas nesta documentação de referência.

### Diretrizes de sintaxe do YAML

O arquivo de definição do fluxo de trabalho é escrito em YAML e segue a [especificação YAML 1.1](#), portanto, tudo o que é permitido nessa especificação também é permitido no YAML do fluxo de trabalho. Se você for novo no YAML, aqui estão algumas diretrizes rápidas para garantir o fornecimento de um código YAML válido.

- Diferenciação entre maiúsculas e minúsculas: o arquivo de definição do fluxo de trabalho diferencia maiúsculas de minúsculas, portanto, certifique-se de usar as maiúsculas e minúsculas mostradas nesta documentação.
- Caracteres especiais: recomendamos o uso de aspas ou aspas duplas em torno dos valores das propriedades que incluam qualquer um dos seguintes caracteres especiais: {, }, [, ], &, \*, #, ?, |, -, <, >, =, !, %, @, :, ` e ,

Se você não incluir as aspas, os caracteres especiais listados anteriormente poderão ser interpretados de forma inesperada.

- Nomes de propriedades: os nomes das propriedades (em oposição aos valores das propriedades) são limitados a caracteres alfanuméricos (a-z, 0-9), hífens (-) e sublinhados (\_). Não são permitidos espaços. Você não pode usar aspas ou aspas duplas para ativar caracteres especiais e espaços nos nomes das propriedades.

Não permitido:

```
'My#Build@action'
```

```
My#Build@action
```

```
My Build Action
```

Permitido:

```
My-Build-Action_1
```

- Códigos de escape: se o valor da sua propriedade incluir códigos de escape (por exemplo, \n ou \t), siga estas diretrizes:

- Use aspas simples para retornar o código de escape como uma string. Por exemplo, 'my string \n my string', retorna a string my string \n my string.
- Use aspas duplas para analisar o código de escape. Por exemplo, "my string \n my new line", retorna:

```
my string
my new line
```

- Comentários: prefacie os comentários com #.

Exemplo:

```
Name: MyWorkflow
This is a comment.
SchemaVersion: 1.0
```

- Triple dash (---): não use --- em seu código YAML. CodeCatalyst ignora tudo depois do---.

## Convenções de nomenclatura

Neste guia, usamos os termos propriedade e seção para nos referirmos aos itens principais em um arquivo de definição de fluxo de trabalho.

- Uma propriedade é qualquer item que inclua dois pontos (:). Por exemplo, no trecho de código a seguir, todas as seguintes são propriedades: Name, SchemaVersion, RunMode, Triggers, Type e Branches.
- Uma seção é qualquer propriedade que tenha subpropriedades. No trecho de código a seguir, há uma seção Triggers.

### Note

Neste guia, as “seções” às vezes são chamadas de “propriedades” e vice-versa, dependendo do contexto.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
```

```
Triggers:
- Type: PUSH
 Branches:
 - main
```

## Propriedades de nível superior

Veja a seguir a documentação de referência das propriedades de nível superior do arquivo de definição do fluxo de trabalho.

```
Name
Name: workflow-name

Schema version
SchemaVersion: 1.0

Run mode
RunMode: QUEUED/SUPERSEDED/PARALLEL

Compute
Compute:
...

Triggers
Triggers:
...

Actions
Actions:
...
```

### Name

(Obrigatório)

O nome do fluxo de trabalho. O nome do fluxo de trabalho é mostrado na lista de fluxos de trabalho e mencionado nas notificações e nos logs. O nome do fluxo de trabalho e o nome do arquivo de definição do fluxo de trabalho podem corresponder, ou você pode nomeá-los de forma diferente. Os nomes de fluxos de trabalho não precisam ser exclusivos. Os nomes de fluxos de trabalho são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hifens (-) e sublinhados (\_). Não são permitidos



espaços. Você não pode usar aspas para habilitar caracteres especiais e espaços nos nomes de fluxo de trabalho.

UI correspondente: editor/Workflow propriedades visuais/nome do fluxo de trabalho

## SchemaVersion

(Obrigatório)

A versão do esquema da definição do fluxo de trabalho. Atualmente, o único valor válido é 1.0.

Interface de usuário correspondente: nenhuma

## RunMode

(Optional)

Como CodeCatalyst lida com várias execuções. É possível usar um dos seguintes valores:

- QUEUED – Várias execuções são colocadas em fila e executadas uma após a outra. É possível ter até 50 execuções em uma fila.
- SUPERSEDED – Várias execuções são colocadas em fila e executadas uma após a outra. Uma fila só pode ter uma execução, portanto, se duas execuções terminarem juntas na mesma fila, a execução posterior substituirá a execução anterior e a execução anterior será cancelada.
- PARALLEL – Várias execuções ocorrem simultaneamente.

Se essa propriedade for omitida, o padrão será QUEUED.

Para obter mais informações, consulte [Configurar o comportamento de enfileiramento das execuções](#).

UI correspondente: editor/Workflow propriedades visuais/Avançado/Modo de execução

## Compute

(Optional)

O mecanismo de computação usado para executar as ações de fluxo de trabalho. É possível especificar a computação em nível de fluxo de trabalho ou em nível de ação, mas não em ambos. Quando especificada em nível de fluxo de trabalho, a configuração de computação se aplica a todas

as ações definidas no fluxo de trabalho. Em nível de fluxo de trabalho, também é possível realizar várias ações na mesma instância. Para obter mais informações, consulte [Compartilhamento de computação entre ações](#).

Para ter mais informações sobre computação, consulte [Configuração de imagens de computação e runtime](#).

Interface de usuário correspondente: nenhuma

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
 Type: EC2 | Lambda
 Fleet: fleet-name
 SharedInstance: true | false
```

Type

(Compute/Type)

(Necessário se Compute estiver definido)

O tipo do mecanismo de computação. É possível usar um dos seguintes valores:

- EC2(editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as execuções de ação.

- Lambda (editor visual) ou Lambda (editor YAML)

Velocidades otimizadas de inicialização da ação.

Para obter informações sobre tipos de dados, consulte [Tipos de computação](#).

UI correspondente: editor/Workflow propriedades visuais/Avançada/Tipo de computação

Fleet

(Compute/Fleet)

(Optional)

Especifique a máquina ou a frota que executará o fluxo de trabalho ou as ações de fluxo de trabalho. Com frotas sob demanda, quando uma ação é iniciada, o fluxo de trabalho provisiona os recursos necessários e as máquinas são destruídas quando a ação termina. Exemplos de frota sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Para ter mais informações sobre frotas sob demanda, consulte [Propriedades da frota sob demanda](#).

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para realizar as ações do fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para ter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#).

Se `Fleet` for omitido, o padrão será `Linux.x86-64.Large`.

Para ter mais informações sobre frotas de computação, consulte [Frotas de computação](#).

UI correspondente: editor/Workflow propriedades visuais/Avançada/Frota de computação

`SharedInstance`

(`Compute/SharedInstance`)

(Optional)

Especifique o recurso de compartilhamento de computação para suas ações. Com o compartilhamento de computação, as ações em um fluxo de trabalho são executadas na mesma instância (imagem do ambiente de runtime). É possível usar um dos seguintes valores:

- `TRUE` significa que a imagem do ambiente de runtime é compartilhada entre as ações do fluxo de trabalho.
- `FALSE` significa que uma imagem separada do ambiente de runtime é iniciada e usada para cada ação em um fluxo de trabalho, portanto, você não pode compartilhar recursos como artefatos e variáveis sem configuração adicional.

Para ter mais informações sobre compartilhamento de computação, consulte [Compartilhamento de computação entre ações](#).

Interface de usuário correspondente: nenhuma

Triggers

(Optional)

Uma sequência de um ou mais gatilhos para esse fluxo de trabalho. Se um gatilho não for especificado, inicie manualmente seu fluxo de trabalho.

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

UI correspondente: editor/workflow diagrama visual/acionadores

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Triggers:
- Type: PUSH
 Branches:
 - branch-name
 FilesChanged:
 - folder1/file
 - folder2/

- Type: PULLREQUEST
 Events:
 - OPEN
 - CLOSED
 - REVISION
 Branches:
 - branch-name
 FilesChanged:
 - file1.txt

- Type: SCHEDULE
 # Run the workflow at 10:15 am (UTC+0) every Saturday
 Expression: "15 10 ? * 7 *"
 Branches:
 - branch-name
```

Type

(Triggers/Type)

(Necessário se Triggers estiver definido)

Especifique o tipo de gatilho. É possível usar um dos seguintes valores:

- Envio (editor visual) ou PUSH (editor YAML)

Um gatilho de envio inicia a execução de um fluxo de trabalho quando uma alteração é enviada para seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação para a qual você está enviando (ou seja, a ramificação de destino).

- Solicitação pull (editor visual) ou PULLREQUEST (editor YAML)

Um gatilho de solicitação pull inicia a execução de um fluxo de trabalho quando uma solicitação pull é aberta, atualizada ou fechada no seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação da a qual você está extraindo (ou seja, a ramificação de origem).

- Programação (editor visual) ou SCHEDULE (editor YAML)

Um gatilho de programação inicia a execução do fluxo de trabalho em uma programação definida por uma expressão cron especificada por você. Uma execução de fluxo de trabalho separada será iniciada para cada ramificação em seu repositório de origem usando os arquivos da ramificação. (Para limitar as ramificações nas quais o gatilho é ativado, use o campo Ramificações (editor visual) ou a propriedade `Branches` (editor YAML).)

Ao configurar um gatilho de programação, siga estas instruções:

- Use apenas um gatilho de programação por fluxo de trabalho.
- Se você definiu vários fluxos de trabalho em seu CodeCatalyst espaço, recomendamos que não agende mais do que 10 deles para começarem simultaneamente.
- Certifique-se de configurar a expressão cron do gatilho com tempo adequado entre as execuções. Para obter mais informações, consulte [Expression](#).

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

UI correspondente: editor/workflow diagrama visual/acionadores/tipo de gatilho

## Events

(Triggers/Events)

(Obrigatório se o gatilho Type estiver definido como PULLREQUEST)

Especifique o tipo de eventos de solicitação pull que iniciarão a execução de um fluxo de trabalho. Os valores válidos são os seguintes:

- A solicitação pull é criada (editor visual) ou OPEN (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma solicitação pull é criada.

- A solicitação pull é fechada (editor visual) ou CLOSED (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma solicitação pull é fechada. O comportamento do evento CLOSED é complicado e é melhor compreendido por meio de um exemplo. Consulte [Exemplo: um gatilho com uma extração, ramificações e um evento “FECHADO”](#) para obter mais informações.

- Uma nova revisão é feita para solicitação pull (editor visual) ou REVISION (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma revisão da solicitação pull é criada. A primeira revisão é criada quando a solicitação pull é criada. Depois disso, uma nova revisão é criada sempre que alguém envia uma nova confirmação para a ramificação de origem especificada na solicitação pull. Se você incluir o evento REVISION em seu gatilho de solicitação pull, poderá omitir o evento OPEN, pois REVISION é um superconjunto de OPEN.

Você pode especificar vários eventos no mesmo gatilho de solicitação pull.

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

UI correspondente: editor/workflow diagrama visual/acionadores/eventos para pull request

Branches

(Triggers/Branches)

(Optional)

Especifique as ramificações em seu repositório de origem que o acionador monitora para saber quando iniciar a execução de um fluxo de trabalho. Você pode usar padrões regex para definir os nomes das ramificações. Por exemplo, use `main.*` para combinar todas as ramificações que começam com `main`.

As ramificações a serem especificadas são diferentes dependendo do tipo de gatilho:

- Para um gatilho de envio, especifique as ramificações para as quais você está enviando, ou seja, as ramificações de destino. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando os arquivos na ramificação correspondente.

Exemplos: `main.*`, `mainline`

- Para um gatilho de solicitação pull, especifique as ramificações para as quais você está enviando, ou seja, as ramificações de destino. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação de origem (não a ramificação correspondente).

Exemplos: `main.*`, `mainline`, `v1\-.*` (corresponde às ramificações que começam com `v1-`)

- Para um gatilho de programação, especifique as ramificações que contêm os arquivos que devem ser usados pela sua execução programada. Uma execução de fluxo de trabalho será iniciada por ramificação correspondente, usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação correspondente.

Exemplos: `main.*`, `version\-1\.*`

#### Note

Se você não especificar ramificações, o gatilho monitorará todas as ramificações em seu repositório de origem e iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e os arquivos de origem em:

- A ramificação para a qual você está enviando (para gatilhos de envio). Para obter mais informações, consulte [Exemplo: um simples gatilho de envio de código](#).
- A ramificação da qual você está extraíndo (para gatilhos de solicitação pull). Para obter mais informações, consulte [Exemplo: um simples gatilho de solicitação pull](#).
- Todas as ramificações (para gatilhos de programação). Uma execução de fluxo de trabalho será iniciada por ramificação em seu repositório de origem. Para obter mais informações, consulte [Exemplo: um gatilho de programação simples](#).

Para ter mais informações sobre ramificações e gatilhos, consulte [Diretrizes para o uso de gatilhos e ramificações](#).

Para obter mais exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

UI correspondente: `visualeditor/workflow diagram/Triggers/Branches`

`FilesChanged`

(Triggers/FilesChanged)

(Opcional se o gatilho Type estiver definido como PUSH, ou PULLREQUEST. Não compatível se o gatilho Type estiver definido como SCHEDULE.)

Especifique os arquivos ou pastas em seu repositório de origem que o acionador monitora para saber quando iniciar a execução de um fluxo de trabalho. Você pode usar expressões regulares para corresponder nomes ou caminhos de arquivos.

Para obter exemplos, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

UI correspondente: editor/workflow diagrama visual/acionadores/arquivos alterados

Expression

(Triggers/Expression)

(Obrigatório se o gatilho Type estiver definido como SCHEDULE)

Especifique a expressão cron que descreve quando você deseja que suas execuções de fluxo de trabalho programadas ocorram.

As expressões Cron CodeCatalyst usam a seguinte sintaxe de seis campos, em que cada campo é separado por um espaço:

*minutes hours days-of-month month days-of-week year*

Exemplos de expressões cron

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano | Significado                                                              |
|---------|-------|-------------|-----|----------------|-----|--------------------------------------------------------------------------|
| 0       | 0     | ?           | *   | SEG-SEX        | *   | Executada um fluxo de trabalho à meia-noite (UTC +0) de segunda a sexta. |
| 0       | 2     | *           | *   | ?              | *   | Executada um fluxo                                                       |



| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano | Significado                                                                                                                |
|---------|-------|-------------|-----|----------------|-----|----------------------------------------------------------------------------------------------------------------------------|
|         |       |             |     |                |     | de trabalho às 2h (UTC +0) todos os dias.                                                                                  |
| 15      | 22    | *           | *   | ?              | *   | Executada um fluxo de trabalho às 22h15 (UTC+0) todos os dias.                                                             |
| 0/30    | 22-2  | ?           | *   | SÁB-DOM        | *   | Executa um fluxo de trabalho a cada 30 minutos de sábado a domingo, entre 22h do dia inicial e 2h do dia seguinte (UTC+0). |

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano       | Significado                                                                                                 |
|---------|-------|-------------|-----|----------------|-----------|-------------------------------------------------------------------------------------------------------------|
| 45      | 13    | L           | *   | ?              | 2023-2027 | Executa um fluxo de trabalho às 13h45 (UTC+0) no último dia do mês entre os anos de 2023 e 2027, inclusive. |

Ao especificar expressões cron em CodeCatalyst, siga estas diretrizes:

- Especifique uma única expressão cron por gatilho SCHEDULE.
- Coloque a expressão cron entre aspas duplas (") no editor YAML.
- Especifique o horário em Tempo Universal Coordenado (UTC). Outros fusos horários não são suportados.
- Configure pelo menos 30 minutos entre as execuções. Não há suporte para uma cadência mais rápida.
- Especifique o *days-of-week* campo *days-of-month* ou, mas não ambos. Se você especificar um valor ou asterisco (\*) em um dos campos, deverá usar um ponto de interrogação (?) no outro. O asterisco significa "tudo" e o ponto de interrogação significa "qualquer".

Para obter mais exemplos de expressões cron e informações sobre curingas como ?, , e \*L, consulte a [referência de expressões Cron no Guia](#) do usuário da Amazon EventBridge . As expressões Cron CodeCatalyst funcionam exatamente da mesma maneira. EventBridge

Para ver exemplos de gatilhos de programação, consulte [Exemplos: gatilhos em fluxos de trabalho](#).

UI correspondente: `visualeditor/workflow diagram/Triggers/Schedule`

## Ações

Uma sequência de uma ou mais ações para esse fluxo de trabalho. CodeCatalyst suporta vários tipos de ação, como ações de criação e teste, que oferecem diferentes tipos de funcionalidade. Cada tipo de ação tem:

- uma propriedade `Identifier` que indica o ID exclusivo e de codificação rígida da ação. Por exemplo, `aws/build@v1` identifica a ação de criação.
- uma seção `Configuration` que contém propriedades específicas da ação.

Para ter mais informações sobre cada tipo de ação, consulte [Tipos de ação](#). O tópico [Tipos de ação](#) tem links para a documentação de cada ação.

Veja a seguir a referência YAML para ações e grupos de ações no arquivo de definição do fluxo de trabalho.

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Actions:
 action-or-gate-name:
 Identifier: identifier
 Configuration:
 ...
 #Action groups
 action-group-name:
 Actions:
 ...
```

`action-or-gate-name`

(Actions/*action-or-gate-name*)

(Obrigatório)

*action-name* Substitua por um nome que você deseja dar à ação. Os nomes das ações devem ser exclusivos no fluxo de trabalho e incluir apenas caracteres alfanuméricos, hifens e sublinhados. Para ter mais informações sobre as regras de sintaxe, consulte [Diretrizes de sintaxe do YAML](#).

Para ter mais informações sobre práticas de nomenclatura para ações, inclusive restrições, consulte [action-or-gate-name](#).

UI correspondente: editor visual//Guia de ***action-name*** configuração/ Nome da ação ou Nome de exibição da ação

action-group-name

(Actions/***action-group-name***)

(Optional)

Um grupo de ações contém uma ou mais ações. O agrupamento de ações em grupos ajuda a manter o fluxo de trabalho organizado e também permite configurar dependências entre grupos diferentes.

***action-group-name*** Substitua por um nome que você deseja dar ao grupo de ação. Os nomes de grupos das ações devem ser exclusivos no fluxo de trabalho e incluir apenas caracteres alfanuméricos, hifens e sublinhados. Para ter mais informações sobre as regras de sintaxe, consulte [Diretrizes de sintaxe do YAML](#).

Para ter mais informações sobre os grupos de ações, consulte [Agrupar ações em grupos de ações](#).

Interface de usuário correspondente: nenhuma

# Acompanhe e organize o trabalho com problemas no CodeCatalyst

Em CodeCatalyst, você pode monitorar recursos, bugs e qualquer outro trabalho envolvido em seu projeto. Cada parte do trabalho é mantida em um registro distinto chamado de problema. Você pode dividir um problema em objetivos menores adicionando uma lista de tarefas a ele. Cada problema pode ter uma descrição, um responsável, status e outras propriedades, que você pode pesquisar, agrupar e filtrar. Você pode visualizar os problemas usando as visualizações padrão ou criar suas próprias visualizações com filtragem, classificação ou agrupamento personalizados. Para ter mais informações sobre conceitos relacionados a problemas, consulte [Conceitos de problemas](#). Para saber como criar seu primeiro problema, consulte [Criando um problema no CodeCatalyst](#).

Confira um possível fluxo de trabalho para uma equipe que usa problemas:

Jorge Souza é um desenvolvedor trabalhando em um projeto. Ele e os colegas do projeto (Li Juan, Mateo Jackson e Wang Xiulan) colaboram para determinar qual trabalho precisa ser feito. Todos os dias, ele e os colegas desenvolvedores realizam uma reunião de sincronização, liderada por Wang Xiulan. Eles abrem o quadro navegando até a visualização do quadro de uma das equipes. Ao criar visualizações, usuários e equipes podem salvar filtros, agrupamentos e classificação de problemas para visualizar facilmente os problemas que atendem aos critérios especificados. A visualização contém problemas agrupados por Responsável e classificados por Prioridade para mostrar os problemas mais importantes e o status dos problemas para cada desenvolvedor. Quando Jorge recebe as tarefas a serem concluídas, ele planeja o trabalho criando um problema para cada tarefa. Ao criar problemas, Jorge pode escolher o Status, a Prioridade e o esforço de Estimativa de trabalho apropriados. Para problemas maiores, Jorge adiciona tarefas ao problema, para dividir o trabalho em objetivos menores. Jorge cria os problemas com um status de rascunho, como backlog, pois ele não planeja começar a resolvê-los imediatamente. Os problemas com status de rascunho aparecem na visualização Rascunhos, onde devem ser planejados e priorizados. Quando Jorge estiver pronto para começar o trabalho, ele vai mover o problema correspondente para o quadro, atualizando o status para outra categoria (Não iniciado, Iniciado ou Concluído). Conforme cada tarefa está sendo trabalhada, a equipe pode filtrar por título, status, responsável, rótulo, prioridade e estimativa para encontrar um problema específico ou problemas semelhantes que correspondam ao parâmetro especificado. Usando o quadro, Jorge e sua equipe podem ver o número de tarefas concluídas em cada edição e acompanhar o day-to-day progresso arrastando cada problema de um status para o outro até que a tarefa seja concluída. Conforme o projeto progride, os problemas concluídos se acumulam no status Concluído. Wang Xiulan decide removê-los da visualização arquivando-os com

o botão de arquivamento rápido, para que os desenvolvedores possam se concentrar nos problemas relacionados ao trabalho atual e futuro.

Ao planejar o trabalho, os desenvolvedores que trabalham no projeto selecionam Classificar por e Agrupar por para encontrar os problemas que desejam mover do backlog para o quadro. Eles podem optar por adicionar problemas ao quadro com base nas solicitações de maior prioridade dos clientes, agrupam o quadro por um rótulo Solicitação do cliente e classificam por Prioridade. Eles também podem classificar por estimativa para garantir que estejam assumindo um volume de trabalho que possam realizar. A gerente do projeto, Saanvi Sarkar, revisa e prepara regularmente o backlog para ajudar a garantir que a prioridade reflita com precisão a importância de cada problema para o sucesso do projeto.

## Tópicos

- [Conceitos de problemas](#)
- [Acompanhamento do trabalho com problemas](#)
- [Organização do trabalho com backlogs, rótulos e quadros](#)
- [Cotas para edições em CodeCatalyst](#)

# Conceitos de problemas

Criar um problema é uma forma rápida e eficiente de monitorar o trabalho que está sendo feito em um projeto. Você pode usar os problemas para discutir o trabalho em reuniões diárias de sincronização, priorizar o trabalho e muito mais.

Esta página inclui uma lista de conceitos que ajudarão você a usar os problemas de forma eficaz em CodeCatalyst.

## Problemas ativos

Problemas ativos são problemas que não estão no status Rascunho ou arquivados. Em outras palavras, problemas ativos são problemas com status em qualquer uma das seguintes categorias de status: Não iniciado, Iniciado e Concluído. Para ter mais informações sobre status e categorias de status, consulte [Status e categorias de status](#).

Você pode visualizar todos os problemas ativos no projeto na visualização padrão de Problemas ativos.

## Problemas arquivados

Um problema arquivado é um problema que não é mais relevante para seu projeto. Por exemplo, você pode [arquivar um problema](#) se ele estiver concluído e você não precisar mais vê-lo na coluna Concluído ou se ele tiver sido criado por engano. Os problemas arquivados podem ser desarquivados, se necessário.

## Destinatário

O destinatário é a pessoa à qual o problema foi atribuído. Se a pessoa não aparecer na lista quando você pesquisar por ela, isso indica que ela não foi adicionada ao projeto. Para adicioná-la, consulte [Convite de um usuário para um projeto](#). Para habilitar vários responsáveis por um problema, consulte [Ativação ou desativação de vários responsáveis](#). Problemas com vários responsáveis aparecerão no quadro com avatares de cores diferentes, cada um representando um dos responsáveis.

## Campos personalizados

Os campos personalizados permitem que você personalize diferentes atributos de um problema de acordo com suas necessidades de rastreamento e manutenção de problemas em um projeto. Por exemplo, você pode adicionar um campo para roteiro, uma data de vencimento específica ou um campo do solicitante.

## Estimativa

No desenvolvimento Agile, a estimativa é conhecida como pontos históricos. Você pode usar a estimativa de um problema para representar a quantidade de trabalho necessária, além da ambiguidade e complexidade do problema. Considere o uso de estimativas mais altas para problemas com maior risco, dificuldade e incógnitas.

Para ter mais informações sobre os tipos de estimativa e como configurá-los, consulte [Configuração da estimativa do esforço do problema](#).

## Problema

Um problema é um registro que rastreia o trabalho relacionado ao seu projeto. Você pode criar um problema para um recurso, uma tarefa, um bug ou qualquer outro trabalho relacionado ao projeto. Se você estiver usando o desenvolvimento ágil, um problema também poderá descrever uma história épica ou de usuário.

## Rótulo

O rótulo é usado para agrupar, classificar e filtrar problemas. Você pode inserir um novo nome de rótulo ou escolher um dos rótulos na lista preenchida. Essa lista consiste em rótulos usados recentemente no projeto. Um problema pode ter vários rótulos e um rótulo pode ser removido de um problema. Para personalizar rótulos, consulte [Classificação do trabalho com rótulos](#).

## Prioridade

Prioridade se refere ao nível de importância do problema. Há quatro opções: Baixa, Média, Alta e Sem prioridade.

## Status e categorias de status

O status é o estado atual do problema e é usado para verificar rapidamente o progresso de um problema em seu ciclo de vida, desde o início até a conclusão. Todos os problemas devem ter um status, e cada status pertence a uma categoria. As categorias de status são usadas para ajudar a organizar seus status e preencher as visualizações do problema padrão.

Há cinco status padrão e quatro categorias de status em CodeCatalyst. Você pode criar outros status, mas não pode criar outras categorias de status. A lista a seguir contém os status padrão e as categorias correspondentes entre parênteses: Backlog (Rascunho), A fazer (Não iniciado), Em andamento (Iniciado), Em revisão (Iniciado) e Concluído (Concluído).

Para ter mais informações sobre como trabalhar com status, consulte [Acompanhamento do trabalho com status personalizados](#).

## Tarefas

As tarefas podem ser adicionadas aos problemas para detalhar e organizar ainda mais o trabalho desse problema. Você pode adicionar tarefas a um problema na criação ou adicionar tarefas a um problema existente. Ao visualizar um problema, você pode reordenar, remover ou marcar as tarefas como concluídas.

## Visualizações

Os problemas em seu CodeCatalyst projeto são exibidos em visualizações. As visualizações podem ser visualizações em grade que mostram problemas em formato de lista ou visualizações em quadro



que mostram problemas como blocos em colunas organizadas por status do problema. Há quatro visualizações padrão e você pode [criar suas próprias visualizações com agrupamento, filtragem e classificação personalizados](#). A lista a seguir contém detalhes sobre as quatro visualizações padrão.

- A visualização Rascunhos é uma visualização em grade que mostra problemas que não estão sendo resolvidos no momento. Qualquer problema criado com um status na categoria Rascunho aparece nessa visualização. Essa visualização pode ser usada pelas equipes para ver quais problemas ainda estão sendo definidos ou aguardando para serem atribuídos e resolvidos.
- A visualização Problemas ativos é uma visão geral de todos os problemas que estão sendo resolvidos no momento. Qualquer problema com um status nas categorias Não iniciado, Iniciado ou Concluído aparecerá nessa visualização.
- A visualização Todos os problemas é uma visualização em grade que mostra todos os problemas do projeto, tanto rascunhos quanto problemas ativos.
- A visualização Arquivado mostra todos os problemas arquivados.

## Acompanhamento do trabalho com problemas

Você pode planejar e acompanhar seu trabalho em um projeto usando problemas. Cada problema é um trabalho mantido em um registro distinto. Os problemas podem ser divididos em tarefas para organizar e acompanhar ainda mais o trabalho desse problema. Você também pode criar links entre problemas para acompanhar o trabalho relacionado, adicionar rótulos para organizar e classificar o trabalho, agrupar problemas, atribuir prioridades ao trabalho e indicar se o trabalho está bloqueado.

Quando estiver pronto para trabalhar em um problema ou conjunto de problemas, você poderá estimar o trabalho, atribuí-lo aos usuários e adicionar comentários para ajudar outras pessoas a entender o trabalho e o andamento. Você também pode exportar problemas para ajudar a trazer as informações que eles contêm para outros formatos.

## Criando um problema no CodeCatalyst

As equipes de desenvolvimento criam problemas para ajudar a monitorar e gerenciar o trabalho. Você pode criar problemas em um projeto com base nas suas necessidades. Por exemplo, é possível criar um problema para acompanhar a atualização de uma variável no código. Você pode atribuir problemas a outros usuários no projeto, usar rótulos para monitorar seu trabalho e muito mais.

Siga estas instruções para criar um problema no CodeCatalyst.

## Para criar um problema

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto onde você deseja criar um problema.
3. Na página inicial do projeto, escolha Criar problema. Como alternativa, no painel de navegação, selecione Problemas.
4. Selecione Criar problema.

### Note

Você também pode adicionar problemas em linha ao usar uma visualização em grade.

5. Insira um título para o problema.
6. (Opcional) Insira uma Descrição. Você pode usar Markdown para adicionar formatação.
7. (Opcional) Escolha um Status, Prioridade e Estimativa para o problema.

### Note

Se a configuração de estimativa do projeto estiver definida como Ocultar estimativas, não haverá um campo Estimativa.

8. (Opcional) Adicione tarefas ao problema. As tarefas podem ser usadas para dividir o trabalho de um problema em objetivos menores. Para adicionar uma tarefa, selecione + Adicionar tarefas. Em seguida, insira o nome da tarefa no campo de texto e pressione enter. Depois de adicionar tarefas, você pode marcá-las como concluídas marcando a caixa de seleção ou reordená-las escolhendo e arrastando a tarefa do lado esquerdo da caixa de seleção.
9. (Opcional) Adicione um rótulo existente ou crie outro rótulo e adicione-o selecionando + Adicionar rótulo.
  - a. Para adicionar um rótulo existente, escolha o rótulo na lista. Você pode inserir um termo de pesquisa no campo para pesquisar todos os rótulos que contêm esse termo no projeto.
  - b. Para criar um rótulo e adicioná-lo, digite o nome do rótulo que você deseja criar no campo de pesquisa e pressione enter.
10. (Opcional) Adicione um destinatário selecionando + Adicionar um destinatário. Você pode se adicionar rapidamente como destinatário selecionando + Me adicionar.

 Tip

Você pode optar por atribuir um problema ao Amazon Q para que o Amazon Q tente resolver o problema. Para obter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#). Esse recurso está disponível somente na região do Oeste dos EUA (Oregon). Essa funcionalidade exige que recursos de IA generativa estejam habilitados para o espaço. Para ter mais informações, consulte [Gerenciar recursos de IA generativa](#).

11. (Opcional) Adicione um campo personalizado existente ou crie outro. Os problemas podem ter vários campos personalizados.
  - a. Para adicionar um campo personalizado existente, escolha o campo personalizado na lista. Você pode inserir um termo de pesquisa no campo para pesquisar todos os campos personalizados que contêm esse termo no projeto.
  - b. Para criar um campo personalizado e adicioná-lo, digite o nome do campo personalizado que você deseja criar no campo de pesquisa e pressione enter. Depois, escolha o tipo de campo personalizado que você deseja criar e defina um valor.
12. Selecione Criar problema. Uma notificação aparece no canto inferior direito: se o problema foi criado, uma mensagem de confirmação será exibida informando que o problema foi criado. Se o problema não tiver sido criado, uma mensagem com o motivo da falha será exibida. Depois, você pode selecionar Tentar novamente para editar e tentar criar o problema novamente ou selecionar Descartar para descartar o problema. As duas opções ignorarão a notificação.

 Note

Você não pode vincular uma solicitação pull a um problema ao criá-lo. No entanto, você pode [editá-lo](#) depois de criá-lo para adicionar links às solicitações pull.

## Práticas recomendadas ao criar e trabalhar com problemas atribuídos ao Amazon Q

Quando você cria problemas, às vezes, alguns deles persistem. As causas para isso podem ser complexas e variáveis. Às vezes, isso acontece porque não está claro quem deve trabalhar nisso. Outras vezes, o problema exige pesquisa ou experiência em uma parte específica da base de código e os melhores candidatos para o trabalho estão ocupados com outros problemas. Geralmente, há outros trabalhos urgentes que devem ser realizados primeiro. Qualquer uma ou todas essas causas

podem resultar em problemas que não foram resolvidos. CodeCatalyst inclui integração com um assistente generativo de IA chamado Amazon Q, que pode analisar um problema com base no título e na descrição. Se você atribuir o problema ao Amazon Q, ele tentará criar um rascunho de solução para você avaliar. Isso pode ajudar você e sua equipe a se concentrarem e otimizarem o trabalho em questões que exigem sua atenção, enquanto o Amazon Q trabalha em uma solução para problemas que você não tem recursos para resolver imediatamente.

#### Note

##### Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo do conteúdo, Recomendar tarefas, Usar o Amazon Q para criar ou adicionar recursos a um projeto e Atribuir problemas ao Amazon Q com os recursos do agente do Amazon Q Developer para desenvolvimento de software são baseados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança, a proteção e o uso responsável de inteligência artificial.

O Amazon Q tem melhor desempenho em questões e problemas simples. Para ter melhores resultados, use uma linguagem simples para explicar claramente o que você quer que seja feito. Confira a seguir algumas práticas recomendadas para ajudar você a criar problemas otimizados para que o Amazon Q possa trabalhar.

#### Important

Os recursos de IA generativa estão disponíveis somente na região Oeste dos EUA (Oregon).

- Seja simples. O Amazon Q se sai melhor com alterações e correções simples de código que podem ser explicadas no título e na descrição do problema. Não atribua problemas com títulos vagos ou descrições muito exageradas ou contraditórias.
- Seja específico. Quanto mais informações você puder fornecer sobre as alterações exatas necessárias para resolver o problema, maior a probabilidade de o Amazon Q criar uma solução

que resolva o problema. Se possível, inclua detalhes específicos, como o nome APIs que você deseja alterar, os métodos que deseja atualizar, os testes que precisam de alterações e quaisquer outros detalhes que você possa imaginar.

- Inclua todos os detalhes no título e na descrição do problema antes de atribuí-lo ao Amazon Q. Você não pode alterar o título ou a descrição de um problema depois de atribuí-lo ao Amazon Q, portanto, tenha todas as informações necessárias em um problema antes de atribuí-lo ao Amazon Q.
- Atribua somente problemas que exijam alterações de código em um único repositório de origem. O Amazon Q só pode trabalhar com código em um único repositório de origem em CodeCatalyst. Não há suporte a repositórios vinculados. Verifique se o problema exige apenas alterações em um único repositório de origem antes de atribuí-lo ao Amazon Q.
- Use o padrão sugerido pelo Amazon Q para aprovar cada etapa. Por padrão, o Amazon Q exigirá sua aprovação para cada etapa realizada. Isso permite que você interaja com o Amazon Q em comentários não apenas sobre o problema, mas também sobre qualquer solicitação pull criada por ele. Isso proporciona uma experiência mais interativa com o Amazon Q, que ajuda você a ajustar a abordagem e refinar o código criado para resolver o problema.

 Note

O Amazon Q não responde a comentários individuais sobre problemas ou solicitações pull, mas os revisará quando solicitado a reconsiderar a abordagem ou criar uma revisão.

- Sempre analise cuidadosamente a abordagem sugerida pelo Amazon Q. Depois de aprovar a abordagem, o Amazon Q começará a trabalhar na geração de código com base nessa abordagem. A abordagem deve parecer correta e incluir todos os detalhes que você espera antes de pedir ao Amazon Q que continue.
- Permita que o Amazon Q trabalhe em fluxos de trabalho somente se você não tiver fluxos de trabalho existentes que possam implantá-los antes de serem revisados. Seu projeto pode ter fluxos de trabalho configurados para iniciar execuções em eventos de solicitação pull. Nesse caso, qualquer solicitação pull criada pela Amazon Q que inclua a criação ou atualização do YAML do fluxo de trabalho pode iniciar uma execução desses fluxos de trabalho incluídos na solicitação pull. Como prática recomendada, não permita que o Amazon Q trabalhe em arquivos de fluxo de trabalho, a menos que tenha certeza de que não há fluxos de trabalho em seu projeto que executem automaticamente esses fluxos de trabalho antes de você revisar e aprovar a solicitação pull criada por ele.

Para ter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#) e [Gerenciamento de recursos de IA generativa](#).

## Estimativa de um problema

No desenvolvimento Agile, a estimativa é conhecida como pontos históricos. Você pode usar a estimativa de um problema para representar a quantidade de trabalho necessária, além da ambiguidade e complexidade do problema. Considere o uso de estimativas mais altas para problemas com maior risco, dificuldade e incógnitas.

Antes de começar a estimar os problemas, você deve primeiro escolher o tipo de estimativa que deseja usar para seu projeto. Existem dois tipos para selecionar por padrão. Para usar Tamanho da camiseta ou Sequenciamento de Fibonacci de forma eficaz, sua equipe deve se alinhar ao que cada tamanho representa. Decidam juntos o que cada estimativa representa para você e comece a aplicar essas estimativas a cada problema. Considere revisar periodicamente

Siga estas etapas para definir a configuração das estimativas de esforço para problemas em CodeCatalyst.

Para configurar a estimativa de esforço para problemas

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Estimativa, na seção Configurações básicas, escolha como os valores da estimativa serão exibidos. Os tipos de estimativas disponíveis são Tamanho de camisetas, Sequenciamento de Fibonacci ou Ocultar estimativas. Se a configuração de estimativa do projeto estiver definida como Ocultar estimativas, não haverá um campo Estimativa nos problemas do projeto.

Quando o tipo de estimativa for atualizado, nenhum dado será perdido e o valor estimado de todos os problemas será convertido automaticamente. O mapeamento da conversão é mostrado na tabela a seguir.

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| PP                  | 1                      |
| PP                  | 2                      |

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| S                   | 3                      |
| M                   | 5                      |
| L                   | 8                      |
| GG                  | 13                     |

Para adicionar ou alterar uma estimativa para um problema, você pode [editar o problema](#).

## Editando e colaborando em questões em CodeCatalyst

### Sumário

- [Edição de um problema](#)
- [Trabalhar com anexos](#)
  - [Visualização e gerenciamento de anexos](#)
- [Gerenciamento de tarefas sobre problemas](#)
- [Vincular um problema a outro](#)
- [Marcação de um problema como bloqueado ou desbloqueado](#)
- [Adição, edição ou exclusão de comentários](#)
  - [Uso de menções em um comentário](#)

### Edição de um problema

Siga estas etapas para editar o título, a descrição, o status, o responsável, a prioridade, a estimativa ou os rótulos de um problema.

#### Para editar um problema

1. Escolha o problema que você deseja editar para ver os detalhes do problema. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Para editar o título do problema, selecione o título, insira um novo e pressione enter.
3. Para editar a descrição, escolha a descrição, insira uma nova descrição e pressione enter. Você pode usar Markdown para adicionar formatação.

4. Em Tarefas, você pode visualizar e gerenciar as tarefas do problema. Se não houver tarefas, você pode fazer com que o Amazon Q analise o problema e recomende tarefas que possam dividir o trabalho no problema em itens separados, cada um atribuído a um usuário. Para obter mais informações, consulte [Gerenciamento de tarefas sobre problemas](#).
5. Para editar o Status, a Estimativa ou a Prioridade, escolha uma opção nos respectivos menus suspensos.
6. Em Rótulos, é possível adicionar um rótulo existente, criar ou remover um rótulo.
  - a. Para adicionar um rótulo existente, selecione + Adicionar rótulo e escolha o rótulo na lista. Você pode inserir um termo de pesquisa no campo para pesquisar todos os rótulos que contêm esse termo no projeto.
  - b. Para criar um rótulo e adicioná-lo, selecione + Adicionar rótulo, digite o nome do rótulo que você deseja criar no campo de pesquisa e pressione enter.
  - c. Para remover um rótulo, escolha o ícone X ao lado do rótulo que você deseja remover. Se você remover um rótulo de todos os problemas, o rótulo aparecerá na seção Rótulos não usados na seção Rótulos das configurações do problema. Os rótulos não usados aparecem no final da lista de rótulos ao usar filtros ou adicionar rótulos a um problema. Você pode encontrar uma visão geral de todos os rótulos (usados e não usados) e dos problemas que os contêm nas configurações do problema.
7. Para atribuir um problema, selecione + Adicionar um destinatário na seção Destinatário, pesquise e escolha o destinatário na lista. Você pode selecionar + Me adicionar para se adicionar rapidamente como destinatário.
8. Em Anexos, você pode adicionar, baixar ou remover anexos. Para obter mais informações, consulte [Trabalhar com anexos](#).
9. Para vincular uma solicitação pull, escolha Vincular solicitação pull e, depois, escolha uma solicitação pull na lista ou insira o URL ou ID. Para desvincular uma solicitação pull, escolha o ícone de desvinculação.

 Tip

Depois de adicionar um link a uma solicitação pull para um problema, você pode navegar rapidamente até ele escolhendo o ID correspondente na lista de solicitações pull vinculadas. Você pode usar o URL de uma solicitação pull para vincular solicitações pull que estão em projetos diferentes do quadro de problemas, mas somente usuários que são membros desse projeto poderão visualizar ou navegar até essa solicitação pull.



10. (Opcional) Adicione e defina um campo personalizado existente, crie ou remova um campo personalizado. Os problemas podem ter vários campos personalizados.
  - a. Para adicionar um campo personalizado existente, escolha o campo personalizado na lista. Você pode inserir um termo de pesquisa no campo para pesquisar todos os campos personalizados que contêm esse termo no projeto.
  - b. Para criar um campo personalizado e adicioná-lo, digite o nome do campo personalizado que você deseja criar no campo de pesquisa e pressione enter. Depois, escolha o tipo de campo personalizado que você deseja criar e defina um valor.
  - c. Para remover um campo personalizado, escolha o ícone X ao lado do campo personalizado que você deseja remover. Se você remover um campo personalizado de todos os problemas, o campo personalizado será excluído e você não o verá mais durante a filtragem.

## Trabalhar com anexos

Você pode adicionar anexos às edições para tornar os CodeCatalyst arquivos relacionados facilmente acessíveis. Siga este procedimento para gerenciar os anexos de um problema.

O tamanho dos anexos adicionados aos problemas conta para as cotas de armazenamento do seu espaço. Para ter mais informações sobre como visualizar e gerenciar os anexos do projeto, consulte [Visualização e gerenciamento de anexos](#).

### Important

Os anexos das edições não são digitalizados nem analisados pela Amazon. CodeCatalyst Qualquer usuário pode adicionar um anexo a um problema que possa conter código ou conteúdo malicioso. Os usuários devem conhecer as práticas recomendadas quando se trata de gerenciar anexos e se proteger contra códigos, conteúdos ou vírus maliciosos.

Para adicionar, baixar ou remover anexos

1. Escolha o problema para o qual deseja gerenciar os anexos. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Para adicionar um anexo, selecione Carregar arquivo. Navegue até o arquivo no explorador de arquivos do sistema operacional e selecione-o. Selecione Abrir para adicioná-lo como anexo.

Para ter informações sobre cotas, como tamanho máximo de anexo, consulte [Cotas para edições em CodeCatalyst](#).

Observe as seguintes restrições aos nomes de arquivos anexos e tipos de conteúdo:

- Os seguintes caracteres não são aceitos em nomes de arquivo:
  - Caracteres de controle: 0x00–0x1f e 0x80–0x9f
  - Caracteres reservados: /, ?, <, >, \, :, \*, | e "
  - Nomes de arquivos reservados do Unix: . e ..
  - Períodos e espaços finais
  - Nomes de arquivos reservados do Windows: CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9
- O tipo de conteúdo do anexo deve seguir este padrão de tipo de mídia:

```
media-type = type "/" [tree "."] subtype ["+" suffix]* [";" parameter];
```

Por exemplo, `.text/html; charset=UTF-8`

3. Para baixar um anexo, escolha o menu de reticências ao lado do anexo que você deseja baixar e escolha Baixar.
4. Para copiar o URL de um anexo, escolha o menu de reticências ao lado do anexo do qual você deseja copiar o URL e selecione Copiar URL.
5. Para remover um anexo, escolha o menu de reticências ao lado do anexo que você deseja remover e escolha Excluir.

## Visualização e gerenciamento de anexos

Você pode ver uma tabela com cada anexo adicionado aos problemas do projeto nas configurações do problema. Essa tabela inclui detalhes de cada anexo, incluindo informações como tipo de conteúdo, quando foi adicionado, o problema ao qual foi adicionado, o status e o tamanho do arquivo.

Essa tabela pode ser usada para identificar facilmente anexos grandes em problemas que foram concluídos ou arquivados para removê-los e liberar espaço de armazenamento.

 Important

Os anexos das edições não são digitalizados nem analisados pela Amazon. CodeCatalyst Qualquer usuário pode adicionar um anexo a um problema que possa conter código ou conteúdo malicioso. Os usuários devem conhecer as práticas recomendadas quando se trata de gerenciar anexos e se proteger contra códigos, conteúdos ou vírus maliciosos.

Para visualizar e gerenciar todos os anexos de problemas em um projeto

1. No painel de navegação, selecione Problemas.
2. Escolha o ícone de reticências e selecione Configurações.
3. Escolha a guia Anexos.

## Gerenciamento de tarefas sobre problemas

As tarefas podem ser adicionadas aos problemas para detalhar, organizar e acompanhar ainda mais o trabalho desse problema. Você mesmo pode criar tarefas ou usar o Amazon Q para recomendar tarefas com base na análise e na complexidade do problema.

O Amazon Q Developer é um assistente conversacional generativo baseado em IA que pode ajudar você a entender, criar, estender e operar aplicativos. AWS Para acelerar sua construção AWS, o modelo que impulsiona o Amazon Q é aprimorado com AWS conteúdo de alta qualidade para produzir respostas mais completas, acionáveis e referenciadas. Para obter mais informações, consulte [O que é Amazon Q Developer?](#) no Guia do usuário do Amazon Q Developer.

Para gerenciar tarefas em um problema

1. Escolha o problema para o qual deseja gerenciar as tarefas. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Em Tarefas, você pode visualizar e gerenciar tarefas do problema.
  1. Para adicionar uma tarefa, insira o nome da tarefa no campo de texto e pressione enter.
  2. Se não houver tarefas para o problema, você pode optar por fazer com que o Amazon Q analise o problema e crie tarefas com base no título, na descrição e na análise da complexidade do problema e no código do repositório. Nesse caso, selecione Recomendar tarefas. Você precisará especificar o repositório de origem que contém o código do problema. Selecione Iniciar recomendação de tarefas para iniciar a análise da recomendação de tarefas.

Essa caixa de diálogo será fechada. Quando a recomendação estiver concluída, selecione Visualizar tarefas recomendadas para revisar as tarefas e realizar qualquer ação necessária, como excluir ou adicionar tarefas à lista ou reordenar as tarefas recomendadas, antes de selecionar Criar tarefas.

Depois que as tarefas forem criadas, você poderá atribuí-las aos usuários e trabalhar com eles da mesma forma que trabalha com tarefas criadas manualmente.

3. Para marcar uma tarefa como concluída, marque a caixa de seleção da tarefa.
4. Para visualizar ou atualizar os detalhes de uma tarefa, escolha-a na lista.
5. Para reordenar as tarefas, escolha e arraste a tarefa do lado esquerdo da caixa de seleção.
6. Para remover uma tarefa, escolha o menu de reticências da tarefa e escolha Remover.

## Vincular um problema a outro

Se um problema estiver relacionado ao trabalho em outro problema, você poderá vinculá-lo. Esta é uma maneira rápida de ajudar você a entender e monitorar as dependências entre os itens de trabalho.

Ao vincular um problema, há quatro estados para escolher:

- Relacionado a: use esse estado para indicar o trabalho relacionado ao problema vinculado.
- Bloqueado por: use esse estado para indicar que o problema está bloqueado pelo problema vinculado.
- Bloqueando para: use esse estado para indicar que o problema está bloqueando o andamento do problema vinculado.
- Duplicado: use esse estado para indicar que o problema duplica o trabalho capturado em outro problema.

Você não pode alterar o estado de um link depois de criá-lo. Os links e o estado do link aparecem em Problemas vinculados tanto no problema em que você criou o link quanto no problema vinculado. Você também pode remover links depois de criados.

Para vincular um problema a outro problema ou outra tarefa

1. Abra o problema que você deseja vincular a outro problema. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).

2. Selecione Vincular problemas.
3. Em Marcar como, escolha o estado do link.

Em N° do problema, insira o número do problema ou escolha-o na lista suspensa.

4. Para adicionar outro link, selecione Adicionar problema vinculado.
5. Quando terminar, selecione Atualizar para atualizar o problema e todos os problemas vinculados ao relacionamento vinculado.

## Marcação de um problema como bloqueado ou desbloqueado

Se algo estiver impedindo você de resolver um problema, convém marcar como bloqueado. Por exemplo, o problema pode ser bloqueado se depender de uma alteração em outra parte da base de código que ainda não foi mesclada.

Quando você marca um problema como bloqueado, CodeCatalyst adiciona um rótulo vermelho de Bloqueado ao problema, tornando-o altamente visível em sua lista de pendências, arquivo ou em seu quadro.

Você pode desbloquear o problema quando circunstâncias externas forem resolvidas.

Para marcar um problema como bloqueado

1. Abra o problema que você deseja marcar como bloqueado. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Selecione Ações e Marcar como bloqueado.

Para desbloquear um problema

1. Abra o problema que você deseja desbloquear. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Selecione Ações e Marcar como desbloqueado.

## Adição, edição ou exclusão de comentários

Você pode deixar um comentário sobre um problema. Nos comentários, você pode marcar outros membros do espaço, outros projetos no espaço, problemas relacionados e código.

## Para adicionar um comentário a um problema

1. Navegue até o projeto.
2. Na barra de navegação, selecione Problemas.
3. Escolha o problema onde você deseja adicionar o comentário. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
4. Insira o comentário no campo Comentários. Você pode usar Markdown para adicionar formatação.
5. Selecione Enviar.

## Para editar um comentário

Você pode editar os comentários que faz sobre problemas. Você só pode editar comentários que foram de sua autoria.

1. Navegue até o projeto.
2. Na barra de navegação, selecione Problemas.
3. Escolha o problema onde você deseja editar um comentário. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
4. Para editar um comentário, encontre o comentário que você deseja editar.

### Tip

Você pode classificar os comentários pelos mais antigos ou pelos mais novos primeiro. São carregados 10 comentários por vez.

5. Escolha ícone de reticências e Editar:
6. Edite o comentário. Você pode usar Markdown para adicionar formatação.
7. Escolha Salvar. O comentário agora está atualizado.

## Para excluir um comentário

Você pode excluir os comentários que faz sobre problemas. Você só pode excluir comentários que foram de sua autoria. Quando um comentário for excluído, seu nome de usuário será exibido, mas com as palavras Este comentário foi excluído no lugar do texto original do comentário.

1. Navegue até o projeto.
2. Na barra de navegação, selecione Problemas.
3. Escolha o problema onde você deseja excluir um comentário. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
4. Escolha o ícone de reticências, selecione Excluir e, depois, Confirmar.

### Uso de menções em um comentário

Você pode mencionar membros do espaço, outros projetos no espaço, problemas relacionados e código nos comentários. Isso cria um link rápido para o usuário ou recurso que você mencionou.

### Para mencionar em um comentário

1. Navegue até o projeto.
2. Na barra de navegação, selecione Problemas.
3. Escolha o problema que você deseja editar para ver os detalhes do problema. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
4. Selecione a caixa de texto Adicionar um comentário.
5. Digite `@user_name` para mencionar outro usuário.
6. Digite `@project_name` para mencionar um projeto.
7. Digite `@issue_name` ou `@issue_number` para mencionar outro problema.
8. Digite `@file_name` para mencionar arquivos ou códigos específicos em um repositório de origem.

#### Note

Uma lista dos 5 principais itens (usuários, repositórios de origem, projetos etc.) contendo os termos que correspondem à `@mention` será preenchida conforme você digita.

9. Escolha o item que deseja mencionar. O caminho que mostra onde o item está localizado será preenchido na caixa de texto do comentário.
10. Conclua seu comentário e selecione Enviar.

## Descoberta e visualização de problemas

As seções a seguir descrevem como pesquisar e visualizar problemas em um CodeCatalyst projeto com eficiência.

### Procura de um problema

Você pode encontrar um problema procurando parâmetros específicos. Para ter mais informações sobre como refinar sua pesquisa, consulte [Pesquisa do código, problemas, projetos e usuários no CodeCatalyst](#).

#### Como procurar um problema

1. Navegue até o projeto.
2. Use a barra de pesquisa para procurar problemas ou informações relacionadas a problemas. Você pode usar parâmetros de consulta para refinar a pesquisa. Para obter mais informações, consulte [Pesquisa do código, problemas, projetos e usuários no CodeCatalyst](#).

### Classificação de problemas

Por padrão, os problemas em CodeCatalyst são classificados por ordem manual. A ordem manual exibe os problemas na ordem para a qual eles são movidos pelos usuários. Você pode arrastar e soltar os problemas quando classificados em ordem manual para alterar a ordem. Essa opção de classificação é útil para organizar o backlog de problemas e priorizá-los.

A tabela a seguir mostra como os problemas podem ser classificados nas visualizações em grade e em quadro.

| Opções de classificação da visualização em grade | Opções de classificação da visualização em quadro |
|--------------------------------------------------|---------------------------------------------------|
| Ordem manual                                     | Ordem manual                                      |
| Última atualização                               | Última atualização                                |
| Prioridade                                       | Prioridade                                        |
| Estimativa                                       | Estimativa                                        |



| Opções de classificação da visualização em grade | Opções de classificação da visualização em quadro |
|--------------------------------------------------|---------------------------------------------------|
| Título                                           | Título                                            |
| ID                                               |                                                   |
| Status                                           |                                                   |
| Bloqueado                                        |                                                   |
| Campos personalizados                            |                                                   |

Use o procedimento a seguir para alterar a forma como os problemas são classificados.

Para classificar problemas

1. Navegue até o projeto.
2. No painel de navegação, selecione Problemas. A visualização padrão é o Quadro.
3. (Opcional) Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegue até uma visualização de problemas diferente.
4. Para classificar uma visualização em grade, há duas opções:
  - a. Escolha o cabeçalho do campo pelo qual você deseja classificar. A escolha do cabeçalho alternará entre a ordem crescente e decrescente.
  - b. Escolha o menu suspenso Classificar por e escolha um parâmetro pelo qual classificar. Os problemas serão classificados em ordem crescente.
5. Para classificar uma visualização em quadro, escolha o menu suspenso Classificar por e escolha um parâmetro para classificar. Os problemas serão classificados em ordem crescente.

## Agrupamento de problemas

O agrupamento é usado para organizar os problemas no quadro por meio de vários parâmetros, como destinatário, rótulos e prioridade.

Para agrupar problemas

1. Navegue até o projeto.

2. No painel de navegação, selecione Problemas. A visualização padrão é o Quadro.
3. (Opcional) Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegue até uma visualização de problemas diferente.
4. Escolha Grupo.
5. Em Agrupar por, escolha um parâmetro pelo qual agrupar:
  - Se você escolher Destinatário ou Prioritário, selecione a Ordem de agrupamento.
  - Se você escolher Rótulo, escolha os rótulos e, depois, selecione Ordem de agrupamento.
6. (Opcional) Escolha a opção Mostrar grupos vazios para mostrar ou ocultar grupos que não têm problemas atualmente atribuídos a eles.
7. A visualização é atualizada conforme você faz suas escolhas. Um problema aparece somente no grupo que corresponde aos parâmetros configurados.

## Filtragem de problemas

Use a filtragem para encontrar problemas que contenham nome, prioridade, rótulo, campos personalizados ou destinatário especificados.

### Para filtrar problemas

1. Navegue até o projeto.
2. No painel de navegação, selecione Problemas.
3. (Opcional) Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegue até uma visualização de problemas diferente.

#### Note

Para filtrar com base em uma string no nome ou na descrição do problema, insira a string na barra de pesquisa de problemas.

4. Selecione Filtro e, depois, + Adicionar filtro.
5. Escolha os parâmetros para filtrar. É possível escolher vários filtros e parâmetros. Você pode configurar filtros para mostrar problemas que correspondam a cada filtro ou a qualquer filtro individual selecionando e ou ou. A visualização será atualizada para mostrar os problemas que correspondem ao filtro.

## Progresso em um problema

Cada problema tem um ciclo de vida. Em CodeCatalyst, os problemas geralmente começam como um rascunho na lista de pendências. Quando for necessário iniciar o trabalho para resolver esse problema, ele será movido para outra categoria de status e passará por vários status até ser concluído e arquivado. Você pode mover ou progredir um problema pelo ciclo de vida das seguintes maneiras:

- Você pode mover um problema entre o backlog e o quadro.
- Você pode mover um problema em andamento por vários estágios de conclusão.
- Você pode arquivar um problema que foi concluído.

### Transferência de um problema entre o backlog e o quadro

Você pode mover um problema do backlog para o quadro assim que começar a trabalhar nele. Você também pode mover um problema de volta para o backlog se o trabalho for adiado.

Para mover um problema entre o backlog e o quadro

1. Navegue até o projeto.
2. No painel de navegação, selecione Problemas. A visualização padrão é o Quadro.
3. Para mover um problema do quadro para o backlog:
  - a. Escolha o problema que você deseja mover. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
  - b. Selecione Backlog no menu suspenso Status.
4. Para mover um problema do backlog para o quadro:
  - a. Para navegar até o backlog, selecione Quadro e Backlog.
  - b. Escolha o problema que você deseja mover. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
  - c. Selecione Adicionar ao quadro ou escolha um Status diferente de Backlog.

### Avançar com um problema pelos estágios do ciclo de vida no quadro

Você pode mover um problema dentro de um quadro por diferentes status até ele ser concluído.

Para mover um problema dentro do quadro

1. No painel de navegação, selecione Problemas. A visualização padrão é o Quadro.
2. Execute um destes procedimentos:
  - Arraste e solte um problema em outro status.
  - Escolha um problema e, depois, escolha um status no menu suspenso Status.
  - Escolha um problema e, em seguida, escolha Mover para: **next-status**.

Para ter informações sobre como arquivar um problema, consulte [Arquivamento de um problema](#).

## Movimentar problemas entre grupos

Você pode [agrupar problemas](#) nas visualizações Todos os problemas e Quadro por meio de vários parâmetros. Se os problemas estiverem agrupados, você poderá mover problemas de um grupo para outro. Mover um problema de um grupo para outro editará automaticamente o campo em que os problemas estão agrupados para corresponder ao grupo de destino.

Como exemplo de cenário, suponha CodeCatalyst que haja uma empresa usando problemas atribuídos a duas pessoas, Wang Xiulan e Saanvi Sarkar. O quadro é agrupado por Assignee, e há dois grupos, um para cada responsável. Mover um problema do grupo Wang Xiulan para o grupo Saanvi Sarkar atualizará o responsável pelo problema para Saanvi Sarkar.

## Arquivamento de um problema

### Note

Os problemas não são excluídos em um projeto; eles são arquivados. Para excluir problemas, você deve excluir o projeto.

Você pode arquivar um problema quando ele não for mais necessário em seu projeto. Ao arquivar um problema, ele é CodeCatalyst removido de todas as visualizações que filtram os problemas arquivados. Os problemas arquivados podem ser vistos na visualização padrão dos Problemas arquivados, na qual podem ser desarquivados, se necessário.

Você vai arquivar um problema se:

- Tiver concluído o problema e não precisar mais dele na coluna Concluído.
- Não tiver planos de trabalhar nisso.
- Tiver criado por engano.
- Tiver atingido o número máximo de problemas ativos.

### Para arquivar um problema

1. Abra o problema que você deseja arquivar. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Selecione Ações e, depois, Mover para arquivo.
3. (Opcional) Para arquivar rapidamente vários problemas com o status Concluído, escolha a elipse vertical na parte superior de qualquer status Concluído no quadro e selecione Arquivar problemas.

### Para desarquivar um problema

1. Abra o problema que você deseja desarquivar. Você pode ver uma lista de problemas arquivados abrindo a visualização Problemas arquivados no menu suspenso do alternador de visualização de problemas. Para receber ajuda com a descoberta do problema, consulte [Descoberta e visualização de problemas](#).
2. Selecione Desarquivar.

## Exportação de problemas

Você pode exportar problemas da visualização atual para um arquivo .xlsx. Para exportar problemas, realize as seguintes etapas.

### Para exportar problemas

1. Navegue até o projeto.
2. Na barra de navegação, selecione Problemas.
3. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegue até a visualização que contém os problemas que você deseja exportar. Somente os problemas mostrados na visualização serão exportados.
4. Escolha o menu de reticências e selecione Exportar para o Excel.

5. O arquivo .xlsx é baixado. Por padrão, ele tem o nome do projeto e a data em que a exportação foi concluída.

## Organização do trabalho com backlogs, rótulos e quadros

Nem todas as equipes trabalham da mesma maneira. Você pode configurar a forma como os problemas aparecem e podem ser atribuídos na Amazon CodeCatalyst para ajudá-lo a entender com precisão o que está sendo trabalhado e o status desse trabalho. Você pode escolher qual método de estimativa permitir para os problemas para que todos os usuários usem o mesmo método. Você pode criar rótulos e status personalizados que também podem ser usados para filtrar a visualização do trabalho. Dependendo de como sua equipe trabalha, você pode configurar se deseja permitir vários responsáveis por um problema ou permitir que apenas um problema seja atribuído a um único usuário. Você também pode criar visualizações personalizadas dos problemas para que o trabalho seja exibido de uma forma que mostre as informações mais relevantes para você ou sua equipe.

## Classificação do trabalho com rótulos

Você pode personalizar rótulos para problemas. Isso inclui editar o rótulo e alterar a cor. Os rótulos podem ajudar você a classificar e organizar seu trabalho. Por exemplo, você pode criar rótulos para aspectos específicos do seu software ou para diferentes grupos ou equipes.

### Tópicos

- [Criação de um rótulo](#)
- [Edição de um rótulo](#)
- [Exclusão de um rótulo](#)

### Criação de um rótulo

Em CodeCatalyst, você cria rótulos adicionando-os ao criar um novo problema ou ao editar um problema existente. Para obter mais informações, consulte [Criando um problema no CodeCatalyst](#) e [Editando e colaborando em questões em CodeCatalyst](#).

### Edição de um rótulo

Use o procedimento a seguir para alterar o nome ou a cor de um rótulo existente.

## Para editar um rótulo

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. No bloco Rótulos, há uma lista dos rótulos usados no projeto. Escolha o ícone de edição ao lado do rótulo que você deseja editar. Siga um dos procedimentos abaixo:
  - a. Edite o nome do rótulo.
  - b. Para alterar a cor, escolha a roda de cores. Use o seletor para escolher uma nova cor.
4. Para salvar as alterações feitas no rótulo, escolha o ícone de marca de seleção.
5. O rótulo alterado agora está visível na sua lista de rótulos disponíveis. Você também pode ver quantos problemas estão usando esse rótulo.

### Note

Você pode escolher o número exibido ao lado de cada rótulo para navegar até a página Todos os problemas e ver todos os problemas que contêm esse rótulo.

## Exclusão de um rótulo

No momento, você não pode excluir um rótulo de problemas em CodeCatalyst. Se você remover um rótulo de todos os problemas, o rótulo aparecerá na seção Rótulos não usados na seção Rótulos das configurações do problema. Os rótulos não usados aparecem no final da lista de rótulos ao usar filtros ou adicionar rótulos a um problema. Você pode encontrar uma visão geral de todos os rótulos (usados e não usados) e dos problemas que os contêm nas configurações do problema.

## Organização do trabalho com campos personalizados

Você pode criar campos personalizados para ajudar a organizar e visualizar o trabalho do projeto. Os campos personalizados são adicionados à lista de filtros disponíveis no Filtro para que você possa filtrar problemas por campos personalizados. Os campos personalizados são pares de nome e valor. Você filtra pelo nome do campo personalizado e, depois, pelo valor desse campo personalizado.

Um problema pode ter vários campos personalizados.

## Criação de um campo personalizado

Em CodeCatalyst, você cria campos personalizados adicionando-os ao criar um problema ou ao editar um problema existente. Para obter mais informações, consulte [Criando um problema no CodeCatalyst](#) e [Editando e colaborando em questões em CodeCatalyst](#).

## Exclusão de um campo personalizado

Para excluir um campo personalizado, você deve remover o campo personalizado de cada problema ao qual ele é adicionado. Quando um campo personalizado for excluído, você não o verá mais no Filtro. Você pode usar filtros para visualizar todos os problemas com um campo personalizado e removê-los editando os problemas. Para ter mais informações, consulte [Descoberta e visualização de problemas](#) e [Edição de um problema](#).

## Acompanhamento do trabalho com status personalizados

Você pode adicionar status personalizados em seu quadro. Cada status personalizado deve pertencer a uma das seguintes categorias: Rascunho, Não iniciado, Iniciado ou Concluído. As categorias de status são usadas para ajudar a organizar os status e preencher as visualizações padrão. Para ter mais informações sobre status e categorias de status, consulte [Status e categorias de status](#). Para obter mais informações sobre visualizações, consulte [Descoberta e visualização de problemas](#).

### Para criar um status

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Status, escolha o ícone de adição ao lado da categoria em que você deseja que o status esteja.
4. Escolha um nome para o status e selecione o ícone de marca de seleção.

#### Note

Escolha o ícone X para cancelar a adição de um status.

O status personalizado agora está visível em seu quadro e é exibido como uma opção ao criar um problema.



## Para editar um status

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Status, escolha o ícone de edição ao lado do status que você deseja editar ou alterar.
4. Edite o status e selecione o ícone de marca de seleção.

O status editado agora está visível no seu quadro.

## Para mover um status

1. No painel de navegação, selecione Problemas.
2. Escolha o ícone de reticências e selecione Configurações.
3. Em Status, escolha um status que você deseja mover.
4. Arraste e solte o status onde você quiser.

### Note

Você só pode mover um status dentro da categoria designada.

Os status agora estão reordenados no quadro.

## Como desativar um status

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Status, escolha um status que você deseja desativar.
4. No status que você deseja desativar, escolha o status. O status agora está cinza.

**Note**

O status desativado aparece no quadro até que todos os problemas sejam retirados dele. Os problemas não podem ser adicionados a um status desativado.

5. Para reativar um status desativado, escolha a opção de ativar o status. O status não está mais cinza.

**Note**

Deve haver pelo menos um status ativo em cada categoria. Se houver apenas um status na categoria, você não poderá desativá-lo.

## Configuração da estimativa do esforço do problema

Siga estas etapas para definir a configuração das estimativas de esforço para problemas em CodeCatalyst.

Para configurar a estimativa de esforço para problemas

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Estimativa, na seção Configurações básicas, escolha como os valores da estimativa serão exibidos. Os tipos de estimativas disponíveis são Tamanho de camisetas, Sequenciamento de Fibonacci ou Ocultar estimativas. Quando o tipo de estimativa for atualizado, nenhum dado será perdido e o valor estimado de todos os problemas será convertido automaticamente. O mapeamento da conversão é mostrado na tabela a seguir.

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| PP                  | 1                      |
| PP                  | 2                      |
| S                   | 3                      |

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| M                   | 5                      |
| L                   | 8                      |
| GG                  | 13                     |

## Ativação ou desativação de vários responsáveis

Siga estas etapas para definir a configuração de vários destinatários para problemas em CodeCatalyst

Para habilitar ou desabilitar vários responsáveis

1. No painel de navegação, selecione Problemas.
2. Selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione Configurações.
3. Em Responsável, na seção Configurações básicas, alterne o indicador para permitir que vários responsáveis sejam atribuídos ao mesmo problema. Um problema tem até 10 responsáveis. Se você não habilitar essa opção, só poderá atribuir um responsável a um problema.

## Criação de uma visualização de problemas

Você pode criar [visualizações](#) para visualizar rapidamente problemas que correspondam a um determinado conjunto de filtros. Isso pode ajudar você a economizar tempo e a visualizar rapidamente os problemas que filtrou, agrupou ou classificou anteriormente.

Para criar uma visualização de problemas

1. No painel de navegação, selecione Problemas.
2. (Opcional) Dependendo do caso de uso, você pode criar uma visualização a partir de uma visualização existente. Para navegar até uma visualização diferente, selecione Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e selecione a visualização.

3. (Opcional) Configure filtros, agrupamento e classificação antes de criar a visualização. Você pode adicioná-los ao criar uma visualização, mas se fizer isso antes, poderá visualizar o que é mostrado na visualização antes de criá-la.
4. Abra o menu suspenso do alternador de visualização de problemas na barra de cabeçalho. Para criar uma visualização do quadro em que os problemas são visualizados em colunas com base no status, selecione + na coluna Quadro. Para criar uma visualização em grade em que os problemas são visualizados em uma lista, selecione + na coluna Grade. Você pode alterar o tipo de visualização antes de criá-la, caso mude de ideia.
5. Na caixa de diálogo Criar visualização, insira um Nome para a visualização.
6. Os campos Filtrar, Agrupar problemas por e Classificar problemas por são preenchidos com base nas configurações da visualização atual. Atualize-o, se necessário.
7. Selecione Criar visualização para criar a visualização e acessá-la.

## Cotas para edições em CodeCatalyst

A tabela a seguir descreve cotas e limites para problemas na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte.. [Cotas para o CodeCatalyst](#)

| Recurso                                                          | Cota padrão                                                                                                                                                                                            |
|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Problemas ativos                                                 | Máximo de 1.000 por projeto.                                                                                                                                                                           |
| Tamanho do anexo                                                 | Máximo de 500 MB por anexo.<br><br>O armazenamento total máximo de anexos é afetado pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte <a href="#">Preço</a> . |
| Número total de problemas (ativos e arquivados)                  | Máximo de 100.000 por projeto.                                                                                                                                                                         |
| Visualizações salvas                                             | Máximo de 50 visualizações de problema salvas por projeto.                                                                                                                                             |
| Número de solicitações pull que você pode vincular a um problema | Máximo de 50 solicitações pull por problema.                                                                                                                                                           |

| Recurso                              | Cota padrão                   |
|--------------------------------------|-------------------------------|
| Status (por projeto)                 | Máximo de 50 por projeto.     |
| Status (por problema)                | Máximo de 50 por problema.    |
| Rótulos (por projeto)                | Máximo de 200 por projeto.    |
| Rótulos (por problema)               | Máximo de 50 por problema.    |
| Campos personalizados (por problema) | Máximo de 50 por problema.    |
| Responsáveis                         | Máximo de 10 por problema.    |
| Comentários                          | Máximo de 1.000 por problema. |
| Tarefas                              | Máximo de 100 por problema.   |

# Configure identidade, permissões e acesso no CodeCatalyst

Ao entrar na Amazon CodeCatalyst pela primeira vez, você cria uma ID de AWS construtor. AWS O construtor IDs não existe em AWS Identity and Access Management. O nome de usuário que você escolhe no primeiro login se torna o ID de usuário exclusivo para sua identidade.

Ao entrar CodeCatalyst, você pode fazer login pela primeira vez de duas maneiras:

- Como parte da criação de um espaço.
- Como parte da aceitação de um convite para um projeto ou espaço em CodeCatalyst.

A função ou funções associadas à sua identidade determinam as ações que você pode realizar CodeCatalyst. Os perfis do projeto, como Administrador do projeto e Colaborador, são específicos de um projeto, então você pode ter um perfil em um projeto e um perfil diferente em outro projeto. Se você criar um espaço, CodeCatalyst atribuirá automaticamente a função de administrador do espaço. Quando os usuários aceitam convites para um projeto, CodeCatalyst adiciona essas identidades ao espaço e atribui a eles a função de acesso limitado. Ao convidar usuários para projetos, você escolhe o perfil que deseja que eles tenham no projeto, o que determina quais ações eles podem e não podem realizar dentro do projeto. A maioria dos usuários que trabalham em um projeto só precisa do perfil de Colaborador para realizar as tarefas. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).

Além da função do projeto, os usuários de um projeto precisam de um token de acesso pessoal (PAT) para acessar os repositórios de origem de um projeto ao usar clientes Git ou ambientes de desenvolvimento integrados (). IDEs Os membros do projeto podem usar esse PAT com aplicativos de terceiros como uma senha específica do aplicativo associada à sua identidade. CodeCatalyst Por exemplo, ao clonar um repositório de origem em um computador local, você deve fornecer um PAT e seu CodeCatalyst nome de usuário.

Você pode configurar o acesso entre CodeCatalyst e AWS recursos usando uma [função de serviço](#) para realizar ações como acessar AWS CloudFormation pilhas e recursos ao implantar ações em fluxos de trabalho. Você deve configurar o acesso entre CodeCatalyst e AWS os recursos para que as ações do fluxo de trabalho incluídas nos modelos de projeto sejam executadas.

## Tópicos

- [Concessão de acesso com perfis de usuário](#)
- [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#)

- [Acessando GitHub recursos com conexões pessoais](#)
- [Configure seu AWS Builder ID para fazer login com autenticação multifator \(MFA\)](#)
- [Segurança na Amazon CodeCatalyst](#)
- [Monitoramento de eventos e chamadas de API usando o registro](#)
- [Cotas para identidade, permissão e acesso em CodeCatalyst](#)
- [Solução de problemas](#)

## Concessão de acesso com perfis de usuário

Na Amazon CodeCatalyst, você pode atribuir funções aos usuários tanto no nível do projeto quanto no nível do espaço. Em um projeto, um perfil especifica o que um usuário tem permissão para fazer em um projeto com os recursos desse projeto. Os usuários ganham participação em um espaço quando participam de um projeto. Você pode adicionar ou remover usuários como administradores de um espaço. A função de administrador do Space tem as permissões mais amplas de qualquer função em CodeCatalyst. Como prática recomendada, atribua aos usuários as permissões mais restritas necessárias para realizar os trabalhos.

Você pode atribuir perfis aos usuários no espaço. Você também pode atribuir perfis aos usuários nos projetos em que eles são membros. Cada usuário só pode ter um perfil em um projeto ou espaço, mas os usuários podem ter perfis diferentes em cada projeto e espaço. Por exemplo, um usuário pode ter o perfil de administrador do projeto em um projeto e o perfil de colaborador em outro projeto.

### Tópicos

- [Noções básicas sobre perfis de usuário em espaços e projetos](#)
- [Visualização das permissões disponíveis para cada perfil](#)
- [Visualização e alteração de perfis do usuário](#)

## Noções básicas sobre perfis de usuário em espaços e projetos

Há três perfis disponíveis para um espaço:

- Administrador do espaço
- Usuário avançado
- Acesso limitado

Os usuários que aceitam um convite para um projeto têm perfil de Acesso limitado automaticamente atribuído a eles no espaço que contém o projeto.

Há quatro perfis disponíveis para membros em um projeto:

- Administrador do projeto
- Contributor (Colaborador)
- Revisor
- Somente leitura

Quando você adiciona um usuário a um projeto, concede CodeCatalyst automaticamente a ele a função de acesso limitado. Se você remover um usuário de todos os projetos, removerá CodeCatalyst automaticamente a função de acesso limitado desse usuário.

## Perfil de administrador do espaço

A função de administrador do Space é a função mais poderosa em CodeCatalyst. Atribua a função de administrador do espaço somente aos usuários que precisam administrar todos os aspectos de um espaço, pois essa função tem todas as permissões em CodeCatalyst. Os usuários com o perfil de Administrador do espaço são os únicos usuários que podem adicionar ou remover outros usuários do perfil de Administrador do espaço e excluir o espaço.

Quando você cria um espaço, atribui CodeCatalyst automaticamente a função de administrador do espaço. Como prática recomendada, recomendamos que você adicione esse perfil a pelo menos um outro usuário que possa atuar nesse perfil caso o criador do espaço original não esteja disponível.

## Perfil de usuário avançado

A função de usuário avançado é a segunda função mais poderosa em CodeCatalyst espaços, mas não tem acesso a projetos em um espaço. Ele foi projetado para usuários que precisam criar projetos em um espaço e ajudar a gerenciar os usuários e recursos do espaço. Atribua o perfil de Usuário avançado a usuários que são líderes de equipe ou gerentes que precisam criar projetos e gerenciar usuários no espaço como parte de seu trabalho.

## Perfil de acesso limitado

A função de acesso limitado é a função que a maioria dos usuários terá nos CodeCatalyst espaços. É o perfil atribuído automaticamente aos usuários quando eles aceitam um convite para um projeto em um espaço. Ele fornece as permissões limitadas necessárias para trabalhar no espaço que



contém esse projeto. Atribua o perfil de Acesso limitado aos usuários que você convida diretamente para o espaço, a menos que o trabalho deles exija o gerenciamento de algum aspecto do espaço.

## Perfil de administrador do projeto

A função de administrador do projeto é a função mais poderosa em um CodeCatalyst projeto. Atribua esse perfil somente aos usuários que precisam administrar todos os aspectos de um projeto, incluindo a edição das configurações do projeto, o gerenciamento de permissões do projeto e a exclusão de projetos.

Os perfis do projeto não têm nenhuma permissão no nível do espaço. Portanto, os usuários com o perfil de Administrador do projeto não podem criar projetos adicionais. Somente usuários com o perfil de Administrador do espaço ou Usuário avançado podem criar projetos.

### Note

A função de administrador do Space tem todas as permissões em CodeCatalyst.

## Perfil de colaborador

A função de Colaborador é destinada à maioria dos membros de um CodeCatalyst projeto. Atribua esse perfil aos usuários que precisam trabalhar com código, fluxos de trabalho, problemas e ações em um projeto.

## Perfil de revisor

A função de revisor é destinada a usuários que precisam ser capazes de interagir com recursos em um projeto, como pull requests e problemas, mas não criar e mesclar código, criar fluxos de trabalho ou iniciar ou interromper a execução do fluxo de trabalho em um projeto. CodeCatalyst atribua o perfil de Revisor aos usuários que precisam aprovar e comentar sobre solicitações pull, criar, atualizar, resolver e comentar sobre problemas, além de visualizar o código e os fluxos de trabalho em um projeto.

## Perfil somente leitura

O perfil Somente leitura é destinado a usuários que precisam visualizar os recursos e o status dos recursos, mas não interagir com eles ou contribuir diretamente com o projeto. Os usuários com essa função não podem criar recursos no CodeCatalyst, mas podem visualizá-los e copiá-los, como

clonar repositórios e baixar anexos de problemas em um computador local. Atribua o perfil Somente leitura aos usuários que precisam visualizar os recursos e o estado do projeto, mas não interagir diretamente com ele.

## Visualização das permissões disponíveis para cada perfil

A tabela a seguir mostra as permissões disponíveis para cada CodeCatalyst função. Use os links para acessar o conjunto apropriado de permissões.

- [Space permissions](#)
- [Extensions permissions](#)
- [Project permissions](#)
- [Source repository permissions](#)
- [Dev Environment permissions](#)
- [Package repository and package permissions](#)
- [Workflow permissions](#)
- [Issues permissions](#)
- [Blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)

| Permissão | Perfil de administrador do espaço | Perfil de usuário avançado | Perfil de acesso limitado | Perfil de administrador do projeto | Perfil de colaborador | Perfil de revisor | Perfil somente leitura |
|-----------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|------------------------|
|-----------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|------------------------|

### Permissões de espaço

|                              |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                       |                                                                                       |                                                                                       |
|------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar espaço                 |  |  |  |  |  |  |  |
| Editar detalhes de faturamen |  |  |  |  |  |  |  |

| Permissão                                            | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| to do espaço                                         |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                       |                                                                                       |                                                                                       |
| Configurar e ativar logon único                      |    |    |    |    |    |    |    |
| Remover logon único                                  |    |    |    |    |    |    |    |
| Habilitar recursos de IA generativa para um espaço   |    |    |    |    |    |    |    |
| Desabilitar recursos de IA generativa para um espaço |  |  |  |  |  |  |  |
| Excluir espaço                                       |  |  |  |  |  |  |  |

| Permissão                                                      | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Adicionar outros usuários ao perfil de administrador do espaço |    |    |    |    |    |    |    |
| Remover outros usuários do perfil de administrador do espaço   |    |    |    |    |    |    |    |
| Criar equipe                                                   |  |  |  |  |  |  |  |
| Excluir equipe                                                 |  |  |  |  |  |  |  |
| Atualizar equipe                                               |  |  |  |  |  |  |  |

| Permissão                                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Desabilitar os recursos da máquina para o espaço |    |    |    |    |    |    |    |
| Habilitar os recursos da máquina para o espaço   |    |    |    |    |    |    |    |
| Criar um projeto                                 |  |  |  |  |  |  |  |
| Associar uma conexão de conta da AWS ao espaço   |  |  |  |  |  |  |  |
| Atualizar uma conexão de conta da AWS            |  |  |  |  |  |  |  |

| Permissão                                                                | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Desassociar uma conexão de conta da AWS do espaço                        |    |    |    |    |    |    |    |
| Excluir uma conexão de conta da AWS e remover do espaço                  |    |    |    |    |    |    |    |
| Habilitar conexões de contas restritas a projetos no espaço <sup>1</sup> |  |  |  |  |  |  |  |

| Permissão                                                                  | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Desabilitar conexões de contas restritas a projetos no espaço <sup>2</sup> |    |    |    |    |    |    |    |
| Convidar outras pessoas para o espaço                                      |    |    |    |    |    |    |    |
| Criar conexão VPC                                                          |  |  |  |  |  |  |  |
| Editar conexão VPC                                                         |  |  |  |  |  |  |  |
| Excluir conexão VPC                                                        |  |  |  |  |  |  |  |
| Visualizar logs de atividade no espaço                                     |  |  |  |  |  |  |  |

| Permissão                           | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar conexões de conta da AWS |    |    |    |    |    |    |    |
| Exibir incidentes para CodeCatalyst |    |    |    |    |    |    |    |
| Visualizar espaço                   |    |    |    |    |    |    |    |
| Visualizar equipes                  |   |   |   |   |   |   |   |
| Visualizar conexões VPC             |  |  |  |  |  |  |  |

<sup>1</sup> Com o perfil de Usuário avançado, você pode ativar restrições de projeto para uma conta, mas só pode configurar o acesso para projetos dos quais é membro.

<sup>2</sup> Com o perfil de Usuário avançado, você pode desativar restrições de projeto para uma conta, mas só pode configurar o acesso para projetos dos quais é membro.

| Permissões de extensões | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Instalar extensões      |  |  |  |  |  |  |  |



| Permissão                                                    | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Atualizar extensões                                          |    |    |    |    |    |    |    |
| Excluir extensões                                            |    |    |    |    |    |    |    |
| Conectar uma GitHub conta                                    |    |    |    |    |    |    |    |
| Desconectar uma conta GitHub                                 |    |    |    |    |    |    |    |
| Conectar um site do Jira                                     |  |  |  |  |  |  |  |
| Desconectar um site do Jira                                  |  |  |  |  |  |  |  |
| Visualizar detalhes da configuração das extensões instaladas |  |  |  |  |  |  |  |

| Permissão                                         | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|---------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar extensões                              |    |    |    |    |    |    |    |
| Permissões do projeto                             | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
| Editar configurações do projeto                   |    |    |    |    |    |    |    |
| Desabilitar os recursos da máquina para o projeto |   |   |   |   |   |   |   |
| Habilitar os recursos da máquina para o projeto   |  |  |  |  |  |  |  |
| Excluir projeto                                   |  |  |  |  |  |  |  |

| Permissão                                 | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Convidar usuários para um projeto         |    |    |    |    |    |    |    |
| Alterar perfis dos usuários em um projeto |    |    |    |    |    |    |    |
| Remover usuários de um projeto            |    |    |    |    |    |    |    |
| Adicionar equipe a um projeto             |  |  |  |  |  |  |  |
| Remover equipe de um projeto              |  |  |  |  |  |  |  |
| Alterar o perfil da equipe no projeto     |  |  |  |  |  |  |  |

| Permissão                           | Perfil de administrador do espaço | Perfil de usuário avançado | Perfil de acesso limitado | Perfil de administrador do projeto | Perfil de colaborador | Perfil de revisor | Perfil somente leitura |
|-------------------------------------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|------------------------|
| Visualizar projeto                  |                                   |                            |                           |                                    |                       |                   |                        |
| Visualizar atividade do projeto     |                                   |                            |                           |                                    |                       |                   |                        |
| Visualizar equipes no projeto       |                                   |                            |                           |                                    |                       |                   |                        |
| Visualizar esquemas                 |                                   |                            |                           |                                    |                       |                   |                        |
| Permissões do repositório de origem | Perfil de administrador do espaço | Perfil de usuário avançado | Perfil de acesso limitado | Perfil de administrador do projeto | Perfil de colaborador | Perfil de revisor | Perfil somente leitura |
| Criar repositórios                  |                                   |                            |                           |                                    |                       |                   |                        |
| Vincular repositórios               |                                   |                            |                           |                                    |                       |                   |                        |

| Permissão                               | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Desvincular repositórios                |    |    |    |    |    |    |    |
| Excluir repositórios                    |    |    |    |    |    |    |    |
| Editar configurações de repositório     |    |    |    |    |    |    |    |
| Visualizar repositórios                 |  |  |  |  |  |  |  |
| Visualizar configurações de repositório |  |  |  |  |  |  |  |
| Clonar repositórios                     |  |  |  |  |  |  |  |
| Criar ramificações                      |  |  |  |  |  |  |  |

| Permissão                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Crie regras de ramificação       |    |    |    |    |    |    |    |
| Alterar ramificação padrão       |    |    |    |    |    |    |    |
| Excluir ramificações             |    |    |    |    |    |    |    |
| Mesclar ramificações             |   |   |   |   |   |   |   |
| Atualizar regras de ramificação  |  |  |  |  |  |  |  |
| Visualizar ramificações          |  |  |  |  |  |  |  |
| Visualizar regras de ramificação |  |  |  |  |  |  |  |

| Permissão                   | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar pastas                |    |    |    |    |    |    |    |
| Excluir pastas              |    |    |    |    |    |    |    |
| Editar pastas               |    |    |    |    |    |    |    |
| Visualizar pastas           |    |    |    |    |    |    |    |
| Criar arquivos              |    |    |    |    |    |    |    |
| Excluir arquivos            |  |  |  |  |  |  |  |
| Editar arquivos             |  |  |  |  |  |  |  |
| Exibir arquivos             |  |  |  |  |  |  |  |
| Criar e enviar confirmações |  |  |  |  |  |  |  |
| Visualizar confirmações     |  |  |  |  |  |  |  |

| Permissão                                                 | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar solicitações pull                                   |    |    |    |    |    |    |    |
| Criar regras de aprovação de solicitações pull            |    |    |    |    |    |    |    |
| Substituir requisitos de mesclagem para solicitações pull |    |    |    |    |    |    |    |
| Atualizar solicitações pull                               |  |  |  |  |  |  |  |
| Atualizar regras de aprovação de solicitações pull        |  |  |  |  |  |  |  |
| Visualizar solicitações pull                              |  |  |  |  |  |  |  |



| Permissão                                                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar regras de aprovação de solicitações pull              |    |    |    |    |    |    |    |
| Fechar solicitações pull                                         |    |    |    |    |    |    |    |
| Aprovar solicitações pull                                        |    |    |    |    |    |    |    |
| Comentário sobre solicitações pull                               |  |  |  |  |  |  |  |
| Interagir com o Amazon Q nos comentários sobre solicitações pull |  |  |  |  |  |  |  |

| Permissão                                                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar uma revisão para uma solicitação pull criada pelo Amazon Q |    |    |    |    |    |    |    |
| Vincular um problema a uma solicitação pull                      |    |    |    |    |    |    |    |
| Desvincular um problema de uma solicitação pull                  |  |  |  |  |  |  |  |
| Permissões do Ambiente de Desenvolvimento                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |

| Permissão                                                         | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar seu próprio Ambiente de Desenvolvimento                     |    |    |    |    |    |    |    |
| Parar seu próprio Ambiente de Desenvolvimento                     |    |    |    |    |    |    |    |
| Parar os ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |
| Retomar seu próprio Ambiente de Desenvolvimento                   |  |  |  |  |  |  |  |

| Permissão                                                              | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar seus próprios ambientes de desenvolvimento                  |    |    |    |    |    |    |    |
| Visualizar os ambientes de desenvolvimento criados por outros usuários |    |    |    |    |    |    |    |
| Editar seu próprio Ambiente de Desenvolvimento                         |  |  |  |  |  |  |  |

| Permissão                                                           | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|---------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Editar os ambientes de desenvolvimento criados por outros usuários  |    |    |    |    |    |    |    |
| Excluir seu próprio Ambiente de Desenvolvimento                     |    |    |    |    |    |    |    |
| Excluir os ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |

| Permissão                                            | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar um devfile para ambientes de desenvolvimento   |    |    |    |    |    |    |    |
| Editar um devfile para ambientes de desenvolvimento  |    |    |    |    |    |    |    |
| Excluir um devfile para ambientes de desenvolvimento |  |  |  |  |  |  |  |

| Permissão                                               | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|---------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar um devfile para ambientes de desenvolvimento |    |    |    |    |    |    |    |
| Repositório de pacotes e permissões de pacotes          | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
| Criar repositório de pacotes                            |  |  |  |  |  |  |  |
| Visualizar repositórios de pacotes                      |  |  |  |  |  |  |  |
| Editar repositório de pacotes                           |  |  |  |  |  |  |  |

| Permissão                                 | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Excluir repositório de pacotes            |    |    |    |    |    |    |    |
| Criar repositório de pacotes de gateway   |    |    |    |    |    |    |    |
| Exibir repositórios de pacotes de gateway |    |    |    |    |    |    |    |
| Excluir repositório de pacotes de gateway |  |  |  |  |  |  |  |
| Adicionar repositório de pacotes upstream |  |  |  |  |  |  |  |



| Permissão                                            | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Editar a ordem de pesquisa dos repositórios upstream |    |    |    |    |    |    |    |
| Remover repositório de pacotes upstream              |    |    |    |    |    |    |    |
| Conectar a um repositório de pacotes                 |  |  |  |  |  |  |  |
| Ler pacotes de um repositório de pacotes             |  |  |  |  |  |  |  |

| Permissão                                      | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Publicar pacotes em um repositório de pacotes  |    |    |    |    |    |    |    |
| Ler e reter pacotes de um repositório upstream |    |    |    |    |    |    |    |
| Visualizar pacotes                             |  |  |  |  |  |  |  |
| Visualizar versões de pacotes                  |  |  |  |  |  |  |  |
| Visualizar ativos da versão do pacote          |  |  |  |  |  |  |  |

| Permissão                                  | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Listar dependências de versão de pacote    |    |    |    |    |    |    |    |
| Atualizar o status da versão do pacote     |    |    |    |    |    |    |    |
| Atualizar configuração de origem do pacote |  |  |  |  |  |  |  |
| Excluir versão de pacote                   |  |  |  |  |  |  |  |
| Permissões de fluxo de trabalho            | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |

| Permissão                               | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar fluxo de trabalho                 |    |    |    |    |    |    |    |
| Atualizar fluxo de trabalho             |    |    |    |    |    |    |    |
| Excluir fluxo de trabalho               |    |    |    |    |    |    |    |
| Iniciar fluxo de trabalho               |    |    |    |    |    |    |    |
| Parar fluxo de trabalho                 |  |  |  |  |  |  |  |
| Criar segredos de fluxo de trabalho     |  |  |  |  |  |  |  |
| Atualizar segredos de fluxo de trabalho |  |  |  |  |  |  |  |

| Permissão                                                | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Excluir segredos de fluxo de trabalho                    |    |    |    |    |    |    |    |
| Criar ambientes                                          |    |    |    |    |    |    |    |
| Excluir ambientes                                        |    |    |    |    |    |    |    |
| Criar frota                                              |    |    |    |    |    |    |    |
| Atualizar frota                                          |  |  |  |  |  |  |  |
| Excluir frota                                            |  |  |  |  |  |  |  |
| Gerenciar recursos de fluxo de trabalho em outras contas |  |  |  |  |  |  |  |

| Permissão                                       | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Associar uma Conta da AWS conexão a um ambiente |    |    |    |    |    |    |    |
| Associar um perfil padrão do IAM a um ambiente  |    |    |    |    |    |    |    |
| Associar uma conexão VPC a um ambiente          |  |  |  |  |  |  |  |
| Desassociar uma conexão VPC a um ambiente       |  |  |  |  |  |  |  |

| Permissão                                                      | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Associe um ambiente conectado à VPC a um fluxo de trabalho     |    |    |    |    |    |    |    |
| Desassociar um ambiente conectado à VPC a um fluxo de trabalho |    |    |    |    |    |    |    |
| Aprovar execuções de fluxo de trabalho                         |  |  |  |  |  |  |  |
| Rastrear uma confirmação em um fluxo de trabalho               |  |  |  |  |  |  |  |

| Permissão                                              | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar ambientes                                   |    |    |    |    |    |    |    |
| Visualizar logs de ações de criação                    |    |    |    |    |    |    |    |
| Visualizar frotas                                      |    |    |    |    |    |    |    |
| Visualizar logs de ações de teste                      |    |    |    |    |    |    |    |
| Visualizar fluxo de trabalho                           |  |  |  |  |  |  |  |
| Visualizar execuções de fluxo de trabalho              |  |  |  |  |  |  |  |
| Visualizar resultados de execução de fluxo de trabalho |  |  |  |  |  |  |  |



| Permissão                                | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar segredos do fluxo de trabalho |    |    |    |    |    |    |    |
| Permissões de problemas                  | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
| Criar problema                           |    |    |    |    |    |    |    |
| Atualizar problema                       |   |   |   |   |   |   |   |
| Visualizar problemas                     |  |  |  |  |  |  |  |
| Criar tarefa                             |  |  |  |  |  |  |  |
| Atualizar tarefa                         |  |  |  |  |  |  |  |
| Visualizar tarefas                       |  |  |  |  |  |  |  |
| Arquivar um problema                     |  |  |  |  |  |  |  |

| Permissão                                                  | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Atribuir um problema ao Amazon Q                           |    |    |    |    |    |    |    |
| Interagir com o Amazon Q nos comentários sobre um problema |    |    |    |    |    |    |    |
| Cancelar a atribuição do Amazon Q a um problema            |  |  |  |  |  |  |  |
| Recomendar tarefas para um problema com o Amazon Q         |  |  |  |  |  |  |  |

| Permissão                                       | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar tarefas recomendadas pelo Amazon Q        |    |    |    |    |    |    |    |
| Atualizar problemas criados por outros usuários |    |    |    |    |    |    |    |
| Visualizar comentários sobre um problema        |  |  |  |  |  |  |  |
| Criar um comentário sobre um problema           |  |  |  |  |  |  |  |
| Atualizar um comentário sobre um problema       |  |  |  |  |  |  |  |

| Permissão                                    | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Criar um rótulo                              |    |    |    |    |    |    |    |
| Atualizar um rótulo                          |    |    |    |    |    |    |    |
| Visualizar rótulos                           |    |    |    |    |    |    |    |
| Adicionar um rótulo a um problema            |    |    |    |    |    |    |    |
| Remover um rótulo de um problema             |   |   |   |   |   |   |   |
| Criar um status personalizado para problemas |  |  |  |  |  |  |  |
| Atualizar um status personalizado            |  |  |  |  |  |  |  |

| Permissão                          | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar um status personalizado |    |    |    |    |    |    |    |
| Mover um status personalizado      |    |    |    |    |    |    |    |
| Desativar um status personalizado  |    |    |    |    |    |    |    |
| Adicionar um anexo a um problema   |  |  |  |  |  |  |  |
| Visualizar um anexo de problema    |  |  |  |  |  |  |  |
| Remover um anexo de um problema    |  |  |  |  |  |  |  |

| Permissão                                   | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|---------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Vincular um problema a outro                |    |    |    |    |    |    |    |
| Desvincular um problema de outro problema   |    |    |    |    |    |    |    |
| Atualizar um link de problema               |    |    |    |    |    |    |    |
| Visualizar links para um problema           |  |  |  |  |  |  |  |
| Vincular uma solicitação pull a um problema |  |  |  |  |  |  |  |

| Permissão                                       | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Desvincular uma solicitação pull de um problema |    |    |    |    |    |    |    |
| Vincular um projeto do Jira                     |    |    |    |    |    |    |    |
| Desvincular um projeto do Jira                  |    |    |    |    |    |    |    |
| Permissões de esquema                           | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
| Criar projeto de esquema personalizado          |  |  |  |  |  |  |  |

| Permissão                                                              | Perfil de administrador do espaço                                                  | Perfil de usuário avançado                                                         | Perfil de acesso limitado                                                          | Perfil de administrador do projeto                                                 | Perfil de colaborador                                                                | Perfil de revisor                                                                    | Perfil somente leitura                                                               |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Publicar uma prévia do esquema personalizado                           |   |   |   |   |   |   |   |
| Publicar um esquema personalizado                                      |   |   |   |   |   |   |   |
| Adicionar um esquema personalizado a um catálogo de esquemas de espaço |  |  |  |  |  |  |  |



| Permissão                                                             | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Remover um esquema personalizado de um catálogo de esquemas de espaço |    |    |    |    |    |    |    |
| Gerenciar as permissões de publicação para um esquema personalizado   |    |    |    |    |    |    |    |
| Gerenciar a versão do catálogo para um esquema personalizado          |  |  |  |  |  |  |  |

| Permissão                                                      | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Atualizar um esquema personalizado                             |    |    |    |    |    |    |    |
| Excluir uma versão do esquema personalizado                    |    |    |    |    |    |    |    |
| Excluir um esquema personalizado                               |  |  |  |  |  |  |  |
| Converter um repositório de origem em um esquema personalizado |  |  |  |  |  |  |  |

| Permissão                                               | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|---------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Adicionar um esquema personalizado a um projeto         |    |    |    |    |    |    |    |
| Desassociar um esquema personalizado de um projeto      |    |    |    |    |    |    |    |
| Atualizar a versão de um esquema personalizado aplicado |  |  |  |  |  |  |  |
| Editar as configurações de um esquema personalizado     |  |  |  |  |  |  |  |

| Permissão                                        | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|--------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar os esquemas personalizados publicados |    |    |    |    |    |    |    |
| Permissões de notificações                       |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                       |                                                                                       |                                                                                       |
| Configurar canal de notificação                  |   |   |   |   |   |   |   |
| Remover canal de notificação                     |  |  |  |  |  |  |  |
| Editar configurações de notificação              |  |  |  |  |  |  |  |

| Permissão                                                             | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Visualizar configurações de notificação                               |    |    |    |    |    |    |    |
| Receba notificações automáticas sobre CodeCatalyst incidentes         |    |    |    |    |    |    |    |
| Configurar notificações por e-mail para sua conta de e-mail associada |  |  |  |  |  |  |  |
| Permissões de pesquisa                                                | Perfil de administrador do espaço                                                   | Perfil de usuário avançado                                                          | Perfil de acesso limitado                                                           | Perfil de administrador do projeto                                                  | Perfil de colaborador                                                                 | Perfil de revisor                                                                     | Perfil somente leitura                                                                |

| Permissão                      | Perfil de administrador do espaço                                                 | Perfil de usuário avançado                                                        | Perfil de acesso limitado                                                         | Perfil de administrador do projeto                                                | Perfil de colaborador                                                               | Perfil de revisor                                                                   | Perfil somente leitura                                                              |
|--------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Pesquisar dentro de um projeto |  |  |  |  |  |  |  |
| Pesquisar em todo o espaço     |  |  |  |  |  |  |  |

## Visualização e alteração de perfis do usuário

Você pode visualizar o perfil atribuído aos usuários. Isso ajuda você a entender quais ações eles podem realizar em um projeto. Você também pode alterar o perfil deles se precisarem de permissões adicionais.

Para visualizar o perfil de um usuário em um projeto

1. Navegue até o projeto em que deseja visualizar os perfis associados a cada membro do projeto.

### Tip

Você pode escolher qual projeto deseja visualizar na barra de navegação superior.

2. No painel de navegação, escolha Configurações do projeto.
3. Na guia Membros, o perfil de cada membro do projeto é exibido em Perfil.

Para alterar perfis dos usuários em um projeto

1. Navegue até o projeto em que deseja alterar os perfis associados aos membros do projeto.

 Tip

Você pode escolher qual projeto deseja visualizar na barra de navegação superior.

2. No painel de navegação, escolha Configurações do projeto.
3. Na guia Membros, em Membros do projeto, escolha os usuários cujos perfis você deseja alterar. Escolha Ações e, depois, escolha Editar perfil.
4. Em Perfil, escolha o perfil do projeto e, depois, selecione Confirmar.

## Visualização e alteração de perfis no espaço

Todos os usuários que aceitam convites para um projeto CodeCatalyst se tornam membros do espaço do projeto. Você pode ver a lista de membros do espaço. Você pode alterar os perfis dos usuários em Acesso limitado ao Administrador do espaço para gerenciar melhor seu espaço e os recursos. A função de administrador do Space é a única função que permite que os usuários criem projetos no CodeCatalyst.

 Warning

A função de administrador do Space é a função mais poderosa em CodeCatalyst. Os usuários com essa função podem realizar qualquer ação CodeCatalyst, inclusive excluir o espaço. Atribua esse perfil somente aos usuários que precisam desse nível de acesso ao seu espaço. Para obter mais informações, consulte [Perfil de administrador do espaço](#).

Para alterar o perfil de um usuário no espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço.

 Tip

Se você pertencer a mais de um espaço, poderá escolher qual espaço deseja visualizar na barra de navegação superior.

3. Escolha a guia Membros.

4. Escolha o usuário cujo perfil você deseja alterar e selecione Alterar perfil.
5. Em Alterar perfil, escolha o perfil que você deseja atribuir e, depois, selecione Confirmar.

## Conceder aos usuários acesso ao repositório com tokens de acesso pessoal

Para acessar alguns CodeCatalyst recursos, como repositórios de origem, em um computador local com um cliente Git ou ambiente de desenvolvimento integrado (IDE), você deve inserir uma senha específica do aplicativo. Você pode criar um token de acesso pessoal (PAT) para usar com essa finalidade. PATs você cria estão associados à sua identidade de usuário em todos os espaços e projetos em CodeCatalyst. Você pode criar mais de um PAT para sua CodeCatalyst identidade.

Você pode ver os nomes e as datas de expiração dos que PATs você criou e pode excluir aqueles que não são mais necessários. Você só pode copiar o segredo do PAT no momento em que é criado.

### Note

Por padrão, PATs expira em 1 ano.

## Criando PATs

PATs estão associados à sua identidade de usuário em CodeCatalyst. Você só pode copiar um segredo do PAT no momento em que é criado.

### Criando PATs (console)

Você pode usar o console para criar PATs em CodeCatalyst.

Para criar um token de acesso pessoal (console)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.



 Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Tokens de acesso pessoal, escolha Criar.

A página Criar PAT é exibida.

4. Em Nome do PAT, insira um nome descritivo para o seu PAT.
5. Em Data de expiração, mantenha a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de vencimento é padronizada para 1 ano a contar da data atual.
6. Escolha Criar.

 Tip

Você também pode criar esse token ao escolher Clonar repositório para um repositório de origem.

7. Para copiar o segredo do PAT, selecione Copiar. Armazene o segredo do PAT onde você poderá recuperá-lo.

 Important

O segredo do PAT só é exibido uma vez. Não é possível recuperá-lo depois de fechar a janela. Se você não salvou o segredo do PAT em um local seguro, é possível criar outro.

## Criando PATs (CLI)

Você pode usar a CLI para criar PATs em. CodeCatalyst

Para criar um token de acesso pessoal (AWS CLI)

1. No terminal ou na linha de comando, execute o comando `create-access-token` da seguinte maneira.

```
aws codecatalyst create-access-token
```

Se houver êxito, o comando retornará informações sobre o PAT criado como o exemplo a seguir.

```
{
 "secret": "value",
 "name": "marymajor-22222EXAMPLE",
 "expiresTime": "2024-02-04T01:56:04.402000+00:00"
}
```

2.

Você só pode ver o segredo do PAT uma vez — ao criar o PAT. Se você um segredo do PAT ou está preocupado com o fato de ele não estar armazenado com segurança, você pode criar outro.

Você pode ver o PATs associado à sua conta de usuário usando AWS CLI o. Você só pode visualizar informações sobre o PAT, e não o valor do segredo do PAT em si.

#### Note

Verifique se você está usando uma versão recente do AWS CLI para trabalhar CodeCatalyst. As versões anteriores podem não conter os CodeCatalyst comandos. Você deve configurar seu AWS CLI perfil antes de poder usá-lo com CodeCatalyst. Para obter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

## Visualizando PATs

Você pode ver PATs em CodeCatalyst. A lista mostra tudo o PATs que você associou à sua identidade de usuário. Seu PAT está associado ao seu perfil de usuário em todos os espaços e projetos em CodeCatalyst. Os expirados PATs não são exibidos porque são excluídos após expirarem.

### Visualização PATs (console)

Você pode usar o console para ver os PATs associados à sua identidade de usuário em CodeCatalyst.

Para ver seus tokens de acesso pessoais (console)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.

 Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Tokens de acesso pessoal, veja os nomes e as datas de expiração dos seus atuais PATs.

## Visualização PATs (CLI)

Você pode usar a CLI para visualizar os PATs associados à sua identidade de usuário em CodeCatalyst

Para ver seus tokens de acesso pessoais (AWS CLI)

- No terminal ou na linha de comando, execute o comando `list-access-tokens` da seguinte maneira.

```
aws codecatalyst list-access-tokens
```

Se for bem-sucedido, o comando retornará informações sobre o PATs associado à sua conta de usuário, como no exemplo a seguir.

```
{
 "items": [
 {
 "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
 "name": "marymajor-22222EXAMPLE",
 "expiresTime": "2024-02-04T01:56:04.402000+00:00"
 },
 {
 "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
 "name": "marymajor-11111EXAMPLE",
 "expiresTime": "2023-03-12T01:58:40.694000+00:00"
 }
]
}
```

```
]
}
```

## Excluindo PATs

Você pode excluir o PATs associado à sua identidade de usuário em CodeCatalyst.

### Excluindo PATs (console)

Você pode usar o console para excluir PATs CodeCatalyst.

Para excluir um token de acesso pessoal (console)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.

#### Tip

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Tokens de acesso pessoal, escolha o seletor ao lado do PAT que você deseja excluir e, depois, selecione Excluir.

Na página Excluir PAT: <name>?, para confirmar a exclusão, digite delete no campo de texto. Escolha Excluir.

### Excluindo PATs (CLI)

Você pode excluir um PAT associado à sua identidade de usuário usando a AWS CLI. Para fazer isso, você deve fornecer o ID do PAT, que pode ser visualizado usando o comando delete-access-token.

#### Note

Verifique se você está usando uma versão recente do AWS CLI para trabalhar CodeCatalyst. As versões anteriores podem não conter os CodeCatalyst comandos. Para obter mais

informações sobre como usar o AWS CLI com CodeCatalyst, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

Como excluir um token de acesso pessoal (AWS CLI)

- No terminal ou na linha de comando, execute o comando `delete-access-token`, fornecendo o ID do PAT que você deseja excluir. Por exemplo, execute o comando a seguir para excluir um PAT com uma ID de `123EXAMPLE`.

```
aws codecatalyst delete-access-token --id a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb
```

Se houver êxito, o comando não retornará resposta nenhuma.

## Acessando GitHub recursos com conexões pessoais

Você pode usar conexões pessoais para autorizar e conectar seus GitHub recursos de terceiros com CodeCatalyst. Por exemplo, use uma conexão pessoal para autorizar CodeCatalyst o acesso à sua GitHub conta e criar um repositório como fonte para seu projeto ou blueprint. A conexão é mapeada para sua CodeCatalyst identidade e pode ser usada para se conectar a um ou mais repositórios de origem. As conexões que você cria são associadas à sua identidade de usuário em todos os espaços e projetos em CodeCatalyst.

### Note

Você pode gerenciar conexões pessoais com blueprints em GitHub organizações nas quais você tem acesso para fazer isso.

Você pode criar uma conexão pessoal para uma identidade de usuário (CodeCatalyst alias) em todos os espaços, por tipo de provedor.

Você pode usar suas conexões pessoais CodeCatalyst para criar um GitHub repositório para um projeto, escolher um repositório de GitHub origem para um blueprint e gerenciar pull requests em CodeCatalyst seu repositório. GitHub

**Note**

O uso de conexões pessoais para associar blueprints a um GitHub repositório não é o mesmo que o uso de extensões CodeCatalyst para vincular um repositório. Para ter mais informações sobre extensões, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

## Criação de conexões pessoais

Você pode usar o console para criar uma conexão pessoal associada à sua identidade de usuário no CodeCatalyst.

Para criar uma conexão pessoal

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.

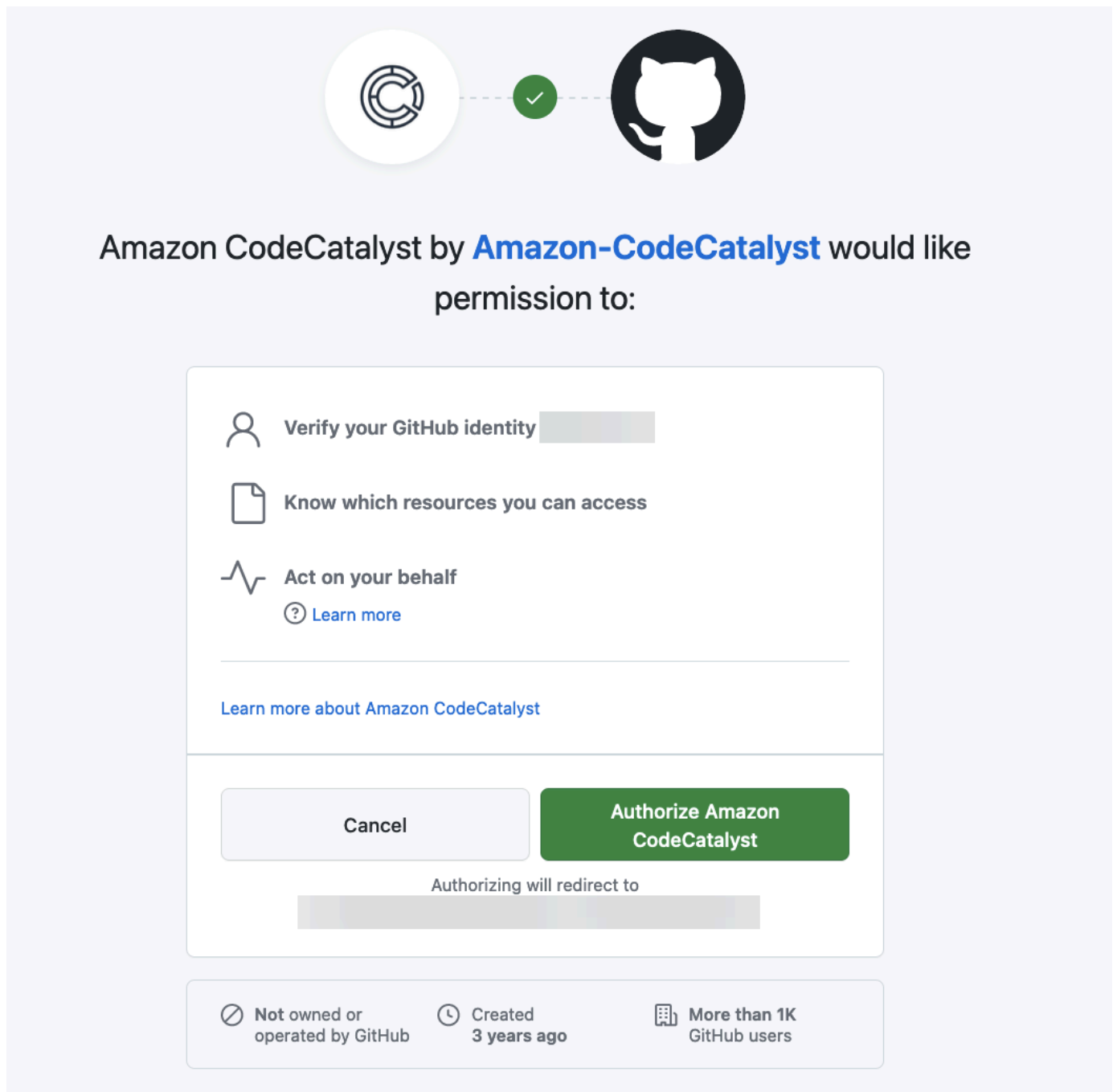
**Tip**

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Conexões pessoais, selecione Criar.

A página Criar conexão é exibida.

4. Escolha Criar. Uma página Criar conexão é exibida.
5. Na página Criar conexão, em Provedor, escolha GitHub. Em Nome da conexão, digite um nome para a conexão. Escolha Criar.
6. Se solicitado, faça login na sua GitHub conta.
7. Na página de confirmação da conexão, selecione Aceitar.
8. Na página de confirmação da instalação, escolha o botão de autorização para confirmar que deseja instalar a aplicação de conexão.



## Excluir conexões pessoais

Você pode excluir uma conexão pessoal associada à sua identidade de usuário em CodeCatalyst.

**Note**

A exclusão da conexão pessoal em CodeCatalyst não desinstala o aplicativo em sua GitHub conta. Se você criar uma conexão pessoal, a instalação da aplicação poderá ser usada. Para desinstalar o aplicativo em GitHub, você pode revogá-lo e reinstalá-lo posteriormente.

Para excluir uma conexão pessoal no CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.

**Tip**

Você também pode encontrar seu perfil de usuário acessando a página de membros de um projeto ou espaço e escolhendo seu nome na lista de membros.

3. Em Conexões pessoais, escolha o seletor ao lado da conexão que você deseja excluir e, depois, selecione Excluir.

Na página Excluir conexão: <name>?, para confirmar a exclusão, digite delete no campo de texto. Escolha Excluir.

1. Faça login GitHub e navegue até as configurações da sua conta para os aplicativos instalados. Escolha o ícone do seu perfil, selecione Configurações e, depois, escolha Aplicações.
2. Na guia GitHub Aplicativos autorizados, na lista de aplicativos autorizados, veja o aplicativo instalado para CodeCatalyst. Para revogar a instalação, selecione Revogar.

## Configure seu AWS Builder ID para fazer login com autenticação multifator (MFA)

Se você criou seu perfil de AWS Builder ID para uso pessoal ou profissional, recomendamos configurar a autenticação multifator (MFA) como outra camada de segurança. Recomendamos especialmente configurar o MFA se você for membro de um espaço e colaborar com outras pessoas



em projetos. Como mais de uma pessoa pode ter acesso a um projeto, existem mais oportunidades para violações de segurança.

Ao habilitar o MFA, você deve entrar na Amazon CodeCatalyst com seu e-mail e senha. Essa parte do login é o primeiro fator, em que você usa algo que conhece. Em seguida, você faz login com um código ou chave de segurança. Esse é o segundo fator, que é algo que você tem. O segundo fator pode ser um código de autenticação gerado pelo seu dispositivo móvel ou ao tocar ou pressionar uma chave de segurança conectada ao seu computador. Juntos, esses vários fatores fornecem maior segurança ao impedir o acesso não autorizado.

## Como registrar um dispositivo para uso com autenticação multifator

Use o procedimento a seguir em Meu perfil > Autenticação multifator (MFA) para registrar seu novo dispositivo para autenticação multifator (MFA).

### Note

Recomendamos que você primeiro baixe a aplicação autenticadora apropriada no dispositivo antes de iniciar as etapas deste procedimento. Para obter uma lista de aplicativos que você pode usar para dispositivos com MFA, consulte [Aplicativos autenticadores](#).

Para registrar o dispositivo para usar com MFA

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No canto superior direito, escolha a seta ao lado do ícone com sua primeira inicial e, depois, selecione Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, selecione Gerenciar perfil e segurança. A página de perfil ID do builder AWS é aberta.
4. No lado esquerdo da página, selecione Segurança.
5. Na página Autenticação multifator, selecione Registrar dispositivo.
6. Na página Registrar dispositivo com MFA, selecione um dos seguintes tipos de dispositivos de MFA e siga as instruções:
  - Chave de segurança ou autenticador integrado
    1. Na página Register your user's security key, siga as instruções fornecidas pelo seu navegador ou plataforma.

 Note

Essa experiência varia de acordo com seu sistema operacional e navegador, portanto, siga as instruções exibidas pelo seu navegador ou plataforma. Depois que o dispositivo for registrado com sucesso, você terá a opção de associar um nome de exibição amigável ao dispositivo recém-inscrito. Se você quiser alterar isso, escolha Rename, insira o novo nome e escolha Save.

- Aplicação autenticadora

1. Na página Configurar o aplicativo autenticador, você pode observar informações de configuração para o novo dispositivo de MFA, inclusive um código QR gráfico. O gráfico é uma representação da chave secreta que está disponível para entrada manual em dispositivos que não suportam códigos de QR.
2. Usando o dispositivo MFA físico, faça o seguinte:
  - a. Abra uma aplicação autenticadora com MFA compatível. Para obter uma lista de aplicativos testados que você pode usar com dispositivos com MFA, consulte [Aplicativo autenticador testado](#). Se a aplicação de MFA comportar vários dispositivos, selecione a opção para criar um dispositivo de MFA.
  - b. Determine se o aplicativo MFA suporta códigos QR e, depois, execute uma das seguintes ações na página Set up the authenticator app:
    - i. No assistente, escolha Show QR code e, depois, use o app para digitalizar o código de QR. Por exemplo, você pode escolher o ícone de câmera ou escolher uma opção semelhante a Digitalizar código. Em seguida, use a câmera do dispositivo para digitalizar o código.
    - ii. Escolha mostrar chave secreta e, depois, insira essa chave secreta em seu aplicativo de MFA.

 Important

Ao configurar um dispositivo de MFA para o ID do builder AWS , recomendamos que você salve uma cópia do código QR ou da chave secreta em um local seguro. Isso pode ajudar se você perder o telefone ou precisar reinstalar a aplicação autenticadora com MFA. Se alguma dessas coisas

acontecer, você poderá reconfigurar rapidamente a aplicação para usar a mesma configuração de MFA.

3. Na página Configurar o aplicativo autenticador, em Código do autenticador, insira a senha de uso único que atualmente é exibida no dispositivo MFA físico.

 Important

Envie sua solicitação imediatamente após gerar o código. Se você gerar o código e esperar muito tempo para enviar a solicitação, o dispositivo de MFA será associado com sucesso ao seu perfil de ID do AWS construtor, mas o dispositivo de MFA ficará fora de sincronia. Isso ocorre porque as senhas únicas baseadas em tempo (time-based one-time passwords, TOTP) expiram após um curto período. Caso isso ocorra, você pode resincronizar o dispositivo.

4. Escolha Assign MFA. O dispositivo MFA agora pode começar a gerar senhas únicas e agora está pronto para uso.

## Aplicativos autenticadores

As aplicações autenticadoras são autenticadores de terceiros baseados em senha de uso único (OTP). Os usuários podem usar uma aplicação autenticadora instalada no dispositivo móvel ou tablet como um dispositivo de MFA autorizado. A aplicação autenticadora de terceiros deve estar em conformidade com RFC 6238, que é um algoritmo de senha de uso único com marcação temporal (TOTP) padrão capaz de gerar códigos de autenticação de seis dígitos.

Quando a MFA for solicitada, você deve inserir um código válido da aplicação autenticadora na caixa de entrada apresentada. Cada dispositivo MFA atribuído a um usuário deve ser exclusivo. Duas aplicações autenticadoras podem ser registradas para qualquer usuário.

### Aplicativo autenticador testado

Embora qualquer aplicação compatível com TOTP funcione com MFA do Centro de Identidade do IAM, você pode escolher entre as aplicações autenticadoras de terceiros conhecidas a seguir.

| Sistema operacional | Aplicativo autenticador testado                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Android             | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">Autenticador</a> , <a href="#">Microsoft LastPass Authenticator</a> , <a href="#">Google Authenticator</a> |
| iOS                 | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">Autenticador</a> , <a href="#">Microsoft LastPass Authenticator</a> , <a href="#">Google Authenticator</a> |

## Alteração dos dispositivos de MFA

Depois de registrar um dispositivo de MFA, você pode alterar o nome ou excluí-lo. Recomendamos sempre ter pelo menos um dispositivo de MFA ativado para uma camada extra de segurança. Você pode ter até cinco dispositivos registrados. Para saber como adicionar mais, consulte [Como registrar um dispositivo para uso com autenticação multifator](#).

### Renomear um dispositivo de MFA

Como renomear um dispositivo de MFA

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No canto superior direito, escolha a seta ao lado do ícone com sua primeira inicial e, depois, selecione Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, selecione Gerenciar perfil e segurança. A página de perfil ID do builder AWS é aberta.
4. Selecione Autenticação multifator no lado esquerdo da página. Você vê que a opção Renomear está esmaecida quando acessa a página.
5. Selecione o dispositivo de MFA que você deseja alterar. Escolha Rename (Renomear). Em seguida, um modal aparece.
6. No prompt aberto, insira o novo nome em Nome do dispositivo de MFA e selecione Renomear. O dispositivo renomeado é exibido em Dispositivos com autenticação multifator (MFA).

### Exclusão de um dispositivo de MFA

Como excluir um dispositivo de MFA

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.

2. No canto superior direito, escolha a seta ao lado do ícone com sua primeira inicial e, depois, selecione Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, selecione Gerenciar perfil e segurança. A página de perfil ID do builder AWS é aberta.
4. Selecione Autenticação multifator no lado esquerdo da página. Você vê que a opção Excluir está esmaecida quando acessa a página.
5. Selecione o dispositivo de MFA que você deseja alterar. Escolha Excluir. Aparece um modal que diz Excluir dispositivo de MFA?. Siga as instruções para excluir seu dispositivo.
6. Escolha Excluir. O dispositivo excluído não é mais exibido em Dispositivos com autenticação multifator (MFA).

## Segurança na Amazon CodeCatalyst

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 dos espaços mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam CodeCatalyst, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- Segurança na nuvem — Sua responsabilidade é determinada pelos AWS serviços que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e os regulamentos aplicáveis

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a Amazon CodeCatalyst. Ele mostra como configurar para atender CodeCatalyst aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus CodeCatalyst recursos.

## Conteúdo

- [Privacidade de dados no Amazon CodeCatalyst](#)
- [Proteção de dados na Amazon CodeCatalyst](#)
- [Identity and Access Management e Amazon CodeCatalyst](#)
- [Validação de conformidade do Amazon CodeCatalyst](#)
- [Resiliência no Amazon CodeCatalyst](#)
- [Segurança da infraestrutura no Amazon CodeCatalyst](#)
- [Análise de configuração e vulnerabilidade no Amazon CodeCatalyst](#)
- [Seus dados e privacidade no Amazon CodeCatalyst](#)
- [Melhores práticas para ações de fluxo de trabalho no Amazon CodeCatalyst](#)
- [Entendendo o modelo de CodeCatalyst confiança](#)

## Privacidade de dados no Amazon CodeCatalyst

O CodeCatalyst coleta informações agregadas no IAM Identity Center. O CodeCatalyst usa esses dados agregados para fornecer acesso a espaços e projetos no CodeCatalyst. As seguintes informações agregadas são coletadas:

- Nome completo do usuário
- E-mail do usuário
- Nome de usuário

Os dados do CodeCatalyst são criptografados automaticamente quando estão em repouso. Não será necessária nenhuma ação do cliente. Quando um usuário, centro de identidade ou espaço é excluído, o CodeCatalyst exclui os dados do cliente em até 24 horas.

O CodeCatalyst usa chaves do AWS KMS de propriedade da AWS. O CodeCatalyst não oferece suporte à criptografia em repouso usando chaves KMS gerenciadas pelo cliente para os atributos de identidade recuperados do IAM Identity Center.

Para obter mais informações sobre chaves do AWS KMS, consulte a documentação do usuário em chaves do [AWS KMS](#).

## Proteção de dados na Amazon CodeCatalyst

Segurança e conformidade são uma responsabilidade compartilhada entre a Amazon CodeCatalyst e o cliente, assim como o modelo de [responsabilidade AWS compartilhada modelo](#) de de se aplica ao uso dos AWS recursos usados em um fluxo de trabalho. Conforme descrito neste modelo, CodeCatalyst é responsável por proteger a infraestrutura global do serviço. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Esse modelo de responsabilidade compartilhada se aplica à proteção de dados em CodeCatalyst.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da sua conta e configure a autenticação multifatorial ao fazer login. Para obter mais informações, consulte [Configure seu AWS Builder ID para fazer login com autenticação multifator \(MFA\)](#).

Não insira informações confidenciais ou sigilosas, como endereços de e-mail dos clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui nomes de recursos e quaisquer outros identificadores que você inserir, CodeCatalyst além de qualquer um conectado Contas da AWS. Por exemplo, não insira informações confidenciais ou sigilosas como parte dos nomes do espaço, do projeto ou da frota de implantação. Quaisquer dados inseridos em tags, nomes ou campos de formato livre usados para nomes podem ser usados para logs de faturamento ou diagnóstico ou podem ser incluídos em caminhos de URL. Isso se aplica ao uso do console, da API AWS CLI, do CodeCatalyst Action Development Kit ou de qualquer outro AWS SDKs.

Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais de segurança no URL para validar a solicitação a esse servidor.

CodeCatalyst os repositórios de origem são criptografados automaticamente em repouso. Nenhuma ação do cliente é necessária. CodeCatalyst também criptografa os dados do repositório em trânsito usando o protocolo HTTPS.

CodeCatalyst não oferece suporte à criptografia em repouso usando chaves KMS gerenciadas pelo cliente para os atributos de identidade recuperados do IAM Identity Center.

CodeCatalyst suporta MFA. Para obter mais informações, consulte [Configure seu AWS Builder ID para fazer login com autenticação multifator \(MFA\)](#).

### Criptografia de dados

CodeCatalyst armazena e transfere dados com segurança dentro do serviço. Os dados são criptografados em trânsito e em repouso. Todos os dados criados ou armazenados pelo

serviço, incluindo quaisquer metadados do serviço, são armazenados nativamente no serviço e criptografados.

#### Note

Embora as informações sobre problemas sejam armazenadas com segurança no serviço, as informações sobre problemas em aberto também são armazenadas no cache local do navegador, onde você visualizou quadros de problemas, atrasos e problemas individuais. Para ter a segurança ideal, limpe o cache do navegador para remover essas informações.

Se você usar recursos vinculados a CodeCatalyst, como uma conexão de conta a um Conta da AWS ou a um repositório vinculado em GitHub, os dados em trânsito desse CodeCatalyst recurso vinculado serão criptografados, mas o tratamento de dados nesse recurso vinculado será gerenciado por esse serviço vinculado. Para ter mais informações, consulte a documentação do serviço vinculado e [Melhores práticas para ações de fluxo de trabalho no Amazon CodeCatalyst](#).

## Gerenciamento de chaves

CodeCatalyst usa chaves KMS AWS de propriedade própria. Isso significa que as chaves são gerenciadas por AWS e o cliente não precisa criá-las ou gerenciá-las diretamente. Isso simplifica o gerenciamento de chaves para cenários comuns de criptografia na AWS.

## Privacidade do tráfego entre redes

Ao criar um espaço CodeCatalyst, você escolhe Região da AWS onde os dados e os recursos serão armazenados para esse espaço. Os dados e metadados do projeto nunca saem da Região da AWS. [No entanto, para oferecer suporte à navegação interna CodeCatalyst, um conjunto limitado de metadados de espaço, projeto e usuário é replicado Regiões da AWS em toda a partição.](#) Ele não será replicado para Regiões da AWS fora dessa partição. Por exemplo, se você escolher Oeste dos EUA (Oregon) como Região da AWS ao criar seu espaço, seus dados não serão replicados para regiões da China ou AWS GovCloud (US). Para obter mais informações, consulte [Gerenciamento Regiões da AWS](#), [infraestrutura AWS global](#) e [endpoints AWS de serviço](#).

Os dados replicados Regiões da AWS dentro de uma partição incluem:

- Um valor de hash criptografado que representa o nome do espaço para garantir a exclusividade dos nomes dos espaços. Esse valor não é legível por humanos e não expõe os nomes reais dos espaços



- O ID exclusivo do espaço
- Metadados do espaço que auxiliam na navegação entre espaços
- O Região da AWS local onde o espaço está localizado
- O único IDs de todos os projetos no espaço
- O ID do perfil que indica o perfil de um usuário em um espaço ou projeto
- Ao se inscrever CodeCatalyst, dados e metadados sobre o processo de inscrição, incluindo:
  - O ID exclusivo do ID do builder AWS
  - O nome de exibição do usuário em seu ID do builder AWS
  - O alias do usuário em seu ID do builder AWS
  - O endereço de e-mail usado quando o usuário se inscreveu no ID do builder AWS
  - O progresso do processo de inscrição
  - Se estiver criando um espaço como parte do processo de inscrição, o Conta da AWS ID usado como conta de cobrança do espaço

Os nomes dos espaços são exclusivos em todos os lados CodeCatalyst. Não inclua dados confidenciais no nome do espaço.

Ao trabalhar com recursos vinculados e contas conectadas, como uma conexão a um repositório Conta da AWS ou a um GitHub repositório, recomendamos configurar seus locais de origem e destino com o nível mais alto de segurança suportado por cada um. CodeCatalyst protege a conexão entre Contas da AWS Regiões da AWS, e as zonas de disponibilidade usando o Transport Layer Security (TLS) 1.2.

## Identity and Access Management e Amazon CodeCatalyst

Na Amazon CodeCatalyst, você cria e usa um AWS Builder ID para fazer login e acessar seus espaços e projetos. Um ID de AWS construtor não é uma identidade no AWS Identity and Access Management (IAM) e não existe em um Conta da AWS. No CodeCatalyst entanto, integra-se ao IAM ao verificar um espaço para fins de cobrança e quando conectado a um Conta da AWS para criar e usar recursos nele. Conta da AWS

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Ao criar um espaço na Amazon CodeCatalyst, você deve se conectar Conta da AWS como a conta de cobrança do seu espaço. Você deve ter permissões de administrador no Conta da AWS para verificar o CodeCatalyst espaço ou ter a permissão. Você também tem a opção de adicionar uma função do IAM ao seu espaço que CodeCatalyst pode ser usada para criar e acessar recursos nesse espaço conectado Conta da AWS. Chamamos isso de um [perfil de serviço](#). Você pode optar por criar conexões com mais de uma Conta da AWS e criar funções de serviço para CodeCatalyst cada uma dessas contas.

#### Note

O faturamento CodeCatalyst ocorre na conta Conta da AWS designada como a cobrança. No entanto, se você criar uma função de CodeCatalyst serviço nessa Conta da AWS ou em qualquer outra conectada Conta da AWS, os recursos criados e usados pela função de CodeCatalyst serviço serão cobrados nessa função conectada Conta da AWS. Para obter mais informações, consulte [Gerenciando o faturamento](#) no Amazon CodeCatalyst Administrator Guide.

## Tópicos

- [Políticas baseadas em identidade no IAM](#)
- [Ações das políticas no IAM](#)
- [Recursos de políticas no IAM](#)
- [Chaves de condição de políticas no IAM](#)
- [Exemplos de políticas baseadas em identidade para conexões CodeCatalyst](#)
- [Uso de tags para controlar o acesso a recursos de conexão de conta](#)
- [CodeCatalyst referência de permissões](#)
- [Uso de perfis vinculados a serviço para o CodeCatalyst](#)
- [AWS políticas gerenciadas para a Amazon CodeCatalyst](#)
- [Conceda acesso aos AWS recursos do projeto com funções do IAM](#)

## Políticas baseadas em identidade no IAM

As políticas baseadas em identidade são documentos JSON de políticas de permissões que você pode anexar a uma entidade. Essa identidade pode ser um usuário, um grupo de usuários ou um perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos

e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criando políticas do IAM](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Saiba mais sobre todos os elementos que podem ser usados em uma política JSON consultando [Referência de Elementos de Política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o CodeCatalyst

Para ver exemplos de políticas CodeCatalyst baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para conexões CodeCatalyst](#)

## Ações das políticas no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [
 "prefix:action1",
 "prefix:action2"
]
```

## Recursos de políticas no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Isso pode

ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

## Chaves de condição de políticas no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica OR. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Para obter mais informações, consulte [Elementos da política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição globais da AWS, consulte [Chaves de contexto de condição globais da AWS](#) no Guia do usuário do IAM.

## Exemplos de políticas baseadas em identidade para conexões CodeCatalyst

Em CodeCatalyst, Contas da AWS são necessários para gerenciar o faturamento de um espaço e acessar recursos nos fluxos de trabalho do projeto. Uma conexão de conta é usada para autorizar a adição de Contas da AWS a um espaço. Políticas baseadas em identidade são usadas nas Contas da AWS conectadas.

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do CodeCatalyst. Eles também não podem realizar tarefas usando a AWS API Console de gerenciamento da AWS, AWS Command Line Interface (AWS CLI) ou. Um administrador do IAM deve criar políticas do

IAM que concedam aos usuários e funções permissão para executar ações nos recursos de que precisem. Por isso, o administrador precisa anexar essas políticas para os usuários que precisem delas.

Os exemplos de políticas do IAM a seguir concedem permissões para ações relacionadas às conexões de contas. Use-os para limitar o acesso para conectar contas CodeCatalyst a.

Exemplo 1: permitir que um usuário aceite solicitações de conexão em uma única Região da AWS

A política de permissões a seguir só permite que os usuários visualizem e aceitem solicitações de conexões entre CodeCatalyst Contas da AWS e. Além disso, a política usa uma condição para permitir somente as ações na região us-west-2 e não de outra. Regiões da AWS Para visualizar e aprovar a solicitação, o usuário faz login Console de gerenciamento da AWS com a mesma conta especificada na solicitação.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codecatalyst:AcceptConnection",
 "codecatalyst:GetPendingConnection"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:RequestedRegion": "us-west-2"
 }
 }
 }
]
}
```

Exemplo 2: Permitir o gerenciamento de conexões no console para um único Região da AWS

A política de permissões a seguir permite que os usuários gerenciem conexões entre CodeCatalyst e Contas da AWS em uma única região. A política usa uma condição para permitir somente as ações

na região us-west-2 e não de outra. Regiões da AWS Depois de criar uma conexão, você pode criar o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* escolhendo a opção no Console de gerenciamento da AWS. No exemplo de política, a condição para a iam:PassRole ação inclui os princípios de serviço para CodeCatalyst. Somente perfis com esse acesso serão criados no Console de gerenciamento da AWS.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codecatalyst:*"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:RequestedRegion": "us-west-2"
 }
 }
 },
 {
 "Effect": "Allow",
 "Action": [
 "iam:CreateRole",
 "iam:CreatePolicy",
 "iam:AttachRolePolicy",
 "iam:ListRoles"
],
 "Resource": "*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "iam:PassedToService": [
```

```
}
[
 }
 }
 }
 }
 "codecatalyst.amazonaws.com",
 "codecatalyst-runner.amazonaws.com"
]
 }
 }
}
```

### Exemplo 3: negar o gerenciamento de conexões

A política de permissões a seguir nega aos usuários qualquer capacidade de gerenciar conexões entre CodeCatalyst e Contas da AWS

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codecatalyst:*"
],
 "Resource": "*"
 }
]
}
```

## Uso de tags para controlar o acesso a recursos de conexão de conta

As tags podem ser anexadas ao recurso ou passadas na solicitação para serviços compatíveis com tags. Os recursos em políticas podem ter tags, e algumas ações em políticas podem incluir tags. As chaves de condição de marcação incluem as chaves de condição `aws:RequestTag` e `aws:ResourceTag`. Ao criar uma política do IAM, é possível usar chaves de condição de tag para controlar o seguinte:

- Quais usuários podem executar ações em um recurso de conexão, com base nas tags que o recurso já tem.

- Quais tags podem ser transmitidas na solicitação de uma ação.
- Se chaves de tags específicas podem ser usadas em uma solicitação.

Os exemplos a seguir demonstram como especificar condições de tag nas políticas para usuários de conexões de CodeCatalyst contas. Para obter mais informações sobre essas chaves de condição, consulte [Chaves de condição de políticas no IAM](#).

#### Exemplo 1: permitir ações com base em tags na solicitação

A política a seguir concede aos usuários permissão para aprovar conexões de conta.

Para fazer isso, ela permitirá as ações `AcceptConnection` e `TagResource` se a solicitação especificar uma tag denominada `Project` com o valor `ProjectA`. (A `aws:RequestTag` chave de condição é usada para controlar quais tags podem ser transmitidas em uma solicitação do IAM.) A `aws:TagKeys` condição garante que a chave de tag faça diferenciação de letras maiúsculas e minúsculas.

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codecatalyst:AcceptConnection",
 "codecatalyst:TagResource"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Project": "ProjectA"
 },
 "ForAllValues:StringEquals": {
 "aws:TagKeys": ["Project"]
 }
 }
 }
]
}
```



## Exemplo 2: permitir ações com base em tags de recursos

A política a seguir concede aos usuários permissão para executar ações e receber informações sobre recursos de conexão de conta.

Para fazer isso, ela permitirá ações específicas se a conexão tiver uma tag denominada `Project` com o valor `ProjectA`. (A `aws:ResourceTag` chave de condição é usada para controlar quais tags podem ser transmitidas em uma solicitação do IAM.)

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codecatalyst:GetConnection",
 "codecatalyst>DeleteConnection",
 "codecatalyst:AssociateIamRoleToConnection",
 "codecatalyst:DisassociateIamRoleFromConnection",
 "codecatalyst:ListIamRolesForConnection",
 "codecatalyst:PutBillingAuthorization"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Project": "ProjectA"
 }
 }
 }
]
}
```

## CodeCatalyst referência de permissões

Esta seção fornece uma referência de permissões para ações usadas com o recurso de conexão da conta para Contas da AWS as quais estão conectadas CodeCatalyst. A seção a seguir descreve ações somente com permissões relacionadas à conexão de contas.

## Permissões necessárias para conexões de contas

As permissões a seguir são necessárias para trabalhar com conexões de contas.

| CodeCatalyst permissões para conexões de contas | Permissões obrigatórias                                                                                                                                      | Recursos                                                                           |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| AcceptConnection                                | Necessário aceitar uma solicitação para conectar essa conta a um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | Compatível apenas com um caractere curinga (*) no elemento de política Resource.   |
| AssociateIamRoleToConnection                    | Necessário para associar um perfil do IAM a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| DeleteConnection                                | Necessário para excluir uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                         | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| DisassociateIamRoleFromConnection               | Necessário para desassociar um perfil do IAM a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                  | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetBillingAuthorization                         | Necessário para descrever a autorização de faturamento para uma conexão de conta. Esta é apenas uma permissão                                                | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

| CodeCatalyst permissões para conexões de contas | Permissões obrigatórias                                                                                                                                                    | Recursos                                                                           |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
|                                                 | de política do IAM, e não uma ação de API.                                                                                                                                 |                                                                                    |
| GetConnection                                   | Necessário para ter uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                                           | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetPendingConnection                            | Necessário para receber uma solicitação pendente para conectar essa conta a um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | Compatível apenas com um caractere curinga (*) no elemento de política Resource.   |
| ListConnections                                 | Necessário para listar conexões de contas que não estejam pendentes. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                | Compatível apenas com um caractere curinga (*) no elemento de política Resource.   |
| ListIamRolesForConnection                       | Necessário para listar perfis do IAM associados a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                             | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

| CodeCatalyst permissões para conexões de contas | Permissões obrigatórias                                                                                                                                             | Recursos                                                                           |
|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| ListTagsForResource                             | Necessário para listar tags associadas a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                               | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| PutBillingAuthorization                         | Necessário para criar ou atualizar a autorização de faturamento para uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.   | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| RejectConnection                                | Obrigatório para rejeitar uma solicitação para conectar essa conta a um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | Compatível apenas com um caractere curinga (*) no elemento de política Resource.   |
| TagResource                                     | Necessário para criar ou editar tags associadas a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                      | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| UntagResource                                   | Necessário para remover tags associadas a uma conexão de conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                              | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

## Permissões necessárias para aplicações do Centro de Identidade do IAM

As permissões a seguir são necessárias para trabalhar com aplicações do Centro de Identidade do IAM.

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias                                                                                                                                                             | Recursos                                                                                                             |
|-----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| AssociateIdentityCenterApplicationToSpace                       | Obrigatório para associar um aplicativo do IAM Identity Center a um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| AssociateIdentityToIdentityCenterApplication                    | Necessário para associar uma identidade a um aplicativo do IAM Identity Center para um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API.  | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchAssociateIdentitiesToIdentityCenterApplication             | É necessário associar várias identidades a um aplicativo do IAM Identity Center para um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchDisassociateIdentitiesFromIdentityCenterApplication        | É necessário desassociar várias identidades de um aplicativo do IAM Identity Center para um CodeCatalyst espaço. Esta é apenas uma                                                  | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias                                                                                                                                                                                             | Recursos                                                                                                             |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|                                                                 | permissão de política do IAM, e não uma ação de API.                                                                                                                                                                |                                                                                                                      |
| CreateIdentityCenterApplication                                 | Necessário para criar uma aplicação do Centro de Identidade do IAM. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                                                          | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| CreateSpaceAdminRoleAssignment                                  | Necessário para criar uma atribuição de função de administrador para um determinado CodeCatalyst espaço e aplicativo do IAM Identity Center. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DeleteIdentityCenterApplication                                 | Necessário para excluir uma aplicação do Centro de Identidade do IAM. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                                                        | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DisassociateIdentityCenterApplicationFromSpace                  | Obrigatório para desassociar um aplicativo do IAM Identity Center de um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                                 | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias                                                                                                                                                                | Recursos                                                                                                             |
|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| DisassociateIdentityFromIdentityCenterApplication               | Necessário para desassociar uma identidade de um aplicativo do IAM Identity Center para um CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| GetIdentityCenterApplication                                    | Necessário para ter informações sobre uma aplicação do Centro de Identidade do IAM. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                             | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListIdentityCenterApplications                                  | Necessário para visualizar uma lista de todas as aplicações do Centro de Identidade do IAM na conta. Esta é apenas uma permissão de política do IAM, e não uma ação de API.            | Compatível apenas com um caractere curinga (*) no elemento de política Resource.                                     |
| ListIdentityCenterApplicationsForSpace                          | Obrigatório para visualizar uma lista de aplicativos do IAM Identity Center por CodeCatalyst espaço. Esta é apenas uma permissão de política do IAM, e não uma ação de API.            | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias                                                                                                                                                                    | Recursos                                                                                                             |
|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| ListSpacesForIdentityCenterApplication                          | Necessário para visualizar uma lista de CodeCatalyst espaços pelo aplicativo IAM Identity Center. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                   | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| SynchronizIdentityCenterApplication                             | Necessário para sincronizar uma aplicação do Centro de Identidade do IAM com o repositório de identidades de apoio. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| UpdateIdentityCenterApplication                                 | Necessário para atualizar uma aplicação do Centro de Identidade do IAM. Esta é apenas uma permissão de política do IAM, e não uma ação de API.                                             | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

## Uso de perfis vinculados a serviço para o CodeCatalyst

O Amazon CodeCatalyst usa [perfis vinculados ao serviço](#) do AWS Identity and Access Management (IAM). Um perfil vinculado ao serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao CodeCatalyst. Os perfis vinculados ao serviço são predefinidos pelo CodeCatalyst e incluem todas as permissões que o serviço requer para chamar outros produtos da AWS em seu nome.

Um perfil vinculado ao serviço facilita a configuração do CodeCatalyst, já que você não precisa adicionar as permissões necessárias manualmente. O CodeCatalyst define as permissões dos perfis vinculados ao serviço e, exceto se definido de outra forma, somente o CodeCatalyst pode assumir



seus perfis. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do CodeCatalyst, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados ao serviço, consulte [serviços da AWS que funcionam com o IAM](#) e procure os serviços que apresentam Sim na coluna Perfis vinculados aos serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

#### Permissões de perfil vinculado a serviço para o CodeCatalyst

O CodeCatalyst usa o perfil vinculado ao serviço chamado `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` – Permite que o Amazon CodeCatalyst tenha acesso somente de leitura aos perfis de instância da aplicação e aos usuários e grupos do diretório associado em seu nome.

O perfil vinculado ao serviço

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` confia nos seguintes serviços para aceitar o perfil:

- `codecatalyst.amazonaws.com`

A política de permissões do perfil chamada

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` permite que o CodeCatalyst conclua as seguintes ações nos recursos especificados:

- Ação: `View application instance profiles and associated directory users and groups` para CodeCatalyst spaces that support identity federation and SSO users and groups

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para obter mais informações, consulte Permissões de perfil vinculado a serviços no Guia do usuário do IAM.

## Criação de um perfil vinculado a serviço para o CodeCatalyst

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um espaço no Console de gerenciamento da AWS, na AWS CLI ou com a API da AWS, o CodeCatalyst cria o perfil vinculado a serviço para você.

### Important

Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Além disso, se você estava usando o serviço do CodeCatalyst antes de 17 de novembro de 2023, quando começou a oferecer suporte aos perfis vinculados a serviços, então o CodeCatalyst criou o perfil de `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` em sua conta. Para saber mais, consulte [Um novo perfil apareceu na minha Conta da AWS](#).

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um espaço, o CodeCatalyst cria o perfil vinculado ao serviço para você novamente.

Você também pode usar o console do IAM para criar um perfil vinculado ao serviço com o caso de uso Visualizar perfis de instâncias de aplicações e usuários e grupos de diretórios associados. Na AWS CLI ou na API do AWS, crie um perfil vinculado a serviço com o nome de serviço `codecatalyst.amazonaws.com`. Para obter mais informações, consulte [Criar uma função vinculada ao serviço](#) no Manual do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

## Edição de um perfil vinculado a serviço para o CodeCatalyst

O CodeCatalyst não permite que você edite o perfil vinculado ao serviço `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

## Exclusão de um perfil vinculado a serviço para o CodeCatalyst

Você não precisa excluir manualmente a função `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Quando você exclui

um espaço na API Console de gerenciamento da AWS, AWS CLI ou AWS, o CodeCatalyst limpa os recursos e exclui o perfil vinculado ao serviço para você.

Também é possível usar o console do IAM, a AWS CLI ou a API da AWS para excluir manualmente a função vinculada ao serviço. Para isso, primeiro você deve limpar manualmente os recursos de sua função vinculada ao serviço e depois excluí-la manualmente.

 **Note**

Se o serviço do CodeCatalyst estiver usando o perfil quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir recursos do CodeCatalyst usados por `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`

- [Exclua o espaço](#).

Para excluir manualmente a função vinculada ao serviço usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada ao serviço `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Para obter mais informações, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados a serviço do CodeCatalyst

O CodeCatalyst é compatível com perfis vinculados ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Regiões e endpoints da AWS](#).

O CodeCatalyst não oferece suporte ao uso de funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Você pode usar a função `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` nas seguintes regiões.

| Nome da região                    | Identidade da região | Suporte no CodeCatalyst |
|-----------------------------------|----------------------|-------------------------|
| Leste dos EUA (Norte da Virgínia) | us-east-1            | Não                     |
| Leste dos EUA (Ohio)              | us-east-2            | Não                     |

| Nome da região                         | Identidade da região | Suporte no CodeCatalyst |
|----------------------------------------|----------------------|-------------------------|
| Oeste dos EUA (N. da Califórnia)       | us-west-1            | Não                     |
| Oeste dos EUA (Oregon)                 | us-west-2            | Sim                     |
| África (Cidade do Cabo)                | af-south-1           | Não                     |
| Ásia-Pacífico (Hong Kong)              | ap-east-1            | Não                     |
| Ásia-Pacífico (Jacarta)                | ap-southeast-3       | Não                     |
| Ásia-Pacífico (Mumbai)                 | ap-south-1           | Não                     |
| Ásia-Pacífico (Osaka)                  | ap-northeast-3       | Não                     |
| Ásia-Pacífico (Seul)                   | ap-northeast-2       | Não                     |
| Ásia-Pacífico (Singapura)              | ap-southeast-1       | Não                     |
| Ásia-Pacífico (Sydney)                 | ap-southeast-2       | Não                     |
| Ásia-Pacífico (Tóquio)                 | ap-northeast-1       | Não                     |
| Canadá (Central)                       | ca-central-1         | Não                     |
| Europa (Frankfurt)                     | eu-central-1         | Não                     |
| Europa (Irlanda)                       | eu-west-1            | Sim                     |
| Europa (Londres)                       | eu-west-2            | Não                     |
| Europa (Milão)                         | eu-south-1           | Não                     |
| Europa (Paris)                         | eu-west-3            | Não                     |
| Europa (Estocolmo)                     | eu-north-1           | Não                     |
| Oriente Médio (Barém)                  | me-south-1           | Não                     |
| Oriente Médio (Emirados Árabes Unidos) | me-central-1         | Não                     |

| Nome da região               | Identidade da região | Suporte no CodeCatalyst |
|------------------------------|----------------------|-------------------------|
| América do Sul (São Paulo)   | sa-east-1            | Não                     |
| AWS GovCloud (Leste dos EUA) | us-gov-east-1        | Não                     |
| AWS GovCloud (Oeste dos EUA) | us-gov-west-1        | Não                     |

## AWS políticas gerenciadas para a Amazon CodeCatalyst

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as [políticas gerenciadas pelo cliente](#) que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Guia do usuário do IAM.

### AWS política gerenciada: AmazonCodeCatalystSupportAccess

Essa é uma política que concede permissões para que todos os administradores e membros do espaço utilizem o plano de suporte premium Business ou Enterprise associado à conta de faturamento do espaço. Essas permissões permitem que administradores e membros do espaço utilizem o plano de suporte premium para os recursos para os quais têm permissões dentro das políticas de CodeCatalyst permissões.

## Detalhes das permissões

Esta política inclui as seguintes permissões.

- **support**— Concede permissões para permitir que os usuários pesquisem, criem e resolvam casos de AWS Support. Também concede permissões para descrever comunicações, níveis de gravidade, anexos e detalhes de casos de suporte relacionados.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "support:DescribeAttachment",
 "support:DescribeCaseAttributes",
 "support:DescribeCases",
 "support:DescribeCommunications",
 "support:DescribeIssueTypes",
 "support:DescribeServices",
 "support:DescribeSeverityLevels",
 "support:DescribeSupportLevel",
 "support:SearchForCases",
 "support:AddAttachmentsToSet",
 "support:AddCommunicationToCase",
 "support:CreateCase",
 "support:InitiateCallForCase",
 "support:InitiateChatForCase",
 "support:PutCaseAttributes",
 "support:RateCaseCommunication",
 "support:ResolveCase"
],
 "Resource": "*"
 }
]
}
```

```
}
```

## AWS política gerenciada: AmazonCodeCatalystFullAccess

Esta é uma política que concede permissões para gerenciar seu CodeCatalyst espaço e contas conectadas na página Amazon CodeCatalyst Spaces no Console de gerenciamento da AWS. Este aplicativo é usado para configurar Contas da AWS que estão conectados ao seu espaço em CodeCatalyst.

### Detalhes das permissões

Esta política inclui as seguintes permissões.

- `codecatalyst`— Concede permissões completas para a página Amazon CodeCatalyst Spaces no Console de gerenciamento da AWS.

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeCatalystResourceAccess",
 "Effect": "Allow",
 "Action": [
 "codecatalyst:*",
 "iam:ListRoles"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeCatalystAssociateIAMRole",
 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
```

```
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "iam:PassedToService": [
 "codecatalyst.amazonaws.com",
 "codecatalyst-runner.amazonaws.com"
]
 }
 }
]
}
```

### AWS política gerenciada: AmazonCodeCatalystReadOnlyAccess

Essa é uma política que concede permissões para visualizar e listar informações de espaços e contas conectadas na página Amazon CodeCatalyst Spaces no Console de gerenciamento da AWS. Este aplicativo é usado para configurar Contas da AWS que estão conectados ao seu espaço em CodeCatalyst.

#### Detalhes das permissões

Esta política inclui as seguintes permissões.

- `codecatalyst`— Concede permissões somente de leitura para a página do Amazon CodeCatalyst Spaces no. Console de gerenciamento da AWS

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codecatalyst:Get*",

```



```
 "codecatalyst:List*"
],
 "Resource": "*"
 }
]
}
```

AWS política gerenciada:

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy

Não é possível anexar

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy às entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite CodeCatalyst realizar ações em seu nome. Para obter mais informações, consulte [Uso de perfis vinculados a serviço para o CodeCatalyst](#).

Essa política permite que os clientes visualizem perfis de instâncias de aplicativos e usuários e grupos de diretórios associados ao gerenciar espaços em CodeCatalyst. Os clientes visualizarão esses recursos ao gerenciar espaços que oferecem suporte à federação de identidades e aos usuários e grupos de SSO.

### Detalhes das permissões

Esta política inclui as seguintes permissões.

- sso— Concede permissões para permitir que os usuários visualizem perfis de instâncias de aplicativos que são gerenciados no IAM Identity Center para espaços associados em CodeCatalyst.

### JSON

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

```

 "Sid":
 "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
 "Effect": "Allow",
 "Action": [
 "sso:ListInstances",
 "sso:ListApplications",
 "sso:ListApplicationAssignments",
 "sso:DescribeInstance",
 "sso:DescribeApplication"
],
 "Resource": "*"
}
]
}

```

## CodeCatalyst atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas CodeCatalyst desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página [Histórico do CodeCatalyst documento](#).

| Alteração                                                                                                       | Descrição                                                                                                                                                                                                   | Data                   |
|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">AmazonCodeCatalyst ServiceRoleForIdentityCenterApplicationSynchronizationPolicy</a> – Nova política | <p>CodeCatalyst adicionou a política.</p> <p>Concede permissões para permitir que CodeCatalyst os usuários visualizem perfis de instâncias de aplicativos e usuários e grupos de diretórios associados.</p> | 17 de novembro de 2023 |
| <a href="#">AmazonCodeCatalyst SupportAccess</a> – Nova política                                                | <p>CodeCatalyst adicionou a política.</p> <p>Concede permissões para permitir que CodeCatalyst os usuários pesquisem,</p>                                                                                   | 20 de abril de 2023    |

| Alteração                                                         | Descrição                                                                                     | Data                |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------|
|                                                                   | criem e resolvam casos de suporte, bem como visualize m comunicações e detalhes relacionados. |                     |
| <a href="#">AmazonCodeCatalyst FullAccess</a> – Nova política     | CodeCatalyst adicionou a política.<br><br>Concede acesso total CodeCatalyst a.                | 20 de abril de 2023 |
| <a href="#">AmazonCodeCatalyst ReadOnlyAccess</a> – Nova política | CodeCatalyst adicionou a política.<br><br>Concede acesso somente para leitura a. CodeCatalyst | 20 de abril de 2023 |
| CodeCatalyst começou a rastrear as alterações                     | CodeCatalyst começou a rastrear as mudanças em suas políticas AWS gerenciadas.                | 20 de abril de 2023 |

## Conceda acesso aos AWS recursos do projeto com funções do IAM

CodeCatalyst pode acessar AWS recursos conectando seu Conta da AWS a um CodeCatalyst espaço. Em seguida, você pode criar os seguintes perfis de serviço e associá-los ao conectar sua conta.

Para ter mais informações sobre os elementos usados em uma política JSON, consulte [Referência de elementos de política JSON do IAM](#) no Guia do usuário do IAM.

- Para acessar recursos em e Conta da AWS para seus CodeCatalyst projetos e fluxos de trabalho, você deve primeiro conceder permissão CodeCatalyst para acessar esses recursos em seu nome. Para fazer isso, você deve criar uma função de serviço em um ambiente conectado Conta da AWS que CodeCatalyst possa assumir em nome de usuários e projetos no espaço. Você pode optar por criar e usar o perfil de serviço CodeCatalystWorkflowDevelopmentRole-*spaceName* ou criar perfis

de serviço personalizados e configurar essas políticas e perfis do IAM manualmente. Como prática recomendada, atribua esses perfis à menor quantidade de permissões necessária.

**Note**

Para funções de serviço personalizadas, é necessário o responsável pelo CodeCatalyst serviço. Para obter mais informações sobre o principal CodeCatalyst de serviço e o modelo de confiança, consulte [Entendendo o modelo de CodeCatalyst confiança](#).

- Para gerenciar o suporte para um espaço por meio do Connected Conta da AWS, você pode escolher criar e usar a função de `AWSRoleForCodeCatalystSupport` serviço que permite que CodeCatalyst os usuários acessem o suporte. Para obter mais informações sobre suporte para um CodeCatalyst espaço, consulte [Suporte para o Amazon CodeCatalyst](#).

Noções básicas sobre o perfil de serviço `CodeCatalystWorkflowDevelopmentRole-spaceName`

Você pode adicionar uma função do IAM ao seu espaço que CodeCatalyst pode ser usada para criar e acessar recursos em um ambiente conectado Conta da AWS. Chamamos isso de um [perfil de serviço](#). A maneira mais simples de criar um perfil de serviço é adicionar um ao criar o espaço e escolher a opção `CodeCatalystWorkflowDevelopmentRole-spaceName` para esse perfil. Isso não apenas cria a função de serviço com o `AdministratorAccess` anexo, mas também cria a política de confiança que permite assumir CodeCatalyst a função em nome dos usuários em projetos no espaço. O perfil de serviço tem como escopo o espaço, não os projetos individuais. Para criar essa função, consulte [Criar o perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` para a conta e o espaço](#). Só será possível criar um perfil para cada espaço em cada conta.

**Note**

Essa função só é recomendada para uso com contas de desenvolvimento e usa a política `AdministratorAccess` AWS gerenciada, dando a ela acesso total para criar novas políticas e recursos nela Conta da AWS.

A política anexada ao perfil `CodeCatalystWorkflowDevelopmentRole-spaceName` foi projetada para funcionar com projetos criados com esquemas no espaço. Ela permite que os usuários desses projetos desenvolvam, criem, testem e implantem código usando recursos na Conta da AWS conectada. Para obter mais informações, consulte [Criação de uma função para um AWS serviço](#).

A política anexada à CodeCatalystWorkflowDevelopmentRole-*spaceName* função é a política AdministratorAccess gerenciada em AWS. Essa é uma política que concede acesso total a todas as AWS ações e recursos. Para ver o documento de política JSON no console do IAM, consulte [AdministratorAccess](#).

A política de confiança a seguir CodeCatalyst permite assumir a CodeCatalystWorkflowDevelopmentRole-*spaceName* função. Para obter mais informações sobre o modelo de CodeCatalyst confiança, consulte [Entendendo o modelo de CodeCatalyst confiança](#).

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

Criar o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* para a conta e o espaço

Siga estas etapas para criar o perfil CodeCatalystWorkflowDevelopmentRole-*spaceName* que será usado para fluxos de trabalho em seu espaço. Para cada conta que você deseja que tenha perfis do IAM para uso em projetos, no seu espaço, você deve adicionar um perfil, como o perfil de desenvolvedor.

Antes de começar, você deve ter privilégios administrativos para seu administrador Conta da AWS ou ser capaz de trabalhar com ele. Para obter mais informações sobre como Contas da AWS as funções do IAM são usadas em CodeCatalyst, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Para criar e adicionar o CodeCatalyst `CodeCatalystWorkflowDevelopmentRole-spaceName`

1. Antes de começar no CodeCatalyst console, abra o e Console de gerenciamento da AWS, em seguida, verifique se você está logado com o mesmo Conta da AWS em seu espaço.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
4. Escolha o link para Conta da AWS onde você deseja criar a função. A página Detalhes da Conta da AWS é exibida.
5. Escolha Gerenciar funções em Console de gerenciamento da AWS.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

6. Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento. O perfil terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço `CodeCatalystWorkflowDevelopmentRole-spaceName`](#).

 Note

Essa função só é recomendada para uso com contas de desenvolvedor e usa a política `AdministratorAccess` AWS gerenciada, dando a ela acesso total para criar novas políticas e recursos nela Conta da AWS.

7. Selecione Criar perfil de desenvolvimento.
8. Na página de conexões, em Funções do IAM disponíveis para CodeCatalyst, veja a `CodeCatalystWorkflowDevelopmentRole-spaceName` função na lista de funções do IAM adicionadas à sua conta.
9. Para retornar ao seu espaço, escolha Go to Amazon CodeCatalyst.

Noções básicas sobre o perfil de serviço `AWSRoleForCodeCatalystSupport`

Você pode adicionar uma função do IAM ao seu espaço que CodeCatalyst os usuários em um espaço possam usar para criar e acessar casos de suporte. Isso é chamado de [perfil de serviço](#)

para suporte. A maneira mais simples de criar um perfil de serviço para suporte é adicionar um ao criar o espaço e escolher a opção `AWSRoleForCodeCatalystSupport` para esse perfil. Isso não apenas cria a política e a função, mas também cria a política de confiança que CodeCatalyst permite assumir a função em nome dos usuários em projetos no espaço. O perfil de serviço tem como escopo o espaço, não os projetos individuais. Para criar essa função, consulte [Criação do perfil `AWSRoleForCodeCatalystSupport` para a conta e o espaço](#).

A política anexada ao perfil `AWSRoleForCodeCatalystSupport` é uma política gerenciada que fornece acesso às permissões de suporte. Para obter mais informações, consulte [AWS política gerenciada: `AmazonCodeCatalystSupportAccess`](#).

A função de confiança da política CodeCatalyst permite assumir a função.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "codecatalyst.amazonaws.com",
 "codecatalyst-runner.amazonaws.com"
]
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

### Criação do perfil `AWSRoleForCodeCatalystSupport` para a conta e o espaço

Siga estas etapas para criar o perfil `AWSRoleForCodeCatalystSupport` que será usado para casos de suporte em seu espaço. O perfil deve ser adicionado à conta de faturamento designada para o espaço.

Antes de começar, você deve ter privilégios administrativos para seu administrador Conta da AWS ou ser capaz de trabalhar com ele. Para obter mais informações sobre como Contas da AWS as

funções do IAM são usadas em CodeCatalyst, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Para criar e adicionar o CodeCatalyst AWSRoleForCodeCatalystSupport

1. Antes de começar no CodeCatalyst console, abra o e Console de gerenciamento da AWS, em seguida, verifique se você está logado com o mesmo Conta da AWS em seu espaço.
2. Navegue até seu CodeCatalyst espaço. Escolha Settings (Configurações) e Contas da AWS.
3. Escolha o link para Conta da AWS onde você deseja criar a função. A página Detalhes da Conta da AWS é exibida.
4. Escolha Gerenciar funções em Console de gerenciamento da AWS.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no Console de gerenciamento da AWS. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessá-la.

5. Em detalhes do CodeCatalyst espaço, escolha Adicionar função de CodeCatalyst Support. Essa opção cria um perfil de serviço que contém a política de permissões e a política de confiança para o perfil de desenvolvimento de demonstração. O perfil terá um nome AWSRoleForCodeCatalystSupport com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço AWSRoleForCodeCatalystSupport](#).
6. Na página Adicionar função para CodeCatalyst Support, deixe o padrão selecionado e escolha Criar função.
7. Em Funções do IAM disponíveis para CodeCatalyst, veja a CodeCatalystWorkflowDevelopmentRole-*spaceName* função na lista de funções do IAM adicionadas à sua conta.
8. Para retornar ao seu espaço, escolha Go to Amazon CodeCatalyst.

Configurando funções do IAM para ações de fluxo de trabalho em CodeCatalyst

Esta seção detalha as funções e políticas do IAM que você pode criar para usar com sua CodeCatalyst conta. Para receber instruções para criar perfis de exemplo, consulte [Criação manual de perfis para ações de fluxo de trabalho](#). Depois de criar o perfil do IAM, copie o ARN da função para adicionar o perfil do IAM à conexão da sua conta e associá-lo ao ambiente do projeto. Para saber mais, consulte [Adicionar perfis do IAM às conexões da conta](#).



## CodeCatalyst função de criação para acesso ao Amazon S3

Para ações CodeCatalyst de criação de fluxo de trabalho, você pode usar a função de CodeCatalystWorkflowDevelopmentRole-**spaceName** serviço padrão ou criar uma função do IAM chamada CodeCatalystBuildRoleforS3Access. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas em CloudFormation recursos em seu Conta da AWS.

Esse perfil concede permissões para o seguinte:

- Gravar em buckets do Amazon S3.
- Support a construção de recursos com CloudFormation. Isso requer acesso do Amazon S3.

Esse perfil usa a seguinte política:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst criar função para CloudFormation

Para ações CodeCatalyst de criação de fluxo de trabalho, você pode usar a função de CodeCatalystWorkflowDevelopmentRole-**spaceName** serviço padrão ou criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas em CloudFormation recursos em seu Conta da AWS.

Esse perfil concede permissões para o seguinte:

- Support a construção de recursos com CloudFormation. Isso é necessário junto com a função de CodeCatalyst criação para acesso ao Amazon S3 e a função de CodeCatalyst implantação para. CloudFormation

As seguintes políticas AWS gerenciadas devem ser anexadas a essa função:

- `AWSCloudFormationFullAccess`
- `IAMFullAcesso`
- `Amazon S3 FullAccess`
- `APIGatewayAdministrador da Amazon`
- `AWSLambdaFullAccess`

### CodeCatalyst função de criação para o CDK

Para CodeCatalyst fluxos de trabalho que executam ações de criação do CDK, como o aplicativo web moderno de três camadas, você pode usar a função de `CodeCatalystWorkflowDevelopmentRole-spaceName` serviço padrão ou criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa inicializar e executar comandos de criação do CDK para CloudFormation recursos em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Gravar em buckets do Amazon S3.
- Support a construção de construções de CDK e pilhas de CloudFormation recursos. Isso requer acesso ao Amazon S3 para armazenamento de artefatos, ao Amazon ECR para suporte ao repositório de imagens e ao SSM para governança e monitoramento do sistema para instâncias virtuais.

Esse perfil usa a seguinte política:

#### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função de implantação para CloudFormation

Para ações CodeCatalyst de implantação de fluxo de trabalho que usam CloudFormation, você pode usar a função de CodeCatalystWorkflowDevelopmentRole-*spaceName* serviço padrão ou usar uma política com permissões de escopo que CodeCatalyst precisa executar tarefas em CloudFormation recursos em seu Conta da AWS.

Esse perfil concede permissões para o seguinte:

- Permita CodeCatalyst invocar uma função  $\lambda$  para realizar a blue/green CloudFormation implantação.
- Permite CodeCatalyst criar e atualizar pilhas e conjuntos de alterações em CloudFormation

Esse perfil usa a seguinte política:

```
{
 "Action": [
 "cloudformation:CreateStack",
 "cloudformation:DeleteStack",
 "cloudformation:Describe*",
 "cloudformation:UpdateStack",
 "cloudformation:CreateChangeSet",
 "cloudformation:DeleteChangeSet",
 "cloudformation:ExecuteChangeSet",
 "cloudformation:SetStackPolicy",
 "cloudformation:ValidateTemplate",
 "cloudformation:List*",
 "iam:PassRole"
],
 "Resource": "resource_ARN",
 "Effect": "Allow"
}
```

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

## CodeCatalyst função de implantação para a Amazon EC2

CodeCatalyst as ações de implantação do fluxo de trabalho usam uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões específicas que CodeCatalyst precisa executar tarefas nos EC2 recursos da Amazon em seu Conta da AWS. A política padrão para a CodeCatalystWorkflowDevelopmentRole-**spaceName** função não inclui permissões para Amazon EC2 ou Amazon EC2 Auto Scaling.

Esse perfil concede permissões para o seguinte:

- Crie EC2 implantações da Amazon.
- Leia as tags em uma instância ou identifique uma EC2 instância da Amazon pelos nomes dos grupos do Auto Scaling.
- Leia, crie, atualize e exclua grupos, ganchos de ciclo de vida e políticas de escalabilidade do Amazon EC2 Auto Scaling.
- Publique informações nos tópicos do Amazon SNS.
- Recupere informações sobre CloudWatch alarmes.
- Leia e atualize o Elastic Load Balancing.

Esse perfil usa a seguinte política:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função de implantação para o Amazon ECS

Para ações CodeCatalyst de fluxo de trabalho, você pode criar uma função do IAM com as permissões necessárias. Você pode usar a função de CodeCatalystWorkflowDevelopmentRole-**spaceName** serviço padrão ou criar uma função do IAM para CodeCatalyst implantar ações a serem usadas em implantações do Lambda. Essa função usa

uma política com permissões definidas que CodeCatalyst precisa executar tarefas nos recursos do Amazon ECS em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Inicie a implantação contínua do Amazon ECS em nome de um CodeCatalyst usuário, em uma conta especificada na CodeCatalyst conexão.
- Ler, atualizar e excluir conjuntos de tarefas do Amazon ECS.
- Atualizar grupos de destino, receptores e regras do Elastic Load Balancing.
- Invocar funções do Lambda.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Esse perfil usa a seguinte política:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Action": [
 "ecs:DescribeServices",
 "ecs:CreateTaskSet",
 "ecs>DeleteTaskSet",
 "ecs:ListClusters",
 "ecs:RegisterTaskDefinition",
 "ecs:UpdateServicePrimaryTaskSet",
 "ecs:UpdateService",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeListeners",
 "elasticloadbalancing:ModifyListener",
 "elasticloadbalancing:DescribeRules",
 "elasticloadbalancing:ModifyRule",
 "lambda:InvokeFunction",
 "lambda:ListFunctions",
 "cloudwatch:DescribeAlarms",
 "sns:Publish",
 "sns:ListTopics",
```

```

 "s3:GetObject",
 "s3:GetObjectVersion",
 "codedeploy:CreateApplication",
 "codedeploy:CreateDeployment",
 "codedeploy:CreateDeploymentGroup",
 "codedeploy:GetApplication",
 "codedeploy:GetDeployment",
 "codedeploy:GetDeploymentGroup",
 "codedeploy:ListApplications",
 "codedeploy:ListDeploymentGroups",
 "codedeploy:ListDeployments",
 "codedeploy:StopDeployment",
 "codedeploy:GetDeploymentTarget",
 "codedeploy:ListDeploymentTargets",
 "codedeploy:GetDeploymentConfig",
 "codedeploy:GetApplicationRevision",
 "codedeploy:RegisterApplicationRevision",
 "codedeploy:BatchGetApplicationRevisions",
 "codedeploy:BatchGetDeploymentGroups",
 "codedeploy:BatchGetDeployments",
 "codedeploy:BatchGetApplications",
 "codedeploy:ListApplicationRevisions",
 "codedeploy:ListDeploymentConfigs",
 "codedeploy:ContinueDeployment"
],
 "Resource": "*",
 "Effect": "Allow"
 }, { "Action": [
 "iam:PassRole"
],
 "Effect": "Allow",
 "Resource": "*",
 "Condition": { "StringLike": { "iam:PassedToService": [
 "ecs-tasks.amazonaws.com",
 "codedeploy.amazonaws.com"
]
 }
 }
}
}]
}

```

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função de implantação para Lambda

Para ações CodeCatalyst de fluxo de trabalho, você pode criar uma função do IAM com as permissões necessárias. Você pode usar a função de CodeCatalystWorkflowDevelopmentRole-*spaceName* serviço padrão ou criar uma função do IAM para CodeCatalyst implantar ações a serem usadas em implantações do Lambda. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas nos recursos do Lambda em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Ler, atualizar e invocar funções do Lambda e aliases.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre alarmes de CloudWatch eventos.
- Publique informações nos tópicos do Amazon SNS.

Esse perfil usa a seguinte política:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função de implantação para Lambda

Para ações CodeCatalyst de fluxo de trabalho, você pode usar a função de CodeCatalystWorkflowDevelopmentRole-**spaceName** serviço padrão ou criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas nos recursos do Lambda em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Ler, atualizar e invocar funções do Lambda e aliases.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Esse perfil usa a seguinte política:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função de implantação para AWS SAM

Para ações CodeCatalyst de fluxo de trabalho, você pode usar a função de CodeCatalystWorkflowDevelopmentRole-**spaceName** serviço padrão ou criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas AWS SAM e CloudFormation recursos em seu Conta da AWS.

Esse perfil concede permissões para o seguinte:

- Permita CodeCatalyst invocar uma função Lambda para realizar a implantação de aplicativos sem servidor AWS SAM e CLI.
- Permite CodeCatalyst criar e atualizar pilhas e conjuntos de alterações em. CloudFormation



Esse perfil usa a seguinte política:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

### CodeCatalyst função somente para leitura na Amazon EC2

Para ações CodeCatalyst de fluxo de trabalho, você pode criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões específicas que CodeCatalyst precisa executar tarefas nos EC2 recursos da Amazon em seu Conta da AWS. A função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço não inclui permissões para a Amazon EC2 nem as ações descritas para a Amazon CloudWatch.

Esse perfil concede permissões para o seguinte:

- Obtenha o status das EC2 instâncias da Amazon.
- Obtenha CloudWatch métricas para EC2 instâncias da Amazon.

Esse perfil usa a seguinte política:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função somente de leitura para Amazon ECS

Para ações CodeCatalyst de fluxo de trabalho, você pode criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas nos recursos do Amazon ECS em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Ler conjuntos de tarefas do Amazon ECS.
- Recupere informações sobre CloudWatch alarmes.

Esse perfil usa a seguinte política:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## CodeCatalyst função somente de leitura para Lambda

Para ações CodeCatalyst de fluxo de trabalho, você pode criar uma função do IAM com as permissões necessárias. Essa função usa uma política com permissões definidas que CodeCatalyst precisa executar tarefas nos recursos do Lambda em seu. Conta da AWS

Esse perfil concede permissões para o seguinte:

- Ler funções e aliases do Lambda.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.

Essa função usa a seguinte política do .

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

## Criação manual de perfis para ações de fluxo de trabalho

CodeCatalyst as ações de fluxo de trabalho usam funções do IAM que você cria, chamadas de função de construção, função de implantação e função de pilha.

Siga estas etapas para criar esses perfis no IAM.

Para criar um perfil de implantação

1. Crie uma política para o perfil da seguinte maneira:
  - a. Faça login em AWS.
  - b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
  - c. No painel de navegação, selecione Políticas.
  - d. Escolha Criar política.
  - e. Escolha a guia JSON.
  - f. Exclua o código existente.
  - g. Cole o seguinte código:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

- h. Escolha Próximo: tags.

- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

**codecatalyst-deploy-policy**

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de implantação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure **codecatalyst-deploy-policy** e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

**codecatalyst-deploy-role**

- i. Em Descrição do perfil, insira:

**CodeCatalyst deploy role**

- j. Selecione Criar perfil.

Agora você criou um perfil de implantação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN do perfil de implantação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (**codecatalyst-deploy-role**).

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de implantação com as permissões apropriadas e obteve o ARN.

## Como criar um perfil de criação

1. Crie uma política para o perfil da seguinte maneira:

- a. Faça login em AWS.
- b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, selecione Políticas.
- d. Escolha Criar política.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

### Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-build-policy
```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Em Políticas de permissões, procure `codecatalyst-build-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

**`codecatalyst-build-role`**

- i. Em Descrição do perfil, insira:

**`CodeCatalyst build role`**

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN do perfil de criação, da seguinte forma:

- a. No painel de navegação, selecione Perfis.
- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (`codecatalyst-build-role`).
- c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de criação com as permissões apropriadas e obteve o ARN.

## Para criar um perfil de pilha

### Note

Você não precisa criar um perfil de pilha, embora isso seja recomendado por motivos de segurança. Se você não criar o perfil de pilha, precisará adicionar as políticas de permissões descritas mais adiante neste procedimento ao perfil de implantação.

1. Faça login AWS usando a conta na qual você deseja implantar sua pilha.
2. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
3. No painel de navegação, escolha Perfis e Criar perfil.
4. Na parte superior, selecione Serviço da AWS .
5. Na lista de serviços, escolha CloudFormation.
6. Escolha Próximo: Permissões.
7. Na caixa de pesquisa, adicione todas as políticas necessárias para acessar os recursos em sua pilha. Por exemplo, se sua pilha inclui uma AWS Lambda função, você precisa adicionar uma política que conceda acesso ao Lambda.

### Tip

Se não tiver certeza de quais políticas adicionar, você pode omiti-las por enquanto. Quando você testa a ação, se você não tiver as permissões corretas, CloudFormation gera erros que mostram quais permissões você precisa adicionar.

8. Escolha Próximo: tags.
9. Selecione Próximo: revisar.
10. Em Nome do perfil, insira:

**codecatalyst-stack-role**

11. Selecione Criar perfil.
12. Para obter o ARN do perfil de pilha, faça o seguinte:
  - a. No painel de navegação, selecione Perfis.

- b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (codecatalyst-stack-role).
- c. Escolha o perfil na lista.
- d. Na página Resumo, copie o valor de ARN do perfil.

## Usando AWS CloudFormation para criar políticas e funções no IAM

Você pode escolher criar e usar AWS CloudFormation modelos para criar as políticas e funções necessárias para acessar recursos em seus CodeCatalyst projetos e fluxos de trabalho. Conta da AWS CloudFormation é um serviço que ajuda você a modelar e configurar seus AWS recursos para que você possa passar menos tempo gerenciando esses recursos e mais tempo se concentrando nos aplicativos que são executados em AWS. Se você pretende criar funções em várias Contas da AWS, criar um modelo pode ajudá-lo a realizar essa tarefa mais rapidamente.

O modelo de exemplo a seguir cria uma política e um perfil de ação de implantação.

```
Parameters:
 CodeCatalystAccountId:
 Type: String
 Description: Account ID from the connections page
 ExternalId:
 Type: String
 Description: External ID from the connections page
Resources:
 CrossAccountRole:
 Type: 'AWS::IAM::Role'
 Properties:
 AssumeRolePolicyDocument:
 Version: "2012-10-17"
 Statement:
 - Effect: Allow
 Principal:
 AWS:
 - !Ref CodeCatalystAccountId
 Action:
 - 'sts:AssumeRole'
 Condition:
 StringEquals:
 sts:ExternalId: !Ref ExternalId
 Path: /
 Policies:
```



```
- PolicyName: CodeCatalyst-CloudFormation-action-policy
PolicyDocument:
 Version: "2012-10-17"
 Statement:
 - Effect: Allow
 Action:
 - 'cloudformation:CreateStack'
 - 'cloudformation>DeleteStack'
 - 'cloudformation:Describe*'
 - 'cloudformation:UpdateStack'
 - 'cloudformation:CreateChangeSet'
 - 'cloudformation>DeleteChangeSet'
 - 'cloudformation:ExecuteChangeSet'
 - 'cloudformation:SetStackPolicy'
 - 'cloudformation:ValidateTemplate'
 - 'cloudformation:List*'
 - 'iam:PassRole'
 Resource: '*'
```

Criar o perfil manualmente para o esquema da aplicação web

O blueprint do aplicativo CodeCatalyst web usa funções do IAM que você cria, chamadas de função de construção para CDK, função de implantação e função de pilha.

Siga estas etapas para criar o perfil no IAM.

Como criar um perfil de criação

1. Crie uma política para o perfil da seguinte maneira:
  - a. Faça login em AWS.
  - b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
  - c. No painel de navegação, selecione Políticas.
  - d. Escolha Create Policy.
  - e. Escolha a guia JSON.
  - f. Exclua o código existente.
  - g. Cole o seguinte código:

 **Note**

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-webapp-build-policy
```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Anexe a política de permissões ao perfil de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise `codecatalyst-webapp-build-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

```
codecatalyst-webapp-build-role
```

- i. Em Descrição do perfil, insira:

```
CodeCatalyst Web app build role
```

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

3. Anexe a política de permissões ao perfil de criação, da seguinte maneira:
  - a. No painel de navegação, selecione Perfis e procure `codecatalyst-webapp-build-role`.
  - b. Selecione `codecatalyst-webapp-build-role` para exibir os detalhes.
  - c. Na guia Permissões, escolha Adicionar permissões e Anexar políticas.
  - d. Procure `codecatalyst-webapp-build-policy`, marque a caixa de seleção e selecione Anexar políticas.

Agora você anexou a política de permissões ao perfil de criação. O perfil de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN do perfil de criação, da seguinte forma:
  - a. No painel de navegação, selecione Perfis.
  - b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (`codecatalyst-webapp-build-role`).
  - c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de criação com as permissões apropriadas e obteve o ARN.

## Criação manual de perfis para o esquema do SAM

O blueprint do CodeCatalyst SAM usa funções do IAM que você cria, chamadas de função de criação CloudFormation e função de implantação do SAM.

Siga estas etapas para criar os perfis no IAM.

Para criar uma função de construção para CloudFormation

1. Crie uma política para o perfil da seguinte maneira:

- a. Faça login em AWS.
- b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, selecione Políticas.
- d. Escolha Create Policy.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:*",
 "cloudformation:*"
],
 "Resource": "*"
 }
]
}
```

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.

- j. Em Nome, insira:

**codecatalyst-SAM-build-policy**

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Anexe a política de permissões ao perfil de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise **codecatalyst-SAM-build-policy** e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

**codecatalyst-SAM-build-role**

- i. Em Descrição do perfil, insira:

**CodeCatalyst SAM build role**

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

3. Anexe a política de permissões ao perfil de criação, da seguinte maneira:

- a. No painel de navegação, selecione Perfis e procure **codecatalyst-SAM-build-role**.
- b. Selecione **codecatalyst-SAM-build-role** para exibir os detalhes.
- c. Na guia Permissões, escolha Adicionar permissões e Anexar políticas.

- d. Procure `codecatalyst-SAM-build-policy`, marque a caixa de seleção e selecione Anexar políticas.

Agora você anexou a política de permissões ao perfil de criação. O perfil de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN do perfil de criação, da seguinte forma:
  - a. No painel de navegação, selecione Perfis.
  - b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (`codecatalyst-SAM-build-role`).
  - c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de criação com as permissões apropriadas e obteve o ARN.

Para criar um perfil de implantação para o SAM

1. Crie uma política para o perfil da seguinte maneira:
  - a. Faça login em AWS.
  - b. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
  - c. No painel de navegação, selecione Políticas.
  - d. Escolha Create Policy.
  - e. Escolha a guia JSON.
  - f. Exclua o código existente.
  - g. Cole o seguinte código:

 Note

Na primeira vez em que o perfil for usado para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

- h. Escolha Próximo: tags.
- i. Selecione Próximo: revisar.
- j. Em Nome, insira:

```
codecatalyst-SAM-deploy-policy

```

- k. Selecione Criar política.

Agora você criou uma política de permissões.

2. Crie o perfil de criação, da seguinte forma:

- a. No painel de navegação, escolha Perfis e Criar perfil.
- b. Selecione Política de confiança personalizada.
- c. Exclua a política de confiança personalizada existente.
- d. Adicione a política de confiança personalizada a seguir:
- e. Escolha Próximo.
- f. Anexe a política de permissões ao perfil de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise `codecatalyst-SAM-deploy-policy` e marque a caixa de seleção.
- g. Escolha Próximo.
- h. Em Nome do perfil, insira:

```
codecatalyst-SAM-deploy-role

```

- i. Em Descrição do perfil, insira:

```
CodeCatalyst SAM deploy role

```

- j. Selecione Criar perfil.

Agora você criou um perfil de criação com uma política de confiança e uma política de permissões.

3. ~~Anexe a política de permissões ao perfil de criação, da seguinte maneira:~~

- a. No painel de navegação, selecione Perfis e procure `codecatalyst-SAM-deploy-role`.
- b. Selecione `codecatalyst-SAM-deploy-role` para exibir os detalhes.
- c. Na guia Permissões, escolha Adicionar permissões e Anexar políticas.
- d. Procure `codecatalyst-SAM-deploy-policy`, marque a caixa de seleção e selecione Anexar políticas.

Agora você anexou a política de permissões ao perfil de criação. O perfil de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN do perfil de criação, da seguinte forma:
  - a. No painel de navegação, selecione Perfis.
  - b. Na caixa de pesquisa, insira o nome do perfil que você acabou de criar (`codecatalyst-SAM-deploy-role`).
  - c. Escolha o perfil na lista.

A página Resumo do perfil é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou o perfil de criação com as permissões apropriadas e obteve o ARN.

## Validação de conformidade do Amazon CodeCatalyst

Para saber se um AWS service (Serviço da AWS) está no escopo de programas de conformidade específicos, consulte [Serviços da AWS no escopo por programa de conformidade](#) e selecione o programa de conformidade em que você está interessado. Para obter informações gerais, consulte [Programas de Conformidade da AWS](#).

É possível baixar relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte [Baixar relatórios no AWS Artifact](#).

Sua responsabilidade de conformidade ao usar o Serviços da AWS é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. Para ter mais informações sobre sua responsabilidade pela conformidade ao usar Serviços da AWS, consulte a [documentação da AWS sobre segurança](#).



## Resiliência no Amazon CodeCatalyst

A infraestrutura global da AWS é criada com base em Regiões da AWS e zonas de disponibilidade. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura global da AWS](#). Para saber mais sobre quais dados do CodeCatalyst são replicados em Regiões da AWS, consulte [Proteção de dados na Amazon CodeCatalyst](#).

## Segurança da infraestrutura no Amazon CodeCatalyst

Como um serviço gerenciado, o Amazon CodeCatalyst é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte [Segurança na Nuvem AWS](#). Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança da infraestrutura, consulte [Proteção de Infraestrutura](#) em Pilar de Segurança: AWS Estrutura bem arquitetada.

Você usa chamadas de API publicadas pela AWS para acessar o CodeCatalyst por meio da rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

## Análise de configuração e vulnerabilidade no Amazon CodeCatalyst

A configuração e os controles de TI são uma responsabilidade compartilhada entre a AWS e você, nosso cliente. Para obter mais informações, consulte o AWS [modelo de responsabilidade compartilhada da](#).

## Seus dados e privacidade no Amazon CodeCatalyst

O Amazon CodeCatalyst leva sua privacidade a sério, e a segurança de suas informações é nossa maior prioridade. Você pode ler mais sobre como lidamos com suas informações no [Aviso de privacidade da AWS](#).

Para solicitar e visualizar seus dados, consulte [Solicitação de dados](#) em Referência geral da AWS.

### Exclusão do seu perfil de ID do builder AWS

Excluir seu perfil é uma ação permanente que não pode ser revertida. O processo de exclusão começa imediatamente após você selecionar Excluir. O Amazon CodeCatalyst começa a excluir seu perfil e todas as informações pessoais associadas. Esse processo pode levar cerca de 90 dias para ser concluído.

Quando seu perfil é excluído, você não pode acessar ou recuperar seus dados no Amazon CodeCatalyst. Isso inclui tokens de acesso pessoal, perfis, associações de usuários e quaisquer espaços do Amazon CodeCatalyst dos quais você seja o único membro. Você não pode mais fazer login no Amazon CodeCatalyst.

Para ter informações sobre como excluir seu perfil de ID do builder AWS, consulte [Exclusão do seu ID do builder AWS](#) em Referência geral da AWS.

## Melhores práticas para ações de fluxo de trabalho no Amazon CodeCatalyst

Há uma série de melhores práticas de segurança a serem consideradas no desenvolvimento dos seus fluxos de trabalho no CodeCatalyst. Veja a seguir as diretrizes gerais. Elas não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

### Tópicos

- [Informações confidenciais](#)
- [Termos do licenciamento](#)
- [Código não confiável](#)
- [GitHub Actions](#)

## Informações confidenciais

Não incorpore informações confidenciais em seu YAML. Em vez de incorporar credenciais, chaves ou tokens em seu YAML, recomendamos que você use segredos do CodeCatalyst. Os segredos fornecem uma maneira fácil de armazenar e referenciar informações confidenciais de dentro do seu YAML.

## Termos do licenciamento

Certifique-se de prestar atenção aos termos de licenciamento da ação que você escolher usar.

## Código não confiável

As ações geralmente são módulos independentes e de propósito único que podem ser compartilhados em um projeto, espaço ou na comunidade em geral. Usar código de outras pessoas pode ser um grande ganho de conveniência e eficiência, mas também introduz um novo vetor de ameaças. Revise as seções a seguir para garantir que você esteja seguindo as melhores práticas para manter seus fluxos de trabalho de CI/CD seguros.

## GitHub Actions

As GitHub Actions são de código aberto, desenvolvidas e mantidas pela comunidade. Seguimos o [modelo de responsabilidade compartilhada](#) e consideramos o código-fonte das GitHub Actions como dados do cliente pelos quais você é responsável. As GitHub Actions podem ter acesso a segredos, tokens de repositório, código-fonte, links de conta e seu tempo de computação. Garanta a confiabilidade e a segurança das GitHub Actions que você planeja executar.

Orientações mais específicas e práticas recomendadas de segurança para as GitHub Actions:

- [Fortalecimento da segurança](#)
- [Evitar solicitações pwn](#)
- [Entrada não confiável](#)
- [Como confiar nos componentes](#)

## Entendendo o modelo de CodeCatalyst confiança

O modelo de CodeCatalyst confiança da Amazon CodeCatalyst permite assumir a função de serviço no conectado Conta da AWS. O modelo conecta a função do IAM, os diretores de CodeCatalyst

serviço e o CodeCatalyst espaço. A política de confiança usa a chave de `aws:SourceArn` condição para conceder permissões ao CodeCatalyst espaço especificado na chave de condição. Para obter mais informações sobre essa chave de condição, consulte [aws: SourceArn](#) no Guia do usuário do IAM.

Uma política de confiança é um documento da política JSON no qual você define os princípios que são confiáveis para assumir a função. Uma política de confiança da função é uma política com base em recurso necessária anexada a uma função no IAM. Para ter mais informações, consulte [Termos e conceitos](#) no Guia do usuário do IAM. Para obter detalhes sobre os princípios de serviço para CodeCatalyst, consulte [Princípios de serviço para CodeCatalyst](#).

Na política de confiança a seguir, as entidades principais de serviço listadas no elemento `Principal` recebem permissões da política baseada em recursos, e o bloco `Condition` é usado para limitar o acesso ao recurso com escopo reduzido.

JSON

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

Na política de confiança, os diretores do CodeCatalyst serviço recebem acesso por meio da chave de `aws:SourceArn` condição, que contém o Amazon Resource Name (ARN) para CodeCatalyst a ID do espaço. O ARN usa o formato a seguir:

```
arn:aws:codecatalyst:::space/spaceId/project/*
```

#### Important

Use o ID do espaço somente em chaves de condição, como `aws:SourceArn`. Não use o ID do espaço nas declarações de política do IAM como um ARN de recurso.

Como prática recomendada, reduza as permissões o máximo possível na política.

- Você pode usar o caractere curinga (\*) na chave de condição `aws:SourceArn` para especificar todos os projetos no espaço com `project/*`.
- Você pode especificar permissões em nível de recurso na chave de condição `aws:SourceArn` para um projeto específico no espaço com `project/projectId`.

## Princípios de serviço para CodeCatalyst

Use o elemento `Principal` em uma política JSON baseada em recursos para especificar a entidade principal cujo acesso a um recurso é permitido ou negado. Os principais que podem ser especificados na política de confiança incluem usuários, funções, contas e serviços. Não é possível usar o elemento `Principal` em uma política baseada em identidade; da mesma forma, você não pode identificar um grupo de usuários como um principal em uma política (como uma política baseada em recursos) porque os grupos são relacionados com permissões, não autenticação, e as entidades principais são entidades autenticadas do IAM.

Na política de confiança, você pode especificar Serviços da AWS no `Principal` elemento de uma política baseada em recursos ou em chaves de condição que suportam os princípios. As entidades principais de serviço são definidas pelo serviço. A seguir estão os princípios de serviço definidos para CodeCatalyst:

- `codecatalyst.amazonaws.com` - Esse principal de serviço é usado para uma função que concederá acesso a. CodeCatalyst AWS

- `codecatalyst-runner.amazonaws.com` - Esse principal de serviço é usado para uma função que concederá acesso a recursos em implantações para fluxos de trabalho. CodeCatalyst AWS CodeCatalyst

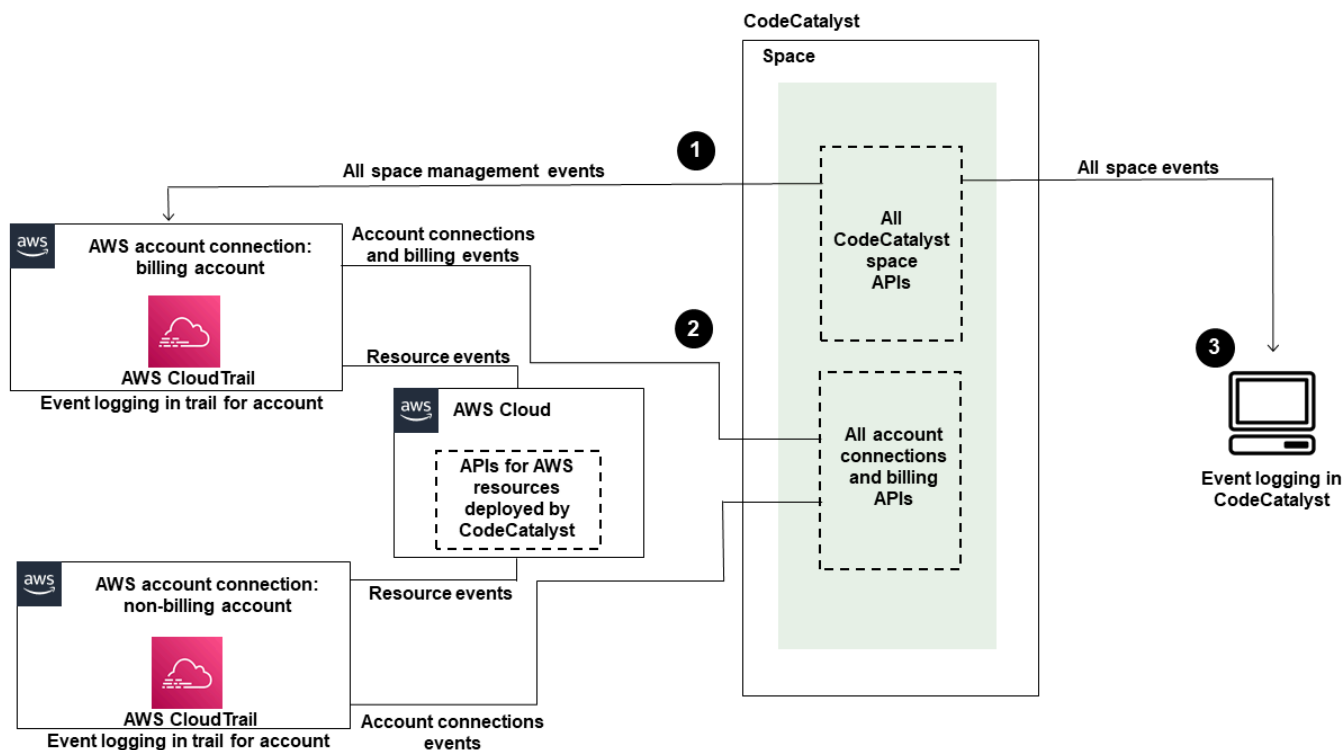
Para obter mais informações, consulte [Elementos de política JSON da AWS : entidade principal](#) no Guia do usuário do IAM.

## Monitoramento de eventos e chamadas de API usando o registro

Na Amazon CodeCatalyst, os eventos de gerenciamento do espaço são coletados AWS CloudTrail e registrados na trilha da conta de cobrança do espaço. CloudTrail o registro é o método principal para gerenciar o registro de CodeCatalyst eventos, e um método secundário é visualizar o login de eventos CodeCatalyst.

Os eventos na conta são registrados com a trilha e o bucket designado que está configurado para a Conta da AWS.

O diagrama a seguir mostra como todos os eventos de gerenciamento do espaço são registrados na conta CloudTrail de cobrança, enquanto os eventos da conta e connections/billing os eventos de AWS recursos são registrados na respectiva CloudTrail conta.



O diagrama ilustra as seguintes etapas:

1. Quando um espaço é criado, uma Conta da AWS é conectada ao espaço e é designado como a conta de cobrança. A trilha usada é a trilha que foi criada CloudTrail para a conta de cobrança, na qual os eventos espaciais são registrados. CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de um CodeCatalyst espaço e entrega os arquivos de log em um bucket do S3 que você especificar. Se a conta de faturamento mudar para outra conta da AWS, os eventos de espaço serão registrados na trilha e no bucket dessa conta. Para obter mais informações sobre eventos de CodeCatalyst gerenciamento registrados por CloudTrail, consulte [Informações do CodeCatalyst no CloudTrail](#).
2. Outras contas conectadas ao espaço, incluindo a conta de cobrança, registram um subconjunto de eventos para conexões de contas e eventos de cobrança. CodeCatalyst fluxos de trabalho que geram eventos de conta para AWS recursos implantados nessa conta também são registrados na trilha e no bucket do. Conta da AWS CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de um CodeCatalyst espaço e entrega os arquivos de log em um bucket do S3 que você especificar. Para obter mais informações sobre eventos de CodeCatalyst gerenciamento registrados por CloudTrail, consulte [Acesso a eventos de logs usando o registro de eventos](#).
3. Você também pode monitorar CodeCatalyst ações em seu espaço dentro de um horário específico no espaço com o [list-event-logs](#) comando usando AWS CLI o. Para obter mais informações, consulte [o Guia de referência de CodeCatalyst APIs da Amazon](#). Você deve ter a função de administrador do espaço para chamar a lista de eventos para CodeCatalyst ações em seu espaço. Para obter mais informações, consulte [Acesso a eventos de logs usando o registro de eventos](#).

 Note

ListEventLogs garante eventos dos últimos 30 dias em um determinado espaço. Você também pode visualizar e recuperar uma lista de eventos de gerenciamento dos últimos 90 dias CodeCatalyst no AWS CloudTrail console visualizando o histórico de eventos ou criando uma trilha para criar e manter um registro de eventos que se estendem por mais de 90 dias. Para obter mais informações, consulte [Trabalhando com o histórico de CloudTrail eventos](#) e [Trabalhando com CloudTrail trilhas](#).

 Note

AWS os recursos que são implantados em contas conectadas para CodeCatalyst fluxos de trabalho não são registrados como parte do CloudTrail registro do espaço. CodeCatalyst Por exemplo, CodeCatalyst os recursos incluem um espaço ou projeto. AWS os recursos incluem um serviço Amazon ECS ou uma função Lambda. Você deve configurar o CloudTrail registro separadamente para cada Conta da AWS local em que os recursos são implantados.

Aqui está um fluxo possível para monitoramento de eventos em CodeCatalyst.

Mary Major é administradora espacial de um CodeCatalyst espaço e visualiza todos os eventos de gerenciamento CodeCatalyst de recursos em nível espacial e em nível de projeto no espaço que está conectado. CloudTrail Veja, [Informações do CodeCatalyst no CloudTrail](#) por exemplo, eventos que estão logados CloudTrail.

Para recursos criados em CodeCatalyst, como ambientes de desenvolvimento, Mary visualiza o histórico de eventos na conta de cobrança do espaço e investiga eventos em que os ambientes de desenvolvimento foram criados pelos membros do projeto. CodeCatalyst O evento fornece o tipo de identidade do IAM do repositório de identidades e as credenciais para o ID do AWS Builder para o usuário que criou o Dev Environment. Para recursos criados AWS quando implantados por fluxos de trabalho em CodeCatalyst, como uma função Lambda para uma implantação sem servidor, o Conta da AWS proprietário pode visualizar o histórico de eventos da trilha associada à ação separada Conta da AWS (que também é uma conta conectada à CodeCatalyst) da ação de implantação do fluxo de trabalho.

Para investigar mais, Mary também pode visualizar eventos para todos CodeCatalyst APIs no espaço usando o [list-event-logs](#) comando no AWS CLI.

## Tópicos

- [Monitoramento de chamadas de API em Contas da AWS usando o registro do AWS CloudTrail](#)
- [Acesso a eventos de logs usando o registro de eventos](#)



## Monitoramento de chamadas de API em Contas da AWS usando o registro do AWS CloudTrail

O Amazon CodeCatalyst é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, um perfil ou um AWS service (Serviço da AWS). O CloudTrail captura chamadas de API realizadas em nome do CodeCatalyst em Contas da AWS conectadas como eventos. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do S3, incluindo eventos para o CodeCatalyst. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos.

O CodeCatalyst suporta as seguintes ações como eventos nos arquivos de log do CloudTrail:

- Os eventos de gerenciamento dos espaços do CodeCatalyst serão registrados na Conta da AWS que é a conta de faturamento designada para o espaço. Para obter mais informações, consulte [Eventos de espaços do CodeCatalyst](#).

### Note

Os eventos de dados para espaços do CodeCatalyst podem ser acessados usando a CLI, conforme detalhado em [Acesso a eventos de logs usando o registro de eventos](#).

- Os eventos dos recursos usados nas ações do fluxo de trabalho do CodeCatalyst que ocorrem em uma Conta da AWS conectada serão registrados como eventos nessa Conta da AWS. Para obter mais informações, consulte [Conexões de contas e eventos de faturamento do CodeCatalyst](#).

### Important

Embora várias contas possam ser associadas a um espaço, o registro do CloudTrail para eventos em espaços e projetos do CodeCatalyst se aplica somente à conta de faturamento.

A conta de faturamento do espaço é sua Conta da AWS que é cobrada pelos recursos do CodeCatalyst além do nível gratuito da AWS. Várias contas podem ser conectadas a um espaço, enquanto somente uma conta pode ser a conta de faturamento designada. A conta de faturamento ou contas adicionais conectadas para o espaço podem ter perfis do IAM que são usadas para implantar recursos e infraestrutura da AWS, como um cluster Amazon ECS ou um bucket do S3,

a partir dos fluxos de trabalho do CodeCatalyst. Você pode usar o YAML do fluxo de trabalho para identificar a Conta da AWS em que você implantou.

 Note

Os recursos da AWS que são implantados em contas conectadas para fluxos de trabalho do CodeCatalyst não são registrados como parte do registro do CloudTrail para o espaço do CodeCatalyst. Por exemplo, os recursos do CodeCatalyst incluem um espaço ou projeto. Os recursos da AWS incluem um serviço do Amazon ECS ou uma função do Lambda. O registro do CloudTrail deve ser configurado separadamente para cada Conta da AWS local em que os recursos são implantados.

O registro do CodeCatalyst em contas conectadas inclui as seguintes considerações:

- O acesso aos eventos do CloudTrail é gerenciado com o IAM na conta conectada e não no CodeCatalyst.
- Conexões de terceiros, como a vinculação a um repositório do GitHub, resultarão na gravação de nomes de recursos de terceiros nos logs do CloudTrail.

 Note

O registro do CloudTrail para eventos do CodeCatalyst está no nível do espaço e não isola os eventos pelos limites do projeto.

Para obter mais informações sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

 Note

Esta seção descreve o registro do CloudTrail para todos os eventos registrados em um espaço do CodeCatalyst e as Contas da AWS conectadas ao CodeCatalyst. Além disso, para revisar todos os eventos registrados em um espaço do CodeCatalyst, você também pode usar a AWS CLI e o comando `aws codecatalyst list-event-logs`. Para obter mais informações, consulte [Acesso a eventos de logs usando o registro de eventos](#).

## Eventos de espaços do CodeCatalyst

As ações no CodeCatalyst para gerenciar recursos em nível de espaço e em nível de projeto são registradas em log na conta de faturamento do espaço. Para registro em log do CloudTrail em um espaço do CodeCatalyst, os eventos são registrados em log com as considerações a seguir.

- Os eventos do CloudTrail se aplicam a todo o espaço e não têm como escopo nenhum projeto único.
- Quando você conecta uma Conta da AWS a um espaço do CodeCatalyst, os eventos registráveis para conexões da conta serão registrados nessa Conta da AWS. Depois de habilitar essa conexão, você não poderá desativá-la.
- Quando você conecta uma Conta da AWS a um espaço do CodeCatalyst e a designa como a conta de faturamento do espaço, os eventos são registrados nessa Conta da AWS. Depois de habilitar essa conexão, você não poderá desativá-la.

Eventos para recursos no nível do espaço e no nível do projeto são registrados somente na conta de faturamento. Para alterar a conta de destino do CloudTrail, atualize a conta de faturamento no CodeCatalyst. No início do próximo ciclo de faturamento mensal, a alteração entra em vigor para a nova conta de faturamento no CodeCatalyst. Depois disso, a conta de destino do CloudTrail é atualizada.

Veja a seguir exemplos de eventos na AWS relacionados a ações no CodeCatalyst para gerenciar recursos no nível do espaço e no nível do projeto. As seguintes APIs são lançadas por meio do SDK e da CLI. Os eventos serão registrados na Conta da AWS especificada como conta de faturamento para o espaço do CodeCatalyst.

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)

- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

## Conexões de contas e eventos de faturamento do CodeCatalyst

Veja a seguir exemplos de eventos na AWS relacionados a ações no CodeCatalyst para conexões de contas ou faturamento:

- `AcceptConnection`
- `AssociateIAMRoletoConnection`
- `DeleteConnection`
- `DissassociateIAMRolefromConnection`
- `GetBillingAuthorization`
- `GetConnection`
- `GetPendingConnection`
- `ListConnections`
- `ListIAMRolesforConnection`
- `PutBillingAuthorization`
- `RejectConnection`

## Informações do CodeCatalyst no CloudTrail

O CloudTrail é habilitado em uma Conta da AWS quando ela é criada. Quando você conecta essa Conta da AWS a um espaço do CodeCatalyst, os eventos desse espaço que ocorrem nessa Conta da AWS são registrados em logs do CloudTrail nessa conta da AWS. Os eventos registráveis no CodeCatalyst são registrados como eventos do CloudTrail nos logs do CloudTrail na conta

conectada e no Histórico de eventos no console do CloudTrail, junto com outros eventos da AWS registráveis nessa conta.

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 por um usuário com o ID do builder AWS.
- Se a solicitação foi feita com credenciais de usuário-raiz ou usuário do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

## Acesso a eventos do CloudTrail

Para ter um registro contínuo de eventos em sua Conta da AWS, incluindo os eventos de atividade do CodeCatalyst na Conta da AWS, crie uma trilha. Uma trilha permite que o CloudTrail forneça arquivos de log para um bucket do S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na partição da AWS e fornece os arquivos de log ao bucket do S3 que você especificar. Além disso, é possível configurar outros produtos da AWS para analisar mais profundamente e agir sobre os dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configuração notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [Receber arquivos de log do CloudTrail de várias contas](#)

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte, e inclui informações sobre a ação solicitada, data e hora da ação, parâmetros da solicitação e assim por diante. Os arquivos de log do

CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública, portanto, não são exibidos em uma ordem específica.

## Exemplo de evento de conexões da conta do CodeCatalyst na AWS

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `ListConnections`. Para uma Conta da AWS que está conectada ao espaço, `ListConnections` é usado para visualizar todas as conexões da conta com o CodeCatalyst para essa Conta da AWS. O evento será registrado na Conta da AWS local especificada em `accountId` e o valor de `arn` será o nome do recurso da Amazon (ARN) do perfil usado para a ação.

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAI44QH8DHBEXAMPLE",
 "arn": "role-ARN",
 "accountId": "account-ID",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAI44QH8DHBEXAMPLE",
 "arn": "role-ARN",
 "accountId": "account-ID",
 "userName": "user-name"
 },
 "webIdFederationData": {},
 "attributes": {
 "creationDate": "2022-09-06T15:04:31Z",
 "mfaAuthenticated": "false"
 }
 }
 },
 "eventTime": "2022-09-06T15:08:43Z",
 "eventSource": "account-ID",
 "eventName": "ListConnections",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.168.0.1",
 "userAgent": "aws-cli/1.18.147 Python/2.7.18 Linux/5.4.207-126.363.amzn2int.x86_64
 boto3/1.18.6",
 "requestParameters": null,
 "responseElements": null,
```

```
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "account-ID",
"eventCategory": "Management"
}
```

## Exemplo de evento de recurso do projeto do CodeCatalyst na AWS

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateDevEnvironment`. Uma Conta da AWS que esteja conectada ao espaço e seja a conta de faturamento designada para o espaço é usada para eventos no nível do projeto no espaço, como a criação de um Ambiente de Desenvolvimento.

Em `userIdentity`, no campo `accountId`, está o ID da conta do IAM Identity Center (432677196278) que hospeda o banco de identidades para todas as identidades de ID do builder AWS. Esse ID da conta contém as seguintes informações sobre o usuário do CodeCatalyst do evento.

- O campo `type` indica o tipo de entidade do IAM para a solicitação. Para eventos do CodeCatalyst para recursos de espaço e de projeto, esse valor é `IdentityCenterUser`. O campo `accountId` especifica a conta proprietária da entidade que foi usada para obter credenciais.
- O campo `userId` contém o identificador de ID do builder AWS para o usuário.
- O campo `identityStoreArn` contém o ARN do perfil da conta e do usuário do repositório de identidades.

O campo `recipientAccountId` contém o ID da conta de faturamento do espaço, com um valor de exemplo aqui de 111122223333.

Para obter mais informações, consulte [Elemento `userIdentity` do CloudTrail](#).

```
{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IdentityCenterUser",
 "accountId": "432677196278",
 "onBehalfOf": {
```

```
{
 "userId": "user-ID",
 "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7",
},
"credentialId": "ABCDefGhiJKLMn11Lmn_1AbCDEFGHijKLmnOPqrs11abEXAMPLE"
},
"eventTime": "2023-05-18T17:10:50Z",
"eventSource": "codecatalyst.amazonaws.com",
"eventName": "CreateDevEnvironment",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.168.0.1",
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0",
"requestParameters": {
 "spaceName": "MySpace",
 "projectName": "MyProject",
 "ides": [{
 "runtime": "public.ecr.aws/q6e8p2q0/cloud9-ide-runtime:2.5.1",
 "name": "Cloud9"
 }],
 "instanceType": "dev.standard1.small",
 "inactivityTimeoutMinutes": 15,
 "persistentStorage": {
 "sizeInGiB": 16
 }
},
"responseElements": {
 "spaceName": "MySpace",
 "projectName": "MyProject",
 "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 "
},
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventCategory": "Management"
}
```



**Note**

Em alguns eventos, o agente do usuário pode não ser conhecido. Nesse caso, o CodeCatalyst fornecerá um valor de Unknown no campo userAgent no evento do CloudTrail.

## Consulta das trilhas de eventos do CodeCatalyst

Você pode criar e gerenciar consultas para os logs do CloudTrail usando uma tabela de consultas no Amazon Athena. Para ter mais informações sobre a criação de uma consulta, veja [Consultar os logs do AWS CloudTrail](#) no Manual do usuário do Amazon Athena.

## Acesso a eventos de logs usando o registro de eventos

Quando os usuários realizam ações no Amazon CodeCatalyst, essas ações são registradas como eventos. Você pode usar a AWS CLI para visualizar logs de eventos em um espaço em um período especificado. Você pode visualizar esses eventos para analisar as ações realizadas no espaço, incluindo a data e a hora das ações, o nome do usuário que realizou a ação e o endereço IP em que o usuário fez a solicitação.

**Note**

Os eventos de gerenciamento de um espaço do CodeCatalyst são registrados no CloudTrail para a conta de faturamento conectada. Para ter mais informações sobre os eventos de gerenciamento do CodeCatalyst que são registrados pelo CloudTrail, consulte [Informações do CodeCatalyst no CloudTrail](#).

Para visualizar um log de eventos de um espaço, você deve ter instalado e configurado a AWS CLI com um perfil para o CodeCatalyst e deve ter o perfil de Administrador do espaço. Para obter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#) e [Perfil de administrador do espaço](#).

**Note**

Para visualizar o registro de eventos que ocorrem em nome do CodeCatalyst nas Contas da AWS conectadas, ou para visualizar o registro de eventos de espaço ou recursos do projeto na conta de faturamento conectada, você pode usar o AWS CloudTrail. Para obter mais

informações, consulte [Monitoramento de chamadas de API em Contas da AWS usando o registro do AWS CloudTrail](#).

1. Abra um terminal ou uma linha de comando e execute o comando `aws codecatalyst list-event-logs`, especificando:
  - O nome do espaço com a opção **--space-name**.
  - A data e a hora em que você deseja começar a revisar os eventos, no formato de carimbo de data/hora universal coordenada (UTC), conforme especificado na [RFC 3339](#), com a opção **--start-time**.
  - A data e a hora em que você deseja parar de revisar os eventos, no formato de carimbo de data/hora universal coordenada (UTC), conforme especificado na [RFC 3339](#), com a opção **--end-time**.
  - (Opcional) O número máximo de resultados a serem retornados em uma única resposta, com a opção **--max-results**. Se o número de resultados for maior que o número especificado, a resposta incluirá um elemento `nextToken` que você pode usar para retornar os próximos resultados.
  - (Opcional) Limite os resultados a um tipo de evento específico que você deseja retornar, com a opção **--event-name**.

Este exemplo retorna eventos de logs no espaço chamado *ExampleCorp* do período *2022-11-30* a *2022-12-01*, e que no máximo *2* eventos sejam retornados na resposta.

```
aws codecatalyst list-event-logs --space-name ExampleCorp --start-time 2022-11-30
--end-time 2022-12-01 --event-name list-event-logs --max-results 2
```

2. Se eventos tiverem ocorrido nesse período, o comando retornará resultados semelhantes a:

```
{
 "nextToken": "EXAMPLE",
 "items": [
 {
 "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
 "eventName": "listEventLogs",
 "eventType": "AwsApiCall",
 "eventCategory": "MANAGEMENT",
 "eventSource": "manage",
```

```

 "eventTime": "2022-12-01T22:47:24.605000+00:00",
 "operationType": "READONLY",
 "userIdentity": {
 "userType": "USER",
 "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111"
 "userName": "MaryMajor"
 },
 "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
 "requestPayload": {
 "contentType": "application/json",
 "data": "{\"spaceName\":\"ExampleCorp\",\"startTime\":
\"2022-12-01T00:00:00Z\",\"endTime\":\"2022-12-10T00:00:00Z\",\"maxResults\":
\"2\"}"
 },
 "sourceIpAddress": "127.0.0.1",
 "userAgent": "aws-cli/2.9.0 Python/3.9.11 Darwin/21.3.0 exe/x86_64
prompt/off command/codecatalyst.list-event-logs"
 },
 {
 "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
 "eventName": "createProject",
 "eventType": "AwsApiCall",
 "eventCategory": "MANAGEMENT",
 "eventSource": "manage",
 "eventTime": "2022-12-01T09:15:32.068000+00:00",
 "operationType": "MUTATION",
 "userIdentity": {
 "userType": "USER",
 "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111",
 "userName": "MaryMajor"
 },
 "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
 "requestPayload": {
 "contentType": "application/json",
 "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject
\", \"displayName\":\"MyFirstProject\"}"
 },
 "responsePayload": {
 "contentType": "application/json",
 "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject
\", \"displayName\":\"MyFirstProject\", \"id\":\"a1b2c3d4-5678-90ab-cdef-
EXAMPLE4444\"}"
 },
 "sourceIpAddress": "192.0.2.23",

```

```

 "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0)
 Gecko/20100101 Firefox/102.0"
 }
]
}

```

3. Execute o comando `list-event-logs` novamente com a opção `--next-token` e o valor do token retornado para recuperar o próximo conjunto de eventos registrados que correspondam à solicitação.

## Cotas para identidade, permissão e acesso em CodeCatalyst

A tabela a seguir descreve cotas e limites de identidade, permissão e acesso na Amazon CodeCatalyst. Para obter mais informações sobre cotas na Amazon CodeCatalyst, consulte [Cotas para o CodeCatalyst](#).

| Recurso                                                               | Informações                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aliases em CodeCatalyst                                               | <p>Qualquer combinação de caracteres permitidos entre 3 e 100 caracteres e deve começar com uma letra. Caracteres válidos: A-Z, a-z e 0-9.</p> <p>Aliases não podem:</p> <ul style="list-style-type: none"> <li>• conter menos de 3 caracteres</li> <li>• conter espaços ou qualquer um dos seguintes caracteres: ? ^ * [ \ ~ :</li> </ul> |
| Número máximo de convites enviados por um usuário por dia             | 500                                                                                                                                                                                                                                                                                                                                        |
| Número máximo de convites enviados para um endereço de e-mail por dia | 25                                                                                                                                                                                                                                                                                                                                         |
| Número máximo de tokens de acesso pessoal (PAT) por usuário           | 100                                                                                                                                                                                                                                                                                                                                        |

| Recurso                                                                                                                             | Informações                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Número máximo de conexões pessoais para cada identidade de usuário (CodeCatalyst aliases) em todos os espaços, por tipo de provedor | 1                                                                                                                                                                                                                                           |
| Senhas em CodeCatalyst                                                                                                              | Qualquer combinação dos caracteres permitidos entre 8 e 64 caracteres. Caracteres válidos: A-Z, a-z e 0-9. A senha pode incluir os seguintes caracteres não alfanuméricos: ( ~ ! @ # \$ % ^ & * _ - + = `   \ { } [ ] : ; " ' < > , . ? / ) |
| Nomes PAT em CodeCatalyst                                                                                                           | Qualquer combinação dos caracteres permitidos entre 1 e 100 caracteres                                                                                                                                                                      |
| Tempo até que o convite de um membro do projeto expire                                                                              | Expira após 24 horas                                                                                                                                                                                                                        |
| Tempo até que o convite de um membro do espaço expire                                                                               | Expira após 24 horas                                                                                                                                                                                                                        |
| Tempo até que a verificação do endereço de e-mail expire                                                                            | Expira 10 minutos após o envio                                                                                                                                                                                                              |

## Solução de problemas

Esta seção pode ajudá-lo a solucionar alguns problemas comuns que você pode encontrar ao acessar seu CodeCatalyst perfil da Amazon.

### Problemas de inscrição

Você pode encontrar alguns problemas ao se inscrever. Temos algumas soluções.

#### Meu endereço de e-mail já está em uso

Se o e-mail inserido já estiver em uso e você reconhecê-lo como seu, talvez já tenha um perfil conosco. Faça login com essa identidade existente. Se você não possuir o e-mail existente, inscreva-se com um e-mail diferente e não utilizado.

## Não consigo concluir a verificação de e-mail

Se você não recebeu o e-mail de verificação

1. Verifique sua pasta de spam, lixo eletrônico e itens excluídos.

### Note

Esse e-mail de verificação vem do endereço `no-reply@signin.aws` ou `no-reply@login.awsapps.com`. Recomendamos que você configure seu sistema de e-mail para que ele aceite e-mails desses remetentes e não os trate como lixo eletrônico ou spam.

2. Aguarde 5 minutos e atualize a caixa de entrada. Confira sua pasta de spam, de lixo eletrônico e itens excluídos novamente.
3. Se você ainda não recebeu o e-mail de verificação, selecione Reenviar código. Se você já saiu dessa página, reinicie seu fluxo de trabalho para se inscrever [na Amazon CodeCatalyst](#).

## Minha senha não satisfaz os requisitos mínimos

Para sua segurança, a senha deve incluir de 8 a 20 caracteres, letras maiúsculas e minúsculas e números.

## Problemas de login

### Esqueci minha senha

Siga as etapas em [Esqueci minha senha](#).

### Minha senha não está funcionando

Você deve seguir estes requisitos sempre que definir ou alterar sua senha:

- As senhas diferenciam maiúsculas de minúsculas.
- As senhas devem ter entre 8 e 64 caracteres com letras maiúsculas e minúsculas, números e pelo menos um caractere não alfanumérico.
- Não é possível reutilizar as três últimas senhas.

## Não consigo habilitar a MFA

Para habilitar a MFA, adicione um ou mais dispositivos de MFA ao seu perfil seguindo as etapas em [Configure seu AWS Builder ID para fazer login com autenticação multifator \(MFA\)](#).

## Não consigo adicionar um dispositivo de MFA

Caso descubra que não é possível adicionar outro dispositivo de MFA, talvez tenha atingido o limite de dispositivos de MFA que podem ser registrados. Talvez seja necessário remover um dispositivo de MFA existente antes de adicionar um novo.

## Não consigo remover um dispositivo de MFA

Caso pretenda desativar a MFA, prossiga com a remoção do dispositivo de MFA seguindo as etapas em [Exclusão de um dispositivo de MFA](#). No entanto, se quiser manter a MFA ativada, você deve adicionar outro dispositivo de MFA antes de tentar excluir um existente. Para obter mais informações sobre a adição de outro dispositivo MFA, consulte [Como registrar um dispositivo para uso com autenticação multifator](#).

## Problemas ao sair

### Não consigo encontrar onde sair

No canto superior direito da página, selecione Sair.

### Sair não me desconecta completamente

O sistema foi projetado para sair imediatamente, mas o desligamento completo pode levar até uma hora.

## Recebo um erro de perfil não existe para um fluxo de trabalho com falha

Problema: depois de criar um projeto a partir da aplicação web ou do esquema sem servidor, o fluxo de trabalho falha com o seguinte erro:

CLIENT\_ERROR: o perfil não existe

Solução possível: depois de configurar um perfil do IAM com as permissões para executar seu fluxo de trabalho e adicionar o perfil do IAM ao YAML do fluxo de trabalho, o fluxo de trabalho ainda

falhará porque talvez seja necessário adicionar o perfil do IAM à conexão da sua conta. Adicione o perfil do IAM à conexão da conta do seu espaço, conforme detalhado em [Adicionar perfis do IAM às conexões da conta](#).

## Recebo um erro de perfil para um fluxo de trabalho com falha

Problema: depois de criar um projeto a partir da aplicação web ou do esquema sem servidor, o fluxo de trabalho falha com o seguinte erro:

CLIENT\_ERROR: o perfil não está configurado corretamente ou não existe

Solução possível: o espaço em que o projeto foi criado pode precisar configurar uma Conta da AWS conexão ou concluir uma solicitação de conexão da conta. Se seu espaço já tiver uma conexão de Conta da AWS ativa, crie e adicione um perfil do IAM com permissões para executar ações de fluxo de trabalho. Adicione o perfil do IAM à conexão da sua conta, conforme detalhado em [Adicionar perfis do IAM às conexões da conta](#).

Solução possível: se o projeto foi criado sem especificar uma conexão, a conexão da conta precisará ser associada ao ambiente de implantação. Se seu espaço já tiver uma Conta da AWS conexão ativa e uma função do IAM adicionada, você deverá adicionar a conexão da conta com a função do IAM ao seu ambiente de implantação, conforme detalhado em [Adição da conexão da conta e dos perfis do IAM ao seu ambiente de implantação](#).

## Preciso atualizar o perfil do IAM no fluxo de trabalho do projeto

Se a Conta da AWS conexão estiver completamente configurada e a função do IAM for criada e adicionada à conexão da conta, você poderá atualizar a função do IAM no fluxo de trabalho do seu projeto.

1. Escolha a opção CI/CD e escolha seu fluxo de trabalho. Selecione o botão YAML.
2. Escolha Editar.
3. No campo `ActionRoleArn:`, substitua o ARN do perfil do IAM pelo ARN do perfil do IAM atualizado. Escolha Validar.
4. Selecione Confirmar.

O fluxo de trabalho é iniciado automaticamente se estiver na ramificação principal. Caso contrário, para executar novamente o fluxo de trabalho, selecione Executar.



## Tenho uma solicitação de revisão para minha GitHub conta depois de criar uma conexão pessoal

Depois de criar uma conexão pessoal com GitHub, o CodeCatalyst aplicativo é instalado na sua GitHub conta como um GitHub aplicativo. Se houver certos recursos CodeCatalyst que exijam permissões de leitura ou gravação atualizadas, talvez seja necessário acessar sua GitHub conta para atualizar as permissões no aplicativo instalado.

1. Faça login GitHub e navegue até as configurações da sua conta para os aplicativos instalados. Escolha o ícone do seu perfil, selecione Configurações e, depois, escolha Aplicações.
2. Na guia GitHub Aplicativos instalados, na lista de aplicativos instalados, visualize o aplicativo instalado para CodeCatalyst. Um link de Solicitação de revisão será exibido se houver permissões para revisar.
3. Escolha o link e confirme suas credenciais quando solicitado. Insira suas credenciais e selecione Verificar.
4. Aceite as novas permissões, indique os repositórios nos quais você deseja aplicar as permissões e selecione Salvar.

## Como preencho um formulário de suporte?

Você pode acessar a [Amazon CodeCatalyst](#) ou preencher um [formulário de Support Feedback](#). Na seção Solicitar informações, em Como podemos ajudá-lo, inclua que você é um CodeCatalyst cliente da Amazon. Forneça o máximo de detalhes possível para que possamos resolver seu problema da maneira mais eficiente.

# Adicione funcionalidade a projetos com extensões no CodeCatalyst

A Amazon CodeCatalyst inclui extensões que ajudam você a adicionar funcionalidades e integrar-se com produtos externos CodeCatalyst. Com extensões do CodeCatalyst catálogo, as equipes podem personalizar suas experiências no CodeCatalyst.

## Tópicos

- [Extensões de terceiros disponíveis](#)
- [Conceitos de extensões](#)
- [Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst](#)
- [Instalação de uma extensão em um espaço](#)
- [Desinstalação de uma extensão em um espaço](#)
- [Conectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst](#)
- [Desconectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst](#)
- [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#)
- [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#)
- [Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst](#)
- [Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros](#)
- [Restrição do acesso ao IP com provedores de repositório de terceiros](#)
- [Bloqueio de mesclagens de terceiros quando os fluxos de trabalho falham](#)
- [Vinculando problemas do Jira a CodeCatalyst pull requests](#)
- [Visualizando CodeCatalyst eventos em edições do Jira](#)

## Extensões de terceiros disponíveis

Você pode adicionar uma funcionalidade específica ao seu CodeCatalyst projeto, dependendo da extensão com a qual escolher integrar os recursos.

### Integrando GitHub repositórios em CodeCatalyst

GitHub é um serviço baseado em nuvem que ajuda os desenvolvedores a armazenar e gerenciar seus códigos. A extensão GitHub repositories permite que você use GitHub repositórios vinculados em projetos da Amazon CodeCatalyst . Você também pode vincular GitHub repositórios ao criar um novo CodeCatalyst projeto. Para obter mais informações, consulte [Criar um projeto com um repositório de terceiros vinculado](#).

#### Note

- Você não pode usar GitHub repositórios vazios ou arquivados com CodeCatalyst projetos.
- A extensão GitHub repositories não é compatível com os repositórios do GitHub Enterprise Server.

Depois de instalar e configurar a extensão GitHub dos repositórios, você poderá:

- Visualize seus GitHub repositórios na lista de repositórios de origem em CodeCatalyst
- Armazene e gerencie arquivos de definição de fluxo de trabalho em seus GitHub repositórios
- Crie, leia, atualize e exclua arquivos armazenados em GitHub repositórios vinculados a partir de ambientes de CodeCatalyst desenvolvimento
- Armazene e indexe arquivos dos GitHub repositórios vinculados no CodeCatalyst
- Crie CodeCatalyst projetos com repositórios existentes de contas conectadas GitHub
- Crie um GitHub repositório com código gerado por um blueprint ao criar um projeto com um blueprint ou adicionar um blueprint
- O CodeCatalyst fluxo de trabalho Start é executado automaticamente quando o código é enviado para GitHub repositórios vinculados ou quando as pull requests são criadas, modificadas ou fechadas em repositórios vinculados GitHub
- Use arquivos de origem GitHub do repositório vinculado em CodeCatalyst fluxos de trabalho
- Leia e execute GitHub ações em CodeCatalyst fluxos de trabalho

- Envie os status CodeCatalyst de execução do fluxo de trabalho para GitHub repositórios vinculados e bloqueie as mesclagens de GitHub pull request com base nos status de confirmação

## Integrando repositórios do Bitbucket em CodeCatalyst

O Bitbucket é um serviço baseado em nuvem que ajuda os desenvolvedores a armazenar e gerenciar o código. A extensão de repositórios do Bitbucket permite que você use repositórios vinculados do Bitbucket em projetos da Amazon. CodeCatalyst Você também pode vincular repositórios do Bitbucket ao criar um novo CodeCatalyst projeto. Para obter mais informações, consulte [Criar um projeto com um repositório de terceiros vinculado](#).

### Note

- Você não pode usar repositórios do Bitbucket vazios ou arquivados com projetos. CodeCatalyst
- A extensão de Repositórios do Bitbucket não é compatível com os repositórios do Bitbucket Data Center.

Depois de instalar e configurar a extensão de Repositórios do Bitbucket, você poderá:

- Visualize seus repositórios do Bitbucket na lista de repositórios de origem em CodeCatalyst
- Armazenar e gerenciar arquivos de definição de fluxo de trabalho nos repositórios do Bitbucket.
- Crie, leia, atualize e exclua arquivos armazenados em repositórios vinculados do Bitbucket a partir CodeCatalyst de ambientes de desenvolvimento
- Crie CodeCatalyst projetos com repositórios existentes de contas Bitbucket conectadas
- Armazene e indexe arquivos dos repositórios vinculados do Bitbucket no CodeCatalyst
- Criar um repositório do Bitbucket com código gerado por um esquema ao criar um projeto com um esquema ou adicionar um esquema
- O CodeCatalyst fluxo de trabalho Start é executado automaticamente quando o código é enviado para repositórios vinculados do Bitbucket ou quando as pull requests são criadas, modificadas ou fechadas em repositórios vinculados do Bitbucket
- Use seus arquivos de origem do repositório Bitbucket vinculado em fluxos de trabalho CodeCatalyst

- Envie os status CodeCatalyst de execução do fluxo de trabalho para repositórios vinculados do Bitbucket e bloqueie as mesclagens de pull request do Bitbucket com base nos status de confirmação

## Integrando GitLab repositórios em CodeCatalyst

GitLab é um serviço baseado em nuvem que ajuda os desenvolvedores a armazenar e gerenciar seus códigos. A extensão GitLab repositories permite que você use repositórios de GitLab projetos vinculados em projetos da Amazon CodeCatalyst . Você também pode vincular repositórios de GitLab projetos ao criar um novo CodeCatalyst projeto. Para obter mais informações, consulte [Criar um projeto com um repositório de terceiros vinculado](#).

### Note

- Você não pode usar repositórios de projetos vazios ou arquivados com GitLab CodeCatalyst projetos.
- A extensão de GitLab repositórios não é compatível com repositórios GitLab autogerenciados.

Depois de instalar e configurar a extensão GitLab dos repositórios, você poderá:

- Visualize os repositórios GitLab do seu projeto na lista de repositórios de origem em CodeCatalyst
- Armazene e gerencie arquivos de definição de fluxo de trabalho nos repositórios GitLab do seu projeto.
- Crie, leia, atualize e exclua arquivos armazenados em repositórios de GitLab projetos vinculados a partir de ambientes de CodeCatalyst desenvolvimento
- Crie CodeCatalyst projetos com repositórios existentes de usuários conectados GitLab
- Armazene e indexe arquivos dos repositórios de GitLab projetos vinculados no CodeCatalyst
- Crie um repositório de GitLab projetos com código gerado por um blueprint ao criar um projeto com um blueprint ou adicionar um blueprint
- O CodeCatalyst fluxo de trabalho Start é executado automaticamente quando o código é enviado para repositórios de GitLab projetos vinculados ou quando as pull requests são criadas, modificadas ou fechadas em repositórios de GitLab projetos vinculados

- Use os arquivos de origem GitLab do repositório de projetos vinculado em CodeCatalyst fluxos de trabalho
- Envie status CodeCatalyst de execução do fluxo de trabalho para repositórios de GitLab projetos vinculados e bloqueie solicitações de GitLab mesclagem com base nos status de confirmação

## Integrando problemas do Jira no CodeCatalyst

O Jira é uma aplicação de software que ajuda as equipes de desenvolvimento Agile a planejar, atribuir, rastrear, gerar relatórios e gerenciar o trabalho. A extensão do Jira Software permite que você use projetos do Jira em projetos da Amazon CodeCatalyst .

### Note

CodeCatalyst só é compatível com o Jira Software Cloud.

Depois de instalar e configurar a extensão do Jira Software para um CodeCatalyst projeto da Amazon, você poderá:

- Acesse projetos do Jira CodeCatalyst vinculando-os a projetos CodeCatalyst
- Atualize os problemas do Jira com CodeCatalyst pull requests
- Visualize o status e as execuções do fluxo de trabalho de CodeCatalyst pull requests vinculados em problemas do Jira

## Conceitos de extensões

Aqui estão alguns conceitos e termos que você deve conhecer ao trabalhar com extensões no CodeCatalyst.

## Extensões

Uma extensão é um complemento que você pode instalar em seu CodeCatalyst espaço para adicionar novas funcionalidades aos seus projetos e integrá-lo a serviços externos. CodeCatalyst As extensões podem ser pesquisadas e instaladas a partir do CodeCatalyst catálogo.

## CodeCatalyst catálogo

O CodeCatalyst catálogo é uma lista centralizada de todas as extensões disponíveis em CodeCatalyst. Você pode navegar no CodeCatalyst catálogo para encontrar extensões que podem melhorar as experiências da sua equipe em áreas CodeCatalyst como fonte, fluxos de trabalho e muito mais.

## Conexão e vínculo

Dependendo dos recursos de terceiros que você deseja usar ou gerenciar, você precisa conectar sua GitHub conta, o espaço de trabalho do Bitbucket ou o projeto do Jira. Em seguida, você precisa vincular seu GitHub repositório, repositório Bitbucket ou projeto Jira ao seu projeto. CodeCatalyst

- GitHub repositórios: Conecte a GitHub conta e, em seguida, vincule os GitHub repositórios.
- Repositórios do Bitbucket: conecte o espaço de trabalho do Bitbucket e, depois, vincule os repositórios do Bitbucket.
- GitLab repositórios: Conecte o GitLab usuário e, em seguida, vincule os repositórios GitLab do projeto.
- Software Jira: conecte o site do Jira e, depois, vincule os projetos do Jira.

## Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst

Este tutorial apresenta um passo a passo das três tarefas abaixo:

1. Instale os GitHub repositórios, repositórios do Bitbucket, GitLab repositórios ou a extensão do Jira Software. Você é solicitado em um site externo a se conectar e CodeCatalyst fornecer acesso aos seus recursos de terceiros, o que é feito como parte da próxima etapa.

### Important

Para instalar os GitHub repositórios, os repositórios do Bitbucket, os GitLab repositórios ou a extensão do Jira Software no seu CodeCatalyst espaço, você deve estar conectado com uma conta que tenha a função de administrador do espaço no espaço.

2. Conecte sua GitHub conta, espaço de trabalho do Bitbucket, GitLab usuário ou site do Jira a. CodeCatalyst

 Important

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket, GitLab usuário ou site do Jira ao seu CodeCatalyst espaço, você deve ser o administrador da fonte terceirizada e o administrador do Space. CodeCatalyst

 Important

Depois de instalar uma extensão de repositório, todos os repositórios aos quais você vincular CodeCatalyst terão seu código indexado e armazenado. CodeCatalyst Isso tornará o código pesquisável em CodeCatalyst. Para entender melhor a proteção de dados do seu código ao usar repositórios vinculados CodeCatalyst, consulte [Proteção de dados](#) no Guia do CodeCatalyst usuário da Amazon.

 Note

Se você estiver usando uma conexão com uma GitHub conta, deverá criar uma conexão pessoal para estabelecer um mapeamento de identidade entre sua CodeCatalyst identidade e sua GitHub identidade. Para obter mais informações, consulte [Conexões pessoais](#) e [Acessando GitHub recursos com conexões pessoais](#).

3. Vincule seu GitHub repositório, repositório Bitbucket, repositório de projetos ou GitLab projeto Jira ao seu projeto. CodeCatalyst

 Important

- Embora você possa vincular um GitHub repositório, um repositório do Bitbucket ou um repositório GitLab do projeto como colaborador, você só pode desvincular um repositório de terceiros como administrador do Space ou administrador do projeto. Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).
- Para vincular seu projeto do Jira ao seu CodeCatalyst projeto, você deve ser administrador do CodeCatalyst Space ou administrador CodeCatalyst do projeto.



 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

 Note

- Um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados, GitHub repositórios Bitbucket ou repositórios de projetos com GitLab projetos. CodeCatalyst
- Você não pode vincular um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto que tenha o mesmo nome de um repositório em um projeto. CodeCatalyst
- A extensão de GitHub repositórios não é compatível com os repositórios do GitHub Enterprise Server.
- A extensão de Repositórios do Bitbucket não é compatível com os repositórios do Bitbucket Data Center.
- A extensão GitLab repositories não é compatível com repositórios de projetos GitLab autogerenciados.
- Não é possível usar os recursos Escrever descrição para mim ou Resumir comentários com repositórios vinculados. Esses recursos só estão disponíveis em pull requests em CodeCatalyst.
- Um CodeCatalyst projeto só pode ser vinculado a um projeto do Jira. Um projeto do Jira pode ser vinculado a vários CodeCatalyst projetos.

Você também pode instalar os GitHub repositórios, repositórios do Bitbucket, extensão de GitLab repositórios, conectar-se à sua GitHub conta, espaço de trabalho do Bitbucket ou GitLab usuário e vincular repositórios de terceiros ao criar um novo projeto. CodeCatalyst Para obter mais informações, consulte [Criar um projeto com um repositório de terceiros vinculado](#).

## Tópicos

- [Etapa 1: instalar uma extensão de terceiros do CodeCatalyst catálogo](#)
- [Etapa 2: Conecte seu provedor terceirizado ao seu CodeCatalyst espaço](#)
- [Etapa 3: vincular seus recursos de terceiros ao seu CodeCatalyst projeto](#)
- [Próximas etapas](#)

## Etapa 1: instalar uma extensão de terceiros do CodeCatalyst catálogo

A primeira etapa para usar recursos de terceiros CodeCatalyst é instalar a extensão de GitHub repositórios do catálogo. CodeCatalyst Para instalar a extensão, execute as etapas a seguir, escolhendo a extensão para os recursos de terceiros que você deseja usar. GitHub repositórios, repositórios do Bitbucket e repositórios permitem que você use GitLab repositórios, repositórios do Bitbucket ou GitHub repositórios de projetos em. GitLab CodeCatalyst O Jira Software permite que você gerencie problemas do Jira no. CodeCatalyst

Para instalar uma extensão do CodeCatalyst catálogo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Navegue até o CodeCatalyst CodeCatalyst catálogo escolhendo o ícone Catálogo



no menu superior. Você pode pesquisar GitHub repositórios, repositórios Bitbucket, repositórios ou o GitLab Jira Software. Também é possível filtrar extensões com base em categorias.

4. (Opcional) Para ver mais detalhes sobre a extensão, como as permissões que ela terá, escolha o nome da extensão.
5. Escolha Instalar. Revise as permissões exigidas pela extensão e, se quiser continuar, escolha Instalar novamente.

Depois de instalar a extensão, você será direcionado para a página de detalhes da extensão. Dependendo da extensão instalada, você pode visualizar e gerenciar provedores conectados e recursos vinculados.

## Etapa 2: Conecte seu provedor terceirizado ao seu CodeCatalyst espaço

Depois de instalar os GitHub repositórios, os repositórios do Bitbucket, GitLab os repositórios ou a extensão do Jira Software, a próxima etapa é conectar sua GitHub conta, espaço de trabalho do Bitbucket, repositório do GitLab projeto ou site do Jira ao seu espaço. CodeCatalyst

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket ou site do Jira ao CodeCatalyst

- Faça o seguinte, dependendo da extensão de terceiros que você instalou:
  - GitHub repositórios: conecte-se a uma GitHub conta.
    1. Na guia GitHub Contas conectadas, escolha Conectar GitHub conta para acessar o site externo GitHub.
    2. Faça login na sua GitHub conta usando suas GitHub credenciais e, em seguida, escolha a conta na qual você deseja instalar a Amazon CodeCatalyst.

### Tip

Se você já conectou uma GitHub conta ao espaço, não será solicitado que você reautorize. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de uma GitHub organização, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas uma GitHub organização. Configure a aplicação para o acesso ao repositório que você deseja permitir e selecione Salvar. Se o botão Salvar não estiver ativo, faça uma alteração na configuração e tente novamente.

3. Escolha se você deseja permitir o acesso CodeCatalyst a todos os repositórios atuais e futuros ou escolha os GitHub repositórios específicos nos quais deseja usar. CodeCatalyst A opção padrão é incluir todos os GitHub repositórios na GitHub conta, incluindo repositórios futuros que serão acessados pelo. CodeCatalyst
4. Revise as permissões concedidas e CodeCatalyst, em seguida, escolha Instalar.

Depois de conectar sua GitHub conta CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitHub repositórios, onde poderá visualizar e gerenciar GitHub contas conectadas e GitHub repositórios vinculados.

- Repositórios do Bitbucket: conecte a um espaço de trabalho do Bitbucket.
  1. Na guia Espaços de trabalho do Bitbucket conectados, selecione Conectar espaço de trabalho do Bitbucket para acessar o site externo do Bitbucket.
  2. Faça login no seu espaço de trabalho do Bitbucket usando suas credenciais do Bitbucket e revise as permissões concedidas a. CodeCatalyst
  3. No menu suspenso Autorizar para espaço de trabalho, escolha o espaço de trabalho do Bitbucket ao qual você deseja fornecer CodeCatalyst acesso e, em seguida, escolha Conceder acesso.

 Tip

Se você já conectou um espaço de trabalho do Bitbucket ao espaço, não poderá dar uma nova autorização. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de um espaço de trabalho do Bitbucket, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas um espaço de trabalho do Bitbucket. Configure a aplicação para a qual você deseja permitir o acesso ao espaço de trabalho e selecione Conceder acesso. Se o botão Conceder acesso não estiver ativo, faça uma alteração na configuração e tente novamente.

Depois de conectar seu espaço de trabalho do Bitbucket ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de repositórios do Bitbucket, onde poderá visualizar e gerenciar espaços de trabalho do Bitbucket conectados e repositórios vinculados do Bitbucket.

- GitLab repositórios: conecte-se a um GitLab usuário.
  1. Escolha Connect GitLab user para acessar o site externo GitLab.
  2. Faça login com seu GitLab usuário usando suas GitLab credenciais e revise as permissões concedidas a. CodeCatalyst

 Tip

Se você já conectou um GitLab usuário ao espaço, não será solicitado que você reautorize. Em vez disso, você será direcionado de volta ao CodeCatalyst console.

### 3. Escolha Autorizar AWS conector para GitLab.

Depois de conectar seu GitLab usuário ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitLab repositórios, onde poderá visualizar e gerenciar repositórios de GitLab usuários e GitLab projetos vinculados.

- Software Jira: conecte um site do Jira.
  1. Na guia Sites do Jira conectados, selecione Conectar site do Jira para acessar o site externo do Atlassian Marketplace.
  2. Escolha Adquira agora para começar a instalar CodeCatalyst no seu site do Jira.

 Note

Se você instalou anteriormente CodeCatalyst em seu site do Jira, você será notificado. Selecione Começar para ir para a etapa final.

### 3. Dependendo do perfil, faça o seguinte:

1. Se você for administrador do site do Jira, no menu suspenso do site, escolha o site do Jira para instalar o CodeCatalyst aplicativo e, em seguida, escolha Instalar aplicativo.

 Note

Se você tiver um site do Jira, essa etapa não aparecerá e você será automaticamente direcionado para a próxima etapa.

2. a. Se você não for administrador do Jira, no menu suspenso do site, escolha o site do Jira para instalar o CodeCatalyst aplicativo e, em seguida, escolha Solicitar aplicativo. Para ter mais informações sobre como instalar aplicações do Jira, consulte [Quem pode instalar aplicações?](#).

- b. Insira o motivo pelo qual você precisa instalar CodeCatalyst no campo de texto de entrada ou mantenha o texto padrão e escolha Enviar solicitação.
4. Revise as ações executadas CodeCatalyst quando o aplicativo é instalado e escolha Obter agora.
5. Depois que o aplicativo for instalado, escolha Retornar CodeCatalyst para para retornar CodeCatalyst a.

Depois de conectar seu site do Jira ao CodeCatalyst, você pode ver o site conectado na guia Sites conectados do Jira da página de detalhes da extensão do Jira Software.

## Etapa 3: vincular seus recursos de terceiros ao seu CodeCatalyst projeto

A terceira e última etapa para usar seus GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos ou gerenciar problemas do Jira CodeCatalyst é vinculá-los ao CodeCatalyst projeto no qual você deseja usá-los.

Para vincular um GitHub repositório, repositório Bitbucket, repositório de projetos ou GitLab projeto do Jira a um CodeCatalyst projeto na página de detalhes da extensão

- Faça o seguinte, dependendo da extensão de terceiros que você instalou e do provedor conectado:
    - GitHub repositórios: vincule um GitHub repositório.
      1. Na guia GitHub Repositórios vinculados, escolha Vincular GitHub repositório.
      2. No menu suspenso da GitHub conta, escolha a GitHub conta que contém o repositório que você deseja vincular.
      3. Na lista suspensa do GitHub repositório, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst
-  **Tip**

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.
4. (Opcional) Se você não vê um GitHub repositório na lista de repositórios, ele pode não ter sido configurado para acesso ao repositório no aplicativo da Amazon CodeCatalyst em.

GitHub Você pode configurar quais GitHub repositórios podem ser usados CodeCatalyst na conta conectada.

- a. Navegue até sua [GitHub](#) conta, escolha Configurações e, em seguida, escolha Aplicativos.
- b. Na guia GitHub Aplicativos instalados, escolha Configurar para o CodeCatalyst aplicativo Amazon.
- c. Siga um destes procedimentos para configurar o acesso aos GitHub repositórios aos quais você deseja vincular: CodeCatalyst
  - Para fornecer acesso a todos os repositórios atuais e futuros, selecione Todos os repositórios.
  - Para fornecer acesso a repositórios específicos, escolha Somente selecionar repositórios, escolha a lista suspensa Selecionar repositórios e, em seguida, escolha um repositório ao qual você deseja permitir a vinculação. CodeCatalyst
5. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o GitHub repositório.
6. Escolha Vincular.

Se você não quiser mais usar um GitHub repositório no CodeCatalyst, você pode desvinculá-lo de um CodeCatalyst projeto. Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- Repositórios do Bitbucket: vincule um repositório do Bitbucket.
  1. Na guia Repositórios do Bitbucket vinculados, selecione Vincular repositório do Bitbucket.
  2. No menu suspenso Espaço de trabalho do Bitbucket, selecione o espaço de trabalho do Bitbucket que contém o repositório que você deseja vincular.
  3. No menu suspenso do repositório Bitbucket, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.

4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o repositório do Bitbucket.
5. Escolha Vincular.

Se você não quiser mais usar um repositório do Bitbucket no CodeCatalyst, você pode desvinculá-lo de um projeto. CodeCatalyst Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- GitLab repositórios: vincule um repositório GitLab do projeto.
  1. Na guia Repositórios de GitLab projetos vinculados, escolha Vincular repositório GitLab do projeto.
  2. No menu suspenso GitLab do usuário, escolha o GitLab usuário que contém o repositório que você deseja vincular.
  3. Na lista suspensa do repositório do GitLab projeto, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.

4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o repositório do GitLab projeto.
5. Escolha Vincular.

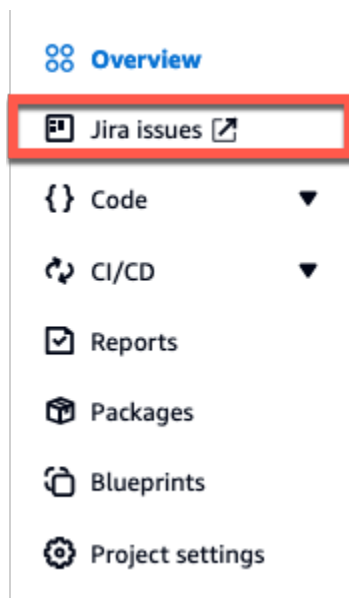
Se você não quiser mais usar um repositório de GitLab projetos no CodeCatalyst, você pode desvinculá-lo de um CodeCatalyst projeto. Quando um repositório do projeto é desvinculado,



os eventos desse repositório do projeto não iniciarão as execuções do fluxo de trabalho e você não poderá usar esse repositório do projeto com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- Software Jira: vincule um projeto do Jira.
  1. Na guia Projetos do Jira vinculados, selecione Vincular projeto do Jira.
  2. No menu suspenso Site do Jira, escolha o site do Jira que contém o projeto que você deseja vincular.
  3. No menu suspenso do projeto do Jira, escolha o projeto que você deseja vincular a um projeto. CodeCatalyst
  4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto que você deseja vincular a um projeto do Jira.
  5. Escolha Vincular.

Depois que um projeto do Jira é vinculado a um CodeCatalyst projeto, o acesso aos CodeCatalyst problemas é totalmente desativado e os problemas no painel de CodeCatalyst navegação serão substituídos por um item de problemas do Jira vinculado ao projeto Jira.



Se você não quiser mais usar um projeto do Jira no CodeCatalyst, você pode desvinculá-lo do seu CodeCatalyst projeto. Quando um projeto do Jira é desvinculado, os problemas do Jira não estarão disponíveis no CodeCatalyst projeto e os CodeCatalyst problemas voltarão a ser o fornecedor do problema. Para obter mais informações, consulte [Desvincular](#)

[repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst.](#)

Você também pode vincular seu GitHub repositório, repositório Bitbucket ou repositório de projetos a um GitLab projeto a partir dos repositórios Source no Code. Para obter mais informações, consulte [Vínculo de recursos de provedores de terceiros conectados.](#)

## Próximas etapas

Depois de instalar os GitHub repositórios, os repositórios do Bitbucket ou a extensão de GitLab repositórios, conectar seu provedor de recursos e vincular seus repositórios de terceiros aos seus CodeCatalyst projetos, você pode usá-lo em fluxos de trabalho e ambientes de desenvolvimento. CodeCatalyst Você também pode criar repositórios de terceiros na GitHub conta conectada, no espaço de trabalho do Bitbucket ou no GitLab usuário com código gerado a partir de um blueprint. Para obter mais informações, consulte [Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros](#) e [Criar um Ambiente de Desenvolvimento](#).

Depois de instalar a extensão do Jira Software, conectar seu site do Jira, vincular seus projetos do Jira ao seu CodeCatalyst projeto e vincular uma pull request, as atualizações do Jira CodeCatalyst são refletidas no seu projeto do Jira. Para ter mais informações sobre como vincular solicitações pull a problemas do Jira, consulte [Vinculando problemas do Jira a CodeCatalyst pull requests](#). Para obter mais informações sobre a visualização de CodeCatalyst eventos no Jira, consulte [Visualizando CodeCatalyst eventos em edições do Jira](#).

## Instalação de uma extensão em um espaço

Você pode instalar extensões para o seu CodeCatalyst espaço que adicionam funcionalidade aos projetos nesse espaço. Você pode visualizar o CodeCatalyst catálogo escolhendo o ícone

Catálogo 

Para saber mais sobre as extensões e suas funcionalidades, consulte [Extensões de terceiros disponíveis](#).

### Important

Para instalar uma extensão, você deve estar conectado com uma conta que tenha o perfil de Administrador do espaço no espaço.

**⚠ Important**

Depois de instalar uma extensão de repositório, todos os repositórios vinculados CodeCatalyst terão seu código indexado e armazenado. CodeCatalyst Isso tornará o código pesquisável em CodeCatalyst. Para entender melhor a proteção de dados do seu código ao usar repositórios vinculados CodeCatalyst, consulte [Proteção de dados](#) no Guia do CodeCatalyst usuário da Amazon.

Para instalar uma extensão do CodeCatalyst catálogo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Navegue até o CodeCatalyst catálogo escolhendo o ícone Catálogo



no menu superior. Você pode pesquisar GitHub repositórios, repositórios Bitbucket, repositórios ou o GitLab Jira Software. Também é possível filtrar extensões com base em categorias.

4. (Opcional) Escolha o nome da extensão para ver mais detalhes sobre a extensão, como as permissões que ela terá.
5. Escolha Instalar. Revise as permissões exigidas pela extensão e, se quiser continuar, escolha Instalar novamente.

Depois de instalar uma extensão, você verá a página de detalhes da extensão instalada. Percorra as guias para ter mais informações sobre a extensão. Na página de detalhes, você também executará configurações adicionais da extensão, se necessário.

## Desinstalação de uma extensão em um espaço

Você pode desinstalar extensões que foram instaladas anteriormente em seu CodeCatalyst espaço. A desinstalação de uma extensão pode remover recursos relacionados a essa extensão do seu CodeCatalyst espaço ou projetos.

 Important

Para desinstalar uma extensão, você deve estar conectado com uma conta que tenha o perfil de administrador do espaço no espaço.

Para desinstalar uma extensão do seu CodeCatalyst espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Para ver uma lista das extensões instaladas em seu espaço, faça o seguinte:
  - a. Escolha Configurações e Extensões instaladas.
  - b. Escolha o ícone Catálogo  no menu superior.
4. Selecione Configurar na extensão que deseja desinstalar.
5. Selecione Desinstalar na página de detalhes da extensão.
6. Revise as informações na caixa de diálogo Desinstalar extensão. Siga as instruções e selecione Desinstalar para desinstalar a extensão.

## Conectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst

Para usar um GitHub repositório, repositório Bitbucket ou repositório de GitLab projetos ou gerenciar um projeto do Jira no CodeCatalyst, você deve primeiro conectar sua fonte terceirizada ao seu espaço. CodeCatalyst Para saber mais sobre as extensões e suas funcionalidades, consulte [Extensões de terceiros disponíveis](#).

 Important

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket, GitLab usuário ou site do Jira ao seu CodeCatalyst espaço, você deve ser o administrador da fonte terceirizada e o administrador do Space. CodeCatalyst

**Note**

Se você estiver usando uma conexão com uma GitHub conta, deverá criar uma conexão pessoal para estabelecer o mapeamento de identidade entre sua CodeCatalyst identidade e sua GitHub identidade. Para obter mais informações, consulte [Conexões pessoais](#) e [Acessando GitHub recursos com conexões pessoais](#).

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket, GitLab usuário ou site do Jira ao CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Para ver uma lista das extensões instaladas em seu espaço, faça o seguinte:
  - a. Escolha Configurações e Extensões instaladas.
  - b. Escolha o ícone Catálogo  no menu superior.
4. Escolha Configurar para uma das seguintes extensões que você deseja configurar: GitHub repositórios, repositórios Bitbucket, repositórios ou GitLab Jira Software.
5. Faça o seguinte, dependendo da extensão de terceiros que você decidiu configurar:
  - GitHub repositórios: conecte-se a uma GitHub conta.
    1. Na guia GitHub Contas conectadas, escolha Conectar GitHub conta para acessar o site externo GitHub.
    2. Faça login na sua GitHub conta usando suas GitHub credenciais e, em seguida, escolha a conta na qual você deseja instalar a Amazon CodeCatalyst.

**Tip**

Se você já conectou uma GitHub conta ao espaço, não será solicitado que você reautorize. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de um GitHub espaço, ou a página de configuração do CodeCatalyst aplicativo

Amazon se você pertencer a apenas um GitHub espaço. Configure a aplicação para o acesso ao repositório que você deseja permitir e selecione Salvar. Se o botão Salvar não estiver ativo, faça uma alteração na configuração e tente novamente.

3. Escolha se você deseja permitir o acesso CodeCatalyst a todos os repositórios atuais e futuros ou escolha os GitHub repositórios específicos nos quais deseja usar. CodeCatalyst A opção padrão é incluir todos os GitHub repositórios na GitHub conta, incluindo repositórios futuros que serão acessados pelo. CodeCatalyst
4. Revise as permissões concedidas e CodeCatalyst, em seguida, escolha Instalar.

Depois de conectar sua GitHub conta CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitHub repositórios, onde poderá visualizar e gerenciar GitHub contas conectadas e GitHub repositórios vinculados.

- Repositórios do Bitbucket: conecte a um espaço de trabalho do Bitbucket.
  1. Na guia Espaços de trabalho do Bitbucket conectados, selecione Conectar espaço de trabalho do Bitbucket para acessar o site externo do Bitbucket.
  2. Faça login no seu espaço de trabalho do Bitbucket usando suas credenciais do Bitbucket e revise as permissões concedidas a. CodeCatalyst
  3. No menu suspenso Autorizar para espaço de trabalho, escolha o espaço de trabalho do Bitbucket ao qual você deseja fornecer CodeCatalyst acesso e, em seguida, escolha Conceder acesso.

 Tip

Se você já conectou um espaço de trabalho do Bitbucket ao espaço, não poderá dar uma nova autorização. Em vez disso, você verá uma caixa de diálogo perguntando onde você gostaria de instalar a extensão se você for membro ou colaborador em mais de um espaço de trabalho do Bitbucket, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas um espaço de trabalho do Bitbucket. Configure a aplicação para a qual você deseja permitir o acesso ao espaço de trabalho e selecione Conceder acesso. Se o botão Conceder acesso não estiver ativo, faça uma alteração na configuração e tente novamente.

Depois de conectar seu espaço de trabalho do Bitbucket ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de repositórios do Bitbucket, onde poderá visualizar e gerenciar espaços de trabalho do Bitbucket conectados e repositórios vinculados do Bitbucket.

- GitLab repositórios: conecte-se a um GitLab usuário.
  1. Escolha Connect GitLab user para acessar o site externo GitLab.
  2. Faça login com seu GitLab usuário usando suas GitLab credenciais e revise as permissões concedidas a. CodeCatalyst

 Tip

Se você já conectou um GitLab usuário ao espaço, não será solicitado que você reautorize. Em vez disso, você será direcionado de volta ao CodeCatalyst console.

3. Escolha Autorizar AWS conector para GitLab.

Depois de conectar seu GitLab usuário ao CodeCatalyst, você será direcionado para a página de detalhes da extensão de GitLab repositórios, onde poderá visualizar e gerenciar repositórios de GitLab usuários e GitLab projetos vinculados.

- Software Jira: conecte um site do Jira.
  1. Na guia Sites do Jira conectados, selecione Conectar site do Jira para acessar o site externo do Atlassian Marketplace.
  2. Escolha Adquirir agora para começar a instalar CodeCatalyst no seu site do Jira.

 Note

Se você instalou anteriormente CodeCatalyst em seu site do Jira, você será notificado. Selecione Começar para ir para a etapa final.

3. Dependendo do perfil, faça o seguinte:
  1. Se você for administrador do site do Jira, no menu suspenso do site, escolha o site do Jira para instalar o CodeCatalyst aplicativo e, em seguida, escolha Instalar aplicativo.

 Note

Se você tiver um site do Jira, essa etapa não aparecerá e você será automaticamente direcionado para a próxima etapa.

2. a. Se você não for administrador do Jira, no menu suspenso do site, escolha o site do Jira para instalar o CodeCatalyst aplicativo e, em seguida, escolha Solicitar aplicativo. Para ter mais informações sobre como instalar aplicações do Jira, consulte [Quem pode instalar aplicações?](#).
- b. Insira o motivo pelo qual você precisa instalar CodeCatalyst no campo de texto de entrada ou mantenha o texto padrão e escolha Enviar solicitação.
4. Revise as ações executadas CodeCatalyst quando o aplicativo é instalado e escolha Obter agora.
5. Depois que o aplicativo for instalado, escolha Retornar CodeCatalyst para para retornar CodeCatalyst a.

Depois de conectar seu site do Jira ao CodeCatalyst, você pode ver o site conectado na guia Sites conectados do Jira da página de detalhes da extensão do Jira Software.

Se você não quiser mais usar GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos, nem gerenciar problemas do Jira no CodeCatalyst, você pode desconectar sua fonte terceirizada. Quando uma GitHub conta, espaço de trabalho do Bitbucket ou GitLab usuário é desconectado, os eventos nos repositórios de terceiros não iniciarão a execução do fluxo de trabalho e você não poderá usar esses repositórios com ambientes de desenvolvimento. CodeCatalyst Quando um site do Jira é desconectado, os problemas do Jira dos projetos do site não estarão disponíveis nos CodeCatalyst projetos, e os CodeCatalyst problemas voltarão a ser o fornecedor do problema. Para obter mais informações, consulte [Desconectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst](#).

## Desconectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst

Se você não quiser mais usar GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos, nem gerenciar problemas do Jira no CodeCatalyst, você pode desconectar sua fonte



terceirizada. Depois que uma GitHub conta, espaço de trabalho do Bitbucket ou GitLab usuário for desconectado, os eventos nos repositórios não iniciarão a execução do fluxo de trabalho e você não poderá usar esses repositórios com ambientes de desenvolvimento. CodeCatalyst Quando um site do Jira é desconectado, os problemas do Jira dos projetos do site não estarão disponíveis nos CodeCatalyst projetos, e os CodeCatalyst problemas voltarão a ser o fornecedor do problema.

#### Note

- Para desconectar uma GitHub conta, primeiro você deve desvincular todos os GitHub repositórios vinculados dessa conta.
- Para desconectar um espaço de trabalho do Bitbucket, você deve primeiro desvincular todos os repositórios vinculados do Bitbucket desse espaço de trabalho.
- Para desconectar um GitLab usuário, primeiro você deve desvincular todos os repositórios de GitLab projetos vinculados desse espaço de trabalho.
- Para desconectar um site do Jira, você deve primeiro desvincular todos os projetos do Jira vinculados dessa conta.

Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

Para desconectar um GitHub projeto, espaço de trabalho do Bitbucket, GitLab usuário ou site do Jira

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Para ver uma lista das extensões instaladas em seu espaço, faça o seguinte:
  - a. Escolha Configurações e Extensões instaladas.
  - b. Escolha o ícone Catálogo  no menu superior.
4. Escolha Configurar para uma das seguintes extensões que você deseja configurar: GitHub repositórios, repositórios Bitbucket, repositórios ou GitLab Jira Software.
5. Faça o seguinte, dependendo da extensão de terceiros que você decidiu configurar:

- GitHub repositórios: desconecte-se de uma conta. GitHub

Na guia **GitHub Contas conectadas**, escolha a GitHub conta que você deseja desconectar e, em seguida, escolha **Desconectar GitHub** conta.

- Repositórios do Bitbucket: desconecte de um espaço de trabalho do Bitbucket.

Na guia **Espaços de trabalho do Bitbucket conectados**, escolha o espaço de trabalho do Bitbucket que você deseja desconectar e, em seguida, selecione **Desconectar espaço de trabalho do Bitbucket**.

- GitLab repositórios: desconecte-se de um usuário. GitLab

Na guia **GitLab Usuários conectados**, escolha o GitLab usuário que você deseja desconectar e, em seguida, escolha **Desconectar GitLab** usuário.

- Software Jira: desconectar de um site do Jira.

Na guia **Sites do Jira conectados**, escolha o site do Jira que você deseja desconectar e, em seguida, selecione **Desconectar site do Jira**.

6. Na caixa de diálogo **Desconectar**, analise os efeitos da desconexão da conta.
7. Digite **desconectar** no campo de entrada de texto e selecione **Desconectar**.

## Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst

Antes de usar um GitHub repositório, repositório Bitbucket ou repositório de projetos, ou gerenciar um GitLab projeto do Jira, você deve conectar a fonte terceirizada à qual o repositório ou projeto pertence ao seu espaço. CodeCatalyst Para obter mais informações, consulte [Conectando GitHub contas, espaços de trabalho do Bitbucket, GitLab usuários e sites do Jira CodeCatalyst](#).

Você pode usar repositórios vinculados, GitHub repositórios do Bitbucket ou repositórios de GitLab projetos em fluxos de trabalho, onde os eventos nos repositórios vinculados iniciam fluxos de trabalho que podem criar, testar ou implantar código, dependendo da configuração do fluxo de trabalho. Os arquivos de configuração do fluxo de trabalho para fluxos de trabalho que usam repositórios vinculados GitHub ou do Bitbucket são armazenados nos repositórios vinculados. Os repositórios vinculados também podem ser usados com ambientes de desenvolvimento para criar, atualizar e excluir arquivos nos repositórios vinculados. Você pode vincular um GitHub repositório, repositório do Bitbucket ou repositório do GitLab projeto a um CodeCatalyst projeto na página de

detalhes dos repositórios, dos GitHub repositórios do Bitbucket ou da extensão dos repositórios, ou na visualização de GitLab repositórios de origem em Código no próprio projeto.

 Important

Embora você possa vincular um repositório GitHub ou do Bitbucket como colaborador, você só pode desvincular um repositório de terceiros como administrador do Space ou administrador do projeto. Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

 Important

Depois de instalar uma extensão de repositório, todos os repositórios aos quais você vincular CodeCatalyst terão seu código indexado e armazenado. CodeCatalyst Isso tornará o código pesquisável em CodeCatalyst. Para entender melhor a proteção de dados do seu código ao usar repositórios vinculados CodeCatalyst, consulte [Proteção de dados](#) no Guia do CodeCatalyst usuário da Amazon.

 Important

CodeCatalyst não suporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, você deve primeiro desvinculá-la CodeCatalyst, alterar a ramificação padrão e depois vinculá-la novamente.

Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

Você pode usar projetos vinculados do Jira para gerenciar problemas e vincular CodeCatalyst pull requests a um problema do Jira. O status resumido de uma pull request e o status dos eventos de CodeCatalyst fluxo de trabalho associados são refletidos em seu problema com o Jira.

 Important

Para vincular seu projeto do Jira ao seu CodeCatalyst projeto, você deve ser administrador do CodeCatalyst Space ou administrador CodeCatalyst do projeto.

 Note

- Um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados, GitHub repositórios Bitbucket ou repositórios de projetos com GitLab projetos. CodeCatalyst
- Você não pode vincular um GitHub repositório, repositório Bitbucket ou GitLab repositório que tenha o mesmo nome de um repositório em um projeto. CodeCatalyst
- A extensão de GitHub repositórios não é compatível com os repositórios do GitHub Enterprise Server.
- A extensão de Repositórios do Bitbucket não é compatível com os repositórios do Bitbucket Data Center.
- A extensão GitLab repositories não é compatível com repositórios de projetos GitLab autogerenciados.
- Não é possível usar os recursos Escrever descrição para mim ou Resumir comentários com repositórios vinculados. Esses recursos só estão disponíveis em pull requests em CodeCatalyst.
- Um CodeCatalyst projeto só pode ser vinculado a um projeto do Jira. Um projeto do Jira pode ser vinculado a vários CodeCatalyst projetos.

## Tópicos

- [Vínculo de recursos de provedores de terceiros conectados](#)
- [Vinculando um repositório de terceiros durante CodeCatalyst a criação do projeto](#)

## Vínculo de recursos de provedores de terceiros conectados

Para vincular um GitHub repositório, repositório Bitbucket, repositório de projetos ou GitLab projeto do Jira a um CodeCatalyst projeto na página de detalhes da extensão

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Para ver uma lista das extensões instaladas em seu espaço, faça o seguinte:

- a. Escolha Configurações e Extensões instaladas.
- b. Escolha o ícone Catálogo



no menu superior.

4. Escolha Configurar para uma das seguintes extensões: GitHub repositórios, repositórios Bitbucket, repositórios ou GitLab Jira Software.
5. Faça o seguinte, dependendo da extensão de terceiros que você decidiu configurar:
  - GitHub repositórios: vincule um GitHub repositório.
    1. Na guia GitHub Repositórios vinculados, escolha Vincular GitHub repositório.
    2. No menu suspenso da GitHub conta, escolha a GitHub conta que contém o repositório que você deseja vincular.
    3. Na lista suspensa do GitHub repositório, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst

### Tip

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.

4. (Opcional) Se você não vê um GitHub repositório na lista de repositórios, ele pode não ter sido configurado para acesso ao repositório no aplicativo da Amazon CodeCatalyst em GitHub. Você pode configurar quais GitHub repositórios podem ser usados CodeCatalyst na conta conectada.
  - a. Navegue até sua [GitHub](#) conta, escolha Configurações e, em seguida, escolha Aplicativos.

- b. Na guia GitHub Aplicativos instalados, escolha Configurar para o CodeCatalyst aplicativo Amazon.
- c. Siga um destes procedimentos para configurar o acesso aos GitHub repositórios aos quais você deseja vincular: CodeCatalyst
  - Para fornecer acesso a todos os repositórios atuais e futuros, selecione Todos os repositórios.
  - Para fornecer acesso a repositórios específicos, escolha Somente selecionar repositórios, escolha a lista suspensa Selecionar repositórios e, em seguida, escolha um repositório ao qual você deseja permitir a vinculação. CodeCatalyst
5. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o GitHub repositório.
6. Escolha Vincular.

Se você não quiser mais usar um GitHub repositório no CodeCatalyst, você pode desvinculá-lo de um CodeCatalyst projeto. Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- Repositórios do Bitbucket: vincule um repositório do Bitbucket.
  1. Na guia Repositórios do Bitbucket vinculados, selecione Vincular repositório do Bitbucket.
  2. No menu suspenso Espaço de trabalho do Bitbucket, selecione o espaço de trabalho do Bitbucket que contém o repositório que você deseja vincular.
  3. No menu suspenso do repositório Bitbucket, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.

4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o repositório do Bitbucket.
5. Escolha Vincular.

Se você não quiser mais usar um repositório do Bitbucket no CodeCatalyst, você pode desvinculá-lo de um projeto. CodeCatalyst Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- GitLab repositórios: vincule um repositório GitLab do projeto.
  1. Na guia Repositórios de GitLab projetos vinculados, escolha Vincular repositório GitLab do projeto.
  2. No menu suspenso do GitLab usuário, escolha o GitLab usuário que contém o repositório do projeto que você deseja vincular.
  3. Na lista suspensa do repositório do GitLab projeto, escolha o repositório que você deseja vincular a um projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver desativado, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto no espaço.

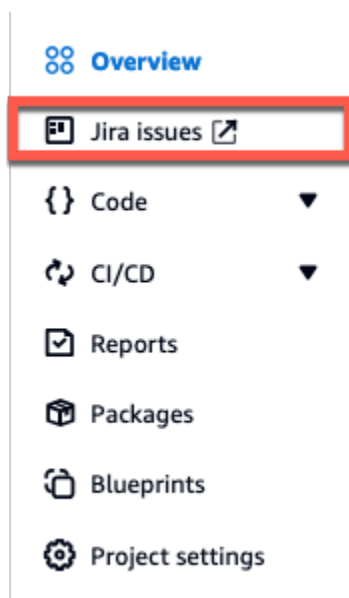
4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto ao qual você deseja vincular o repositório do GitLab projeto.
5. Escolha Vincular.

Se você não quiser mais usar um repositório de GitLab projetos no CodeCatalyst, você pode desvinculá-lo de um CodeCatalyst projeto. Quando um repositório do projeto é desvinculado, os eventos desse repositório do projeto não iniciarão as execuções do fluxo de trabalho e você não poderá usar esse repositório do projeto com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

- Software Jira: vincule um projeto do Jira.
  1. Na guia Projetos do Jira vinculados, selecione Vincular projeto do Jira.
  2. No menu suspenso Site do Jira, escolha o site do Jira que contém o projeto que você deseja vincular.

3. No menu suspenso do projeto do Jira, escolha o projeto que você deseja vincular a um projeto. CodeCatalyst
4. No menu suspenso do CodeCatalyst projeto, escolha o CodeCatalyst projeto que você deseja vincular a um projeto do Jira.
5. Escolha Vincular.

Depois que um projeto do Jira é vinculado a um CodeCatalyst projeto, o acesso aos CodeCatalyst problemas é totalmente desativado e os problemas no painel de CodeCatalyst navegação serão substituídos por um item de problemas do Jira vinculado ao projeto Jira.



Se você não quiser mais usar um projeto do Jira no CodeCatalyst, você pode desvinculá-lo do seu CodeCatalyst projeto. Quando um projeto do Jira é desvinculado, os problemas do Jira não estarão disponíveis no CodeCatalyst projeto e os CodeCatalyst problemas voltarão a ser o fornecedor do problema. Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

Para vincular um GitHub repositório, repositório Bitbucket ou repositório de projeto a um GitLab projeto a partir da página de repositórios de origem em um CodeCatalyst projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.



4. Selecione Adicionar repositório e Vincular repositório.
5. No menu suspenso Provedor do repositório, escolha um dos seguintes provedores de repositório terceirizados: GitHub, Bitbucket, GitLab
6. Realize um destes procedimentos, dependendo do provedor de repositório de terceiros que você selecionar para vincular:

- GitHub repositórios: vincule um GitHub repositório.
  1. No menu suspenso da GitHub conta, escolha a GitHub conta que contém o repositório que você deseja vincular.
  2. No menu suspenso do GitHub repositório, escolha o GitHub repositório ao qual você deseja vincular seu projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver em cinza, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto na Amazon. CodeCatalyst

3. (Opcional) Se você não vê um GitHub repositório na lista de repositórios, ele pode não ter sido configurado para acesso ao repositório no aplicativo da Amazon CodeCatalyst em. GitHub Você pode configurar quais GitHub repositórios podem ser usados CodeCatalyst na conta conectada.
  - a. Navegue até sua [GitHub](#) conta, escolha Configurações e, em seguida, escolha Aplicativos.
  - b. Na guia GitHub Aplicativos instalados, escolha Configurar para o CodeCatalyst aplicativo Amazon.
  - c. Siga um destes procedimentos para configurar o acesso aos GitHub repositórios aos quais você deseja vincular: CodeCatalyst
    - Para fornecer acesso a todos os repositórios atuais e futuros, selecione Todos os repositórios.
    - Para fornecer acesso a repositórios específicos, escolha Somente selecionar repositórios, escolha a lista suspensa Selecionar repositórios e, em seguida, escolha um repositório ao qual você deseja permitir a vinculação. CodeCatalyst
- Repositórios do Bitbucket: vincule um repositório do Bitbucket.

1. No menu suspenso Espaço de trabalho do Bitbucket, selecione o espaço de trabalho do Bitbucket que contém o repositório que você deseja vincular.
2. No menu suspenso do repositório do Bitbucket, escolha o repositório do Bitbucket ao qual você deseja vincular seu projeto. CodeCatalyst

 Tip

Se o nome do repositório estiver em cinza, você não poderá vincular esse repositório porque ele já foi vinculado a outro projeto na Amazon. CodeCatalyst

- GitLab repositórios: vincule um repositório GitLab do projeto.
  1. No menu suspenso GitLab do usuário, escolha o GitLab usuário que contém o repositório do projeto que você deseja vincular.
  2. No menu suspenso do repositório do GitLab projeto, escolha o repositório do GitLab projeto ao qual você deseja vincular seu projeto. CodeCatalyst

 Tip

Se o nome do repositório do projeto estiver em cinza, você não poderá vincular esse repositório do projeto porque ele já foi vinculado a outro projeto na Amazon. CodeCatalyst

## 7. Escolha Vincular.

Se você não quiser mais usar um GitHub repositório, um repositório Bitbucket ou um repositório de GitLab projetos no CodeCatalyst, você pode desvinculá-lo de um projeto. CodeCatalyst Quando um repositório é desvinculado, os eventos desse repositório não iniciarão a execução do fluxo de trabalho e você não poderá usar esse repositório com ambientes de desenvolvimento. CodeCatalyst Para obter mais informações, consulte [Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst](#).

Depois de vincular seu GitHub repositório, repositório Bitbucket ou repositório de projetos ao seu GitLab CodeCatalyst projeto, você pode usá-lo em CodeCatalyst fluxos de trabalho e ambientes de desenvolvimento. Você também pode usar os repositórios vinculados com o Amazon Q Developer,

esquemas e muito mais. Para obter mais informações, consulte [Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros](#) e [Criar um Ambiente de Desenvolvimento](#).

Depois de vincular seu projeto do Jira ao seu CodeCatalyst projeto e vincular uma pull request, as atualizações do Jira CodeCatalyst são refletidas no seu projeto do Jira. Para ter mais informações sobre como vincular solicitações pull a problemas do Jira, consulte [Vinculando problemas do Jira a CodeCatalyst pull requests](#). Para obter mais informações sobre a visualização de CodeCatalyst eventos no Jira, consulte [Visualizando CodeCatalyst eventos em edições do Jira](#).

## Vinculando um repositório de terceiros durante CodeCatalyst a criação do projeto

Você pode vincular um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto a um novo projeto ao criar o novo CodeCatalyst projeto. CodeCatalyst Para obter mais informações, consulte [Criar um projeto com um repositório de terceiros vinculado](#).

## Desvincular repositórios, GitHub repositórios do Bitbucket, repositórios de projetos e GitLab projetos do Jira no CodeCatalyst

Se você não quiser mais usar um GitHub repositório, repositório Bitbucket ou repositório de projetos, nem gerenciar um GitLab projeto do Jira no CodeCatalyst, você pode desvincular o repositório ou o projeto do seu projeto. CodeCatalyst

Desvincular um GitHub repositório, repositório do Bitbucket ou repositório GitLab do projeto não exclui o repositório nem faz alterações nele. Isso não exclui nenhum arquivo de configuração do fluxo de trabalho armazenado nesse repositório vinculado. No entanto, depois de desvincular um GitHub repositório, repositório do Bitbucket ou repositório GitLab do projeto, os eventos desse repositório não iniciarão mais a execução do fluxo de trabalho e você não poderá usar o repositório com ambientes de desenvolvimento. Você pode desvincular um GitHub repositório, repositório do Bitbucket ou repositório do GitLab projeto de um CodeCatalyst projeto na página de detalhes dos repositórios, dos GitHub repositórios do Bitbucket ou da extensão dos repositórios, ou na visualização de GitLab repositórios de origem no Código no próprio projeto.

Desvincular um projeto do Jira não exclui o projeto, incluindo itens de planejamento ou informações de desenvolvimento, nem faz alterações nele. No entanto, depois de desvincular um projeto do Jira, os problemas do Jira do projeto não estarão mais disponíveis para vinculação ao CodeCatalyst projeto, e CodeCatalyst Issues voltará a ser o provedor do problema.

**⚠ Important**

Para desvincular seu GitHub repositório, repositório Bitbucket ou repositório de projetos do GitLab do seu projeto, você deve ser o administrador do Space ou o administrador do CodeCatalyst projeto.

Para desvincular um GitHub repositório, repositório Bitbucket, repositório de GitLab projeto ou projeto Jira em um CodeCatalyst projeto da página de detalhes da extensão

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Para ver uma lista das extensões instaladas em seu espaço, faça o seguinte:
  - a. Escolha Configurações e Extensões instaladas.
  - b. Escolha o ícone Catálogo  no menu superior.
4. Escolha Configurar para uma das seguintes extensões que você deseja configurar: GitHub repositórios, repositórios Bitbucket, repositórios ou GitLab Jira Software.
5. Faça o seguinte, dependendo da extensão de terceiros que você decidiu configurar:
  - GitHub repositórios: desvincule um GitHub repositório.

Na guia GitHub repositórios, escolha o GitHub repositório que você deseja desvincular e, em seguida, escolha Desvincular repositório. GitHub
  - Repositórios do Bitbucket: desvincule um repositório do Bitbucket.

Na guia Repositórios do Bitbucket, escolha o repositório do Bitbucket que você deseja desvincular e, depois, selecione Desvincular repositório do Bitbucket.
  - GitLab repositórios: desvincule um repositório de GitLab projeto.

Na guia Repositórios do GitLab projeto, escolha o repositório do GitLab projeto que você deseja desvincular e, em seguida, escolha GitLab Desvincular repositório do projeto.
  - Software Jira: desvincule um projeto do Jira.

Na guia Projetos do Jira, escolha o projeto do Jira que você deseja desvincular e, depois, selecione Desvincular projeto do Jira.

6. Na caixa de diálogo Desvincular, revise os efeitos da desvinculação do repositório.
7. Insira desvincular no campo de entrada de texto e selecione Desvincular.

Para desvincular um GitHub repositório, repositório Bitbucket ou repositório de projeto em um GitLab CodeCatalyst projeto da página de repositórios de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.
4. Escolha o botão de opção do repositório que você deseja desvincular e, depois, selecione Desvincular repositório.
5. Revise as informações na caixa de diálogo. Siga as instruções e selecione Desvincular para desvincular o repositório.

## Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst

Depois de vincular GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos, você pode visualizá-los CodeCatalyst para confirmar e configurar os recursos. Você também pode pesquisar problemas vinculados do Jira em CodeCatalyst.

### Tópicos

- [Visualizando repositórios de terceiros no CodeCatalyst](#)
- [Pesquisando problemas com o Jira no CodeCatalyst](#)

## Visualizando repositórios de terceiros no CodeCatalyst

Você pode visualizar os repositórios vinculados, GitHub os repositórios do Bitbucket ou os repositórios GitLab do projeto na lista de repositórios de origem do seu projeto ou na página de detalhes dos repositórios, GitHub repositórios do Bitbucket ou da extensão dos repositórios. GitLab Escolhê-los na lista de repositórios não os abre em CodeCatalyst. Em vez disso, eles abrem no

provedor de repositórios terceirizado, onde é possível visualizar e trabalhar no código no repositório vinculado.

Para visualizar repositórios vinculados, GitHub repositórios do Bitbucket ou GitLab repositórios de projetos no CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e, depois, selecione Repositórios de origem.

Para visualizar repositórios vinculados, GitHub repositórios do Bitbucket ou repositórios de GitLab projetos na página de detalhes da extensão

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço e escolha a guia Extensões instaladas.
3. Dependendo dos repositórios de terceiros que deseja visualizar, realize uma das seguintes ações:
  - Em GitHub repositórios, escolha Configurar e, em seguida, escolha GitHub Repositórios vinculados para visualizar todos os GitHub repositórios conectados aos CodeCatalyst projetos em seu espaço. CodeCatalyst
  - Nos repositórios do Bitbucket, escolha Configurar e, em seguida, escolha Repositórios vinculados do Bitbucket para visualizar todos os repositórios do Bitbucket conectados aos projetos em seu espaço. CodeCatalyst CodeCatalyst
  - Em GitLab repositórios, escolha Configurar e, em seguida, escolha Repositórios de GitLab projetos vinculados para visualizar todos os repositórios de GitLab projetos conectados aos CodeCatalyst projetos em seu espaço. CodeCatalyst

Os GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos vinculados ao seu CodeCatalyst projeto são mostrados na lista. Escolha o GitHub repositório, o repositório do Bitbucket ou o repositório GitLab do projeto para visualizar e editar arquivos no provedor de repositórios terceirizado.

 Note

Se um fluxo de trabalho usar um GitHub repositório, repositório Bitbucket ou repositório de GitLab projeto em uma ação de origem, as alterações feitas no fluxo de trabalho YAML no editor visual ou no editor YAML em CodeCatalyst serão automaticamente confirmadas e enviadas para o repositório de terceiros.

## Pesquisando problemas com o Jira no CodeCatalyst

Depois de vincular um projeto do Jira, você pode pesquisar problemas no projeto vinculado do Jira usando a barra de pesquisa CodeCatalyst global. Você também pode pesquisar problemas do Jira CodeCatalyst enquanto vincula os problemas a partir de uma pull request. Para obter mais informações sobre como vincular problemas do Jira a uma CodeCatalyst pull request, consulte.

[Vinculando problemas do Jira a CodeCatalyst pull requests](#)

Como procurar problemas do Jira em projetos vinculados do Jira

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. Na barra de pesquisa global, pesquise problemas em um projeto vinculado do Jira ou problemas do Jira que você deseja vincular a uma solicitação pull.

## Início automático da execução de um fluxo de trabalho após eventos de repositórios de terceiros

Você pode usar um GitHub repositório vinculado, um repositório do Bitbucket ou um repositório GitLab do projeto como fonte para um fluxo de trabalho, no qual as alterações em uma ramificação especificada em um GitHub repositório vinculado, repositório do Bitbucket ou repositório do GitLab projeto iniciam automaticamente a execução do fluxo de trabalho.

Fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar o código como parte de um sistema de integração contínua e entrega contínua (CI/CD). Um fluxo de trabalho define uma série de etapas ou ações a serem realizadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que fazem com que o fluxo de trabalho seja iniciado. Para configurar um fluxo de trabalho, você cria um arquivo de definição de fluxo de trabalho usando o [editor visual ou YAML](#) do CodeCatalyst console.

**Tip**

Para ver rapidamente como usar fluxos de trabalho em um projeto, [crie um projeto com um esquema](#). Cada esquema implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Quando você configura um fluxo de trabalho para usar um GitHub repositório vinculado, repositório do Bitbucket ou repositório GitLab do projeto, o arquivo de configuração do fluxo de trabalho é armazenado nesse GitHub repositório, repositório do Bitbucket ou repositório do projeto. GitLab A configuração do fluxo de trabalho é um arquivo YAML que define o nome, os gatilhos, os recursos, os artefatos e as ações do fluxo de trabalho. Para ter mais informações sobre o arquivo de configuração do fluxo de trabalho, consulte [Definição do YAML do fluxo de trabalho](#).

O arquivo de configuração do fluxo de trabalho deve estar no `./codecatalyst/workflows/` diretório do seu GitHub repositório, repositório Bitbucket ou GitLab repositório do projeto.

Você pode usar o editor de fluxo de trabalho para criar e configurar fluxos de trabalho. Para obter mais informações, consulte [Conceitos básicos de fluxos de trabalho](#) e [Conectar repositórios de origem aos fluxos de trabalho](#).

## Adição de gatilhos para iniciar a execução do fluxo de trabalho

Você pode configurar um CodeCatalyst fluxo de trabalho para iniciar automaticamente uma execução quando o código for enviado para a ramificação especificada do seu repositório GitHub ou do Bitbucket. Para iniciar a execução automática de um fluxo de trabalho, adicione um gatilho à seção `Triggers` do arquivo de configuração do fluxo de trabalho.

### Exemplo: um simples gatilho de envio de código

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que o código é enviado para qualquer ramificação no seu repositório de origem.

```
Triggers:
- Type: PUSH
```

### Exemplo: um simples gatilho de solicitação pull

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma solicitação pull é criada em qualquer ramificação no seu repositório de origem.



**Triggers:**

- Type: PULLREQUEST

**Events:**

- OPEN

Para obter mais informações, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

## Restrição do acesso ao IP com provedores de repositório de terceiros

Você pode restringir o acesso aos seus GitHub repositórios, repositórios do Bitbucket ou repositórios de GitLab projetos com base em endereços IP definindo regras ou configurações. Faça isso por meio das configurações ou dos recursos de controle de acesso do provedor de terceiros.

Dependendo do provedor de repositório de terceiros que você está usando, observe o seguinte:

- A extensão de CodeCatalyst GitHub repositórios da Amazon é compatível com as [restrições de acesso IP do GitHub Enterprise Cloud](#). Ao configurar uma organização do GitHub Enterprise Cloud para restringir o acesso a endereços IP específicos, você também pode [permitir que GitHub os aplicativos configurem a lista de permissões, que permitirá](#) CodeCatalyst registrar seus endereços IP automaticamente com GitHub. Como alternativa, você pode [adicionar manualmente os endereços CodeCatalyst IP](#).
- A extensão CodeCatalyst de repositórios Amazon Bitbucket é compatível com as restrições de acesso do [Bitbucket Cloud Premium](#). Ao configurar um espaço de trabalho do Bitbucket Cloud Premium para restringir o acesso a endereços IP específicos, você também pode [adicionar endereços IP ou blocos de rede para um conjunto de endereços IP a uma lista de permissões](#).
- A extensão de CodeCatalyst GitLab repositórios da Amazon é compatível com [restrições GitLab de endereço IP](#). Ao configurar um grupo GitLab Premium ou Ultimate para restringir o acesso a endereços IP específicos, você também pode [adicionar endereços IP ou blocos de rede para um conjunto de endereços IP a uma lista de permissões](#).

Se os endereços CodeCatalyst IP não estiverem na lista de permissões de um repositório de terceiros, o CodeCatalyst aplicativo da Amazon não poderá acessar seus repositórios de terceiros. Para obter mais informações, consulte [Endereços IP usados pela extensão de repositórios de terceiros](#).

## Endereços IP usados pela extensão de repositórios de terceiros

Os seguintes endereços IP são usados pelas extensões de terceiros para acessar seus recursos de terceiros:

- GitHub repositórios:

```
us-west-2
 52.32.242.246
 54.148.176.49
 35.164.118.94
eu-west-1
 34.241.64.10
 34.246.255.80
 3.248.38.7
```

- Repositórios e GitLab repositórios do Bitbucket:

```
us-west-2
 35.160.210.199
 54.71.206.108
 54.71.36.205
eu-west-1
 34.242.64.82
 52.18.37.201
 54.77.75.62
```

## Bloqueio de mesclagens de terceiros quando os fluxos de trabalho falham

Depois de vincular um repositório GitHub ou do Bitbucket a CodeCatalyst, você pode adicionar CodeCatalyst fluxos de trabalho para pull requests. Da mesma forma, depois de vincular um repositório de GitLab projetos, CodeCatalyst você pode adicionar CodeCatalyst fluxos de trabalho para solicitações de mesclagem. Uma ou mais execuções de fluxo de trabalho podem ocorrer em uma confirmação específica, e o status de execução de cada fluxo de trabalho em também CodeCatalyst é refletido como parte do status de confirmação em GitHub, Bitbucket ou GitLab. Quando uma nova confirmação é enviada, os novos [status de execução](#) do fluxo de trabalho são refletidos no GitHub Bitbucket ou GitLab para essa nova confirmação. Se você executar um fluxo

de trabalho novamente para uma confirmação, o novo status de execução do fluxo de trabalho substituirá o status anterior dessa confirmação e do fluxo de trabalho.

Você pode definir regras de proteção de ramificações no Bitbucket GitHub ou no Bitbucket para bloquear uma mesclagem de pull request ou GitLab para bloquear uma solicitação de mesclagem quando a confirmação mais recente tiver um status de falha na execução do fluxo de trabalho. Com as regras de proteção de ramificações, o status da confirmação mais recente afeta a capacidade de mesclar uma pull request no GitHub Bitbucket ou. GitLab Para saber mais sobre fluxos de trabalho, consulte [Executar um fluxo de trabalho](#) e [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

Dependendo do provedor de repositório terceirizado que você está usando, veja o seguinte:

- GitHub repositórios: GitHub documentação [sobre verificações de status](#) e [Sobre filiais protegidas](#).
- Repositórios do Bitbucket: documentação do Bitbucket para [Uso de permissões de ramificação](#) e [Assumir o controle com permissões de ramificação no Bitbucket Cloud](#).
- GitLab repositórios: GitLab documentação para [mesclagem automática](#) e ramificações [protegidas](#).

## Vinculando problemas do Jira a CodeCatalyst pull requests

Você pode vincular pull requests criadas em um repositório de CodeCatalyst origem aos problemas do Jira. Depois de vincular um problema do Jira, o problema é exibido como uma propriedade da solicitação pull. Como resultado, eventos de solicitação pull, eventos de fluxo de trabalho e eventos de implantação são enviados para o Jira e adicionados ao problema do Jira. As solicitações pull podem ser vinculadas a um ou mais problemas do Jira. Você só pode vincular pull requests que estão em um repositório de CodeCatalyst origem, não aquelas em um repositório de terceiros, como. GitHub Antes de vincular problemas do Jira a uma pull request, seu projeto do Jira deve estar vinculado ao CodeCatalyst projeto. Para obter mais informações sobre como vincular um projeto do Jira a um CodeCatalyst projeto, consulte. [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#)

### Note

Você não pode criar uma pull request sem um repositório de origem com duas ramificações em seu CodeCatalyst projeto. Para obter mais informações sobre pull requests, consulte Como [trabalhar com pull requests em CodeCatalyst](#).

## Para vincular problemas do Jira a uma CodeCatalyst pull request

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e Solicitações pull.
4. Selecione Criar solicitação pull para inserir os detalhes da solicitação pull.
5. No menu suspenso Repositório de origem, escolha o repositório de origem ao qual você deseja vincular uma solicitação pull.
6. No menu suspenso Ramificação de origem, escolha a ramificação que contém as alterações que você deseja revisar.
7. No menu suspenso Ramificação de destino, escolha a ramificação na qual você deseja mesclar as alterações revisadas.
8. No campo de entrada de texto Título da solicitação pull, insira o título das suas solicitações pull.
9. Selecione Vincular problemas para o campo Problemas do Jira - opcional, escolha o menu suspenso e pesquise os problemas do Jira que você deseja adicionar do projeto vinculado do Jira.
10. Selecione os problemas do Jira que você deseja adicionar à solicitação pull.
11. Selecione Criar para criar a solicitação pull.

Depois de vincular os problemas do Jira a uma CodeCatalyst pull request, um resumo da pull request fica disponível. O resumo inclui execuções do fluxo de trabalho, problemas vinculados, revisores obrigatórios, revisores opcionais e o autor.

### Note

As informações de Destinatário e Criado por associadas à edição do Jira não estão disponíveis em. CodeCatalyst

Depois de vincular uma pull request, o CodeCatalyst projeto sincronizado e o projeto Jira permitem que as atualizações sejam refletidas em seu projeto do CodeCatalyst Jira. O status da solicitação pull vinculada e todos os eventos do fluxo de trabalho relacionados à solicitação pull aparecerão no problema do Jira ao visualizá-lo no Jira. Para obter mais informações sobre a visualização de CodeCatalyst eventos no Jira, consulte [Visualizando CodeCatalyst eventos em edições do Jira](#).

## Visualizando CodeCatalyst eventos em edições do Jira

Se seus CodeCatalyst projetos e projetos do Jira estiverem vinculados, o status resumido da pull request e o status dos eventos de CodeCatalyst fluxo de trabalho associados serão refletidos em seu problema no Jira. Por exemplo, se você fechar ou mesclar uma pull request CodeCatalyst, a atualização de status será refletida no problema do Jira. CodeCatalyst CI/CD os eventos do fluxo de trabalho relacionados a uma CodeCatalyst pull request são sincronizados, portanto, uma execução bem-sucedida do fluxo de trabalho também seria enviada para o problema do Jira.

Para visualizar CodeCatalyst eventos em uma edição do Jira

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de CodeCatalyst navegação, escolha Código, escolha Pull requests e, em seguida, escolha a pull request com o problema do Jira que você deseja visualizar em seu projeto do Jira.
4. No painel Informações adicionais, selecione o problema do Jira que você deseja visualizar no seu projeto do Jira.
5. No painel Detalhes do projeto Jira, selecione a solicitação pull listada para Desenvolvimento para ver os detalhes da solicitação pull.
6. (Opcional) Para ver as versões mais recentes, selecione a guia Compilações.
7. (Opcional) Para ver o status de desenvolvimento, selecione a guia Implantações.

# Pesquisa do código, problemas, projetos e usuários no CodeCatalyst

Use a barra de pesquisa ou uma janela de resultados de pesquisa dedicada no CodeCatalyst para pesquisar códigos, problemas, projetos e usuários do CodeCatalyst.

Você pode encontrar recursos em seu espaço e projetos inserindo consultas, como nome, descrição e status na barra de pesquisa. Você também pode refinar suas consultas de pesquisa usando a linguagem da consulta de pesquisa.

## Tópicos

- [Refinamento da sua consulta de pesquisa](#)
- [Considerações ao trabalhar com a pesquisa](#)
- [Referência de campos pesquisáveis](#)

## Para pesquisar

1. Na barra de pesquisa na barra de navegação superior, insira uma consulta de pesquisa.
2. (Opcional) Refine sua consulta de pesquisa usando a linguagem de consulta de pesquisa do CodeCatalyst. Para obter mais informações, consulte [Refinamento da sua consulta de pesquisa](#).
3. Execute um destes procedimentos:
  - Para pesquisar recursos no projeto em que você está atualmente, selecione Este projeto.
  - Para pesquisar recursos em todos os projetos no espaço em que você está atualmente, selecione Este espaço.
4. Visualize os resultados da pesquisa em uma janela de resultados da pesquisa dedicada fazendo o seguinte:
  - Na parte inferior da janela de resultados da pesquisa rápida, selecione Visualizar todos os resultados em nome do projeto | nome do espaço para visualizar todos os resultados da pesquisa.
  - Pressione Enter para visualizar todos os resultados da pesquisa.

 Tip

Mencione outros usuários do projeto em um comentário ou descrição de solicitação pull ou em um comentário ou descrição do problema, usando o sinal @ seguido pelo nome de exibição ou nome de usuário. Você também pode vincular recursos, como problemas ou arquivos de código, usando o sinal @ seguido pelo nome do problema ou do arquivo de código.

## Refinamento da sua consulta de pesquisa

Se você não conseguir encontrar o que está procurando depois de pesquisar, pode refinar sua pesquisa com a linguagem de consulta especializada do CodeCatalyst. Os campos individuais não têm limite de caracteres, mas a consulta geral tem um limite de 1.024 caracteres.

### Tópicos

- [Refinamento por tipo](#)
- [Refinamento por campo](#)
- [Refinamento com operadores booleanos](#)
- [Refinamento por projeto](#)

## Refinamento por tipo

Para refinar o escopo da sua pesquisa para um tipo específico de informação, inclua *type:result-type* em sua pesquisa, onde o *tipo de resultado* é code, issue, project ou user.

### Exemplos:

- `type:code AND java` – Mostre os resultados do código em campos relacionados ao código que contêm “java”.

Para obter mais informações, consulte [Campos de código](#).

- `type:issue AND Bug` – Mostre os resultados do problema em campos relacionados ao problema que contêm “Bug”.

Para obter mais informações, consulte [Campos de problema](#).

- `type:user AND MaryMajor` – Mostre os resultados do usuário em campos relacionados ao usuário que contêm “MaryMajor”.

Para obter mais informações, consulte [Campos de usuário](#).

- `type:project AND Datafeeder` – Mostre os resultados do projeto que contêm “Datafeeder”.

Para obter mais informações, consulte [Campos do projeto](#).

## Refinamento por campo

Para refinar o escopo da sua pesquisa para um campo específico, inclua *field-name:query* na pesquisa, onde o *nome do campo* é `title`, `username`, `project`, `description` e assim por diante, e a *consulta* é o texto que você está pesquisando. Para ver uma lista de campos, consulte [Referência de campos pesquisáveis](#). Você pode pesquisar várias consultas usando parênteses.

Exemplos:

- `title:bug`: mostre os resultados em que o título contém “bug”.
- `username:John`: mostre os resultados em que o nome do usuário contém “John”.
- `project:DataFeeder`: mostre os resultados no projeto “DataFeeder”. A consulta não faz distinção entre maiúsculas e minúsculas.
- `description:overview`: mostre os resultados onde a descrição contém “visão geral”.

## Refinamento com operadores booleanos

Para especificar restrições nas frases de pesquisa, você pode usar os operadores booleanos AND, OR e NOT. Se você listar várias frases, o CodeCatalyst as unirá com OR por padrão. Você pode agrupar frases de pesquisa usando parênteses.

- `exception AND type:code` – Mostre apenas os resultados do código para “exceção”.
- `path:README.md AND repo:ServerlessAPI` – Mostre os resultados para caminhos com “README.md” em que o repositório é denominado “ServerlessAPI”.
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)` – Mostre os resultados para “buildspec.yml” em que o repositório é “ServerlessAPI” ou “ServerlessWebApp”.
- `path:java NOT (path:py OR path:ts)` – Mostre os resultados em que o caminho contém “java”, mas não “py” ou “ts”.



## Refinamento por projeto

Para refinar o escopo da sua pesquisa para um projeto específico, inclua *project:name AND query* em sua pesquisa, onde *nome* é o projeto no qual você está pesquisando e *consulta* é o conteúdo que você está pesquisando.

- *project:name AND query* – Mostre os resultados em que o caminho contém a consulta e o nome do projeto.

## Considerações ao trabalhar com a pesquisa

Atualizações de conteúdo atrasadas – Pode levar vários minutos para que atualizações de conteúdo, como alterações de nome ou reatribuições de problemas, sejam refletidas nos resultados da pesquisa. Grandes atualizações, como uma migração de base de código, podem levar mais tempo para aparecer nos resultados da pesquisa.

Caracteres especiais de escape: os seguintes caracteres especiais requerem consideração especial em suas consultas de pesquisa: + - & & || ! ( ) { } [ ] ^ " ~ \* ? : \. Caracteres especiais não influenciarão a consulta e você deverá removê-los ou evitá-los. Para evitar um caractere, adicione uma barra invertida (\) na frente dele. Por exemplo, a consulta de pesquisa [Recurso] deve ser Recurso ou \[Recurso].

Limitar a pesquisa: a pesquisa não faz distinção entre maiúsculas e minúsculas. Pesquisar em letras minúsculas evita que suas consultas dividam palavras na mudança de maiúsculas e minúsculas. Por exemplo, para consultar por MyService e somente MyService, considere consultar myservice para evitar resultados que contenham somente my ou service.

A pesquisa une palavras e partes de palavras com a conjunção OR-wise por padrão. Por exemplo, `new function` poderia retornar resultados contendo `new` e `function` e também resultados com apenas `new` ou `function`. Para evitar o último, combine várias palavras com AND. Por exemplo, você pode pesquisar por `new AND function`.

Ramificações padrão – A pesquisa retornará somente os resultados do código da confirmação mais recente na ramificação padrão do repositório de origem. Para encontrar código em outras ramificações ou confirmações, considere [clonar o repositório localmente](#), [abrir a ramificação em um Ambiente de Desenvolvimento](#) ou [visualizar as ramificações e os detalhes na interface do usuário do CodeCatalyst](#). A alteração da ramificação padrão resulta em atualizações nos arquivos que podem

ser descobertos pela pesquisa. Para obter mais informações, consulte [Gerenciar a ramificação padrão de um repositório](#).

#### Important

O CodeCatalyst não comporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, primeiro você deve desvinculá-la do CodeCatalyst, alterar a ramificação padrão e vinculá-la novamente. Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#). Como prática recomendada, sempre verifique se você tem a versão mais recente da extensão antes de vincular um repositório.

## Referência de campos pesquisáveis

O CodeCatalyst pesquisa os campos a seguir quando você insere consultas de pesquisa. Aliases são outro nome que você pode usar para referenciar o campo na linguagem de consulta avançada.

### Campos de código

| Campo      | Alias | Descrição                                                                                                                                              |
|------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| branchName | ramo  | Nome da ramificação em que o arquivo de código está.                                                                                                   |
| código     | N/D   | Informações sobre o conteúdo do código na forma de trechos de código indicando partes do código-fonte que corresponderam à pesquisa.                   |
| commitId   | N/D   | ID de confirmação da confirmação na qual o arquivo de código retornado foi atualizado pela última vez. Pode ou não ser o ID de confirmação na ponta do |

| Campo                        | Alias   | Descrição                                                                                                                                                                                                                                                                                                     |
|------------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              |         | nome da ramificação especificado em <code>branchName</code> .                                                                                                                                                                                                                                                 |
| <code>commitMessage</code>   | N/D     | Mensagem de confirmação da confirmação na qual o arquivo de código foi atualizado pela última vez. Pode ou não ser a mensagem de confirmação na ponta do nome da ramificação especificado em <code>branchName</code> e . Se nenhuma mensagem de confirmação for fornecida , esse valor será uma string vazia. |
| <code>filePath</code>        | caminho | Caminho do arquivo desse arquivo de código.                                                                                                                                                                                                                                                                   |
| <code>lastUpdatedBy</code>   | N/D     | Usuário do CodeCatalyst que atualizou o arquivo de código pela última vez. Se o nome de usuário não estiver disponível, esse valor será o endereço de e-mail do usuário conforme configurado no arquivo de configuração do Git.                                                                               |
| <code>lastUpdatedById</code> | N/D     | ID exclusiva gerada pelo sistema do usuário que atualizou o arquivo de código pela última vez. Se o ID de usuário não estiver disponível, esse valor poderá ser o endereço de e-mail do usuário.                                                                                                              |

| Campo           | Alias                 | Descrição                                                                                                                                                    |
|-----------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| lastUpdatedTime | N/D                   | Hora em que os dados da pesquisa foram atualizados pela última vez com a confirmação que continha o arquivo de código (em Tempo Universal Coordenado (UTC)). |
| projectId       | N/D                   | ID exclusivo do projeto gerado pelo sistema.                                                                                                                 |
| projectName     | projectNames, project | O nome de exibição do projeto que contém o repositório de origem no qual o arquivo de código foi confirmado.                                                 |
| repositoryId    | repold                | O ID exclusivo gerado pelo sistema do repositório de origem.                                                                                                 |
| repositoryName  | repositório, repo     | O nome de exibição do repositório de origem no qual o arquivo de código foi confirmado.                                                                      |

## Campos de problema

| Campo         | Alias        | Descrição                                                                   |
|---------------|--------------|-----------------------------------------------------------------------------|
| assigneelds   | assigneeld   | Os IDs exclusivos gerados pelo sistema dos usuários atribuídos ao problema. |
| destinatários | destinatário | Nomes de usuário dos usuários atribuídos ao problema.                       |

| Campo           | Alias    | Descrição                                                                               |
|-----------------|----------|-----------------------------------------------------------------------------------------|
| createdBy       | N/D      | O nome de exibição do usuário que criou o problema.                                     |
| createdById     | N/D      | O ID exclusivo do usuário que criou o problema gerado pelo sistema.                     |
| createTime      | N/D      | A hora em que a edição foi criada (em Tempo Universal Coordenado (UTC)).                |
| description     | N/D      | Descrição do problema.                                                                  |
| isArchived      | archived | Um valor booleano que indica se o problema deve ser criado em um estado arquivado.      |
| isBlocked       | blocked  | Um valor booleano que indica se o problema foi marcado como bloqueado.                  |
| labelIds        | labelId  | IDs exclusivos dos rótulos gerados pelo sistema para um problema.                       |
| lastUpdatedBy   | N/D      | Nome de exibição do usuário que atualizou o problema pela última vez.                   |
| lastUpdatedById | N/D      | O ID exclusivo do usuário que atualizou o problema pela última vez gerado pelo sistema. |

| Campo           | Alias                 | Descrição                                                                                      |
|-----------------|-----------------------|------------------------------------------------------------------------------------------------|
| lastUpdatedTime | N/D                   | A hora em que o problema foi atualizado pela última vez (em Tempo Universal Coordenado (UTC)). |
| priority        | N/D                   | Prioridade do problema, se houver um atribuído.                                                |
| projectId       | N/D                   | ID exclusivo do projeto gerado pelo sistema.                                                   |
| projectName     | projectNames, project | Projeto no qual esse problema pode ser encontrado.                                             |
| shortId         | N/D                   | Identificador abreviado e de incremento automático para o problema.                            |
| status          | N/D                   | Status do problema que indica se o problema está em pendência ou na coluna do quadro.          |
| statusId        | N/D                   | Identificador do status do sistema.                                                            |
| título          | N/D                   | Título do problema.                                                                            |

## Campos do projeto

| Campo           | Alias | Descrição                                               |
|-----------------|-------|---------------------------------------------------------|
| description     | N/D   | Uma descrição do projeto.                               |
| lastUpdatedTime | N/D   | A hora em que os metadados do projeto foram atualizados |

| Campo       | Alias   | Descrição                                                                                                            |
|-------------|---------|----------------------------------------------------------------------------------------------------------------------|
|             |         | pela última vez (em Tempo Universal Coordenado (UTC)).                                                               |
| projectName | project | Nome do projeto no espaço.                                                                                           |
| projectPath | N/D     | Nome roteável por URL do projeto, definido durante a criação do projeto. Usado em URLs que exigem o nome do projeto. |

## Campos de usuário

| Campo            | Alias    | Descrição                                                                                                                                                        |
|------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nome de exibição | N/D      | Nome usado para o usuário no CodeCatalyst. Os nomes de exibição não são exclusivos.                                                                              |
| email            | N/D      | O endereço de e-mail do usuário.                                                                                                                                 |
| lastUpdatedTime  | N/D      | A hora em que os metadados do usuário foram atualizados pela última vez (em formato de Tempo Universal Coordenado (UTC)).                                        |
| userName         | username | Nome de usuário escolhido pelo usuário quando ele se inscreveu no CodeCatalyst. Ao contrário dos nomes de exibição, os nomes de usuário não podem ser alterados. |

# Solução de problemas do Amazon CodeCatalyst

As informações a seguir podem ajudar a solucionar problemas comuns no CodeCatalyst. Você também pode usar o relatório de integridade do Amazon CodeCatalyst para determinar se há problemas de serviço que possam estar afetando sua experiência.

## Tópicos

- [Solução de problemas de acesso gerais](#)
- [Solução de problemas de suporte](#)
- [Alguns ou todos os Amazon CodeCatalyst não estão disponíveis](#)
- [Não consigo criar um projeto no CodeCatalyst](#)
- [Não consigo acessar meu espaço BID como um novo usuário ou não posso ser adicionado como um novo usuário de SSO devido ao nome de usuário truncado](#)
- [Adicionar um usuário do SSO ao meu espaço federado como um novo usuário criou um usuário duplicado](#)
- [Quero enviar feedback no CodeCatalyst](#)
- [Solucionar problemas com repositórios de origem](#)
- [Solução de problemas de projetos e esquemas](#)
- [Solucionar problemas com Ambientes de Desenvolvimento](#)
- [Solucionar problemas com fluxos de trabalho](#)
- [Solução de problemas](#)
- [Solucionar problemas com pesquisa no CodeCatalyst](#)
- [Solução de problemas com extensões](#)
- [Solução de problemas de contas associadas ao seu espaço](#)
- [Solução de problemas entre o Amazon CodeCatalyst e os SDKs da AWS ou AWS CLI](#)

## Solução de problemas de acesso gerais

### Esqueci minha senha

Problema: esqueci a senha que uso para meu ID do builder AWS e Amazon CodeCatalyst.



Possíveis correções: a maneira mais fácil de corrigir esse problema é redefinir sua senha.

1. Abra o [Amazon CodeCatalyst](#) e insira seu endereço de e-mail. Depois, escolha Continuar.
2. Escolha Esqueci a senha
3. Enviaremos um e-mail com um link para você alterar sua senha. Se você não vir o e-mail em sua conta, verifique a pasta de spam.

## Alguns ou todos os Amazon CodeCatalyst não estão disponíveis

Problema: acessei ou segui um link para o console do CodeCatalyst, mas um erro é exibido.

Possíveis correções: os motivos mais comuns para esse problema são que você seguiu um link para um projeto ou espaço para o qual não foi convidado, ou há um problema de disponibilidade geral com o serviço. Confira o [relatório de integridade](#) para ver se há algum problema conhecido com o serviço. Caso contrário, entre em contato com a pessoa que o convidou para o projeto ou espaço e peça outro convite. Se você não foi convidado para nenhum projeto ou espaço, pode se inscrever e [criar seu próprio espaço e projetos](#).

## Não consigo criar um projeto no CodeCatalyst

Problema: quero criar um projeto, mas o botão Criar projeto aparece como indisponível ou recebo uma mensagem de erro.

Possíveis correções: os motivos mais comuns para esse problema são que você está conectado ao console com um ID do builder AWS que não tem o perfil de administrador do espaço. Você deve ter esse perfil para criar projetos em um espaço.

Se você tiver essa função e o botão não aparecer como disponível, pode haver um problema transitório com o serviço. Atualize o navegador e tente novamente.

## Solução de problemas de suporte

### Recebo uma mensagem de erro ao acessar o Suporte para o Amazon CodeCatalyst

Problema: quando escolho a opção Suporte para o Amazon CodeCatalyst, recebo a seguinte mensagem de erro:

## Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Conta da AWS that is the billing account for the space.

Possíveis correções: adicione o perfil necessário para a Conta da AWS que é a conta de faturamento do espaço. A conta designada como conta de faturamento do espaço usa o perfil `AWSRoleForCodeCatalystSupport` e a política gerenciada `AmazonCodeCatalystSupportAccess`. Para obter mais informações, consulte [Criação do perfil `AWSRoleForCodeCatalystSupport` para a conta e o espaço](#).

### Note

Um ID do builder AWS só pode ter suporte para o alias com o qual está autenticado e somente para recursos com base nas permissões no CodeCatalyst. O suporte de conta e faturamento está disponível para todos os usuários do espaço. No entanto, os construtores só podem ter suporte para recursos e informações para os quais tenham permissões no CodeCatalyst.

## Não consigo criar casos de suporte técnico para meu espaço

Problema: não consigo criar casos de suporte técnico para meu espaço.

Correções: um plano Business Support ou Enterprise Support precisa ser adicionado à conta de faturamento de espaço para que os usuários do espaço criem casos de suporte técnico. Peça para que o administrador do espaço adicione um plano de Suporte à sua conta de faturamento do espaço ou acesse <https://repost.aws/> para perguntar à comunidade da AWS.

## Minha conta para casos de suporte não está mais conectada ao meu espaço no CodeCatalyst

Problema: minha conta para casos de suporte não está mais conectada ao meu espaço no CodeCatalyst.

Correções: se um usuário com o perfil de administrador do espaço mudar de conta de faturamento do espaço, isso desconectará o plano de Suporte e todos os casos associados do espaço. Os casos de Suporte associados à antiga conta de faturamento do espaço não estarão mais visíveis no

Suporte para o Amazon CodeCatalyst. O usuário-raiz dessa conta de faturamento pode visualizar e resolver casos antigos a partir do Console de gerenciamento da AWS e pode configurar permissões do IAM para o Suporte para que outros usuários visualizem e resolvam casos antigos. Você não poderá continuar recebendo suporte técnico para o CodeCatalyst da antiga conta de faturamento de espaço por meio do Console de gerenciamento da AWS, mas poderá receber suporte técnico para outros serviços até que seu plano de Suporte seja cancelado.

Para ter mais informações, consulte [Updating, resolving, and reopening your case](#) no Guia do usuário do Suporte.

## Não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) no Suporte para o Amazon CodeCatalyst

Problema: não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) no Suporte para o CodeCatalyst.

Possíveis correções: você só pode abrir casos de suporte do CodeCatalyst a partir do Suporte para o CodeCatalyst. Se você precisar de suporte para serviços ou recursos implantados do CodeCatalyst para outro serviço da AWS, Amazon ou de terceiros, crie um caso por meio do Console de gerenciamento da AWS ou canal de suporte de serviço de terceiros. Para ter mais informações, consulte [Creating support cases and case management](#) no Guia do usuário do Suporte.

## Alguns ou todos os Amazon CodeCatalyst não estão disponíveis

Problema: acessei ou segui um link para o console do CodeCatalyst, mas um erro é exibido.

Possíveis correções: os motivos mais comuns para esse problema são que você seguiu um link para um projeto ou espaço para o qual não foi convidado, ou há um problema de disponibilidade geral com o serviço. Confira o [relatório de integridade](#) para ver se há algum problema conhecido com o serviço. Caso contrário, entre em contato com a pessoa que o convidou para o projeto ou espaço e peça outro convite. Se você não foi convidado para nenhum projeto ou espaço, pode se inscrever e [criar seu próprio espaço e projetos](#).

## Não consigo criar um projeto no CodeCatalyst

Problema: quero criar um projeto, mas o botão Criar projeto aparece como indisponível ou recebo uma mensagem de erro.

Possíveis correções: os motivos mais comuns para esse problema são que você está conectado ao console com um ID do builder AWS que não tem o perfil de administrador do espaço. Você deve ter esse perfil para criar projetos em um espaço.

Se você tiver essa função e o botão não aparecer como disponível, pode haver um problema transitório com o serviço. Atualize o navegador e tente novamente.

## Não consigo acessar meu espaço BID como um novo usuário ou não posso ser adicionado como um novo usuário de SSO devido ao nome de usuário truncado

Problema: o CodeCatalyst trunca os nomes de usuário após 100 caracteres, o que pode fazer com que alguns nomes de usuário pareçam idênticos. Como um novo usuário acessando um espaço do CodeCatalyst, tenho esse problema dependendo do tipo de espaço, da seguinte forma:

- Tenho um ID do builder AWS que quero usar para entrar no CodeCatalyst. Quando tento entrar no espaço, recebo uma mensagem informando que meu nome de usuário não é válido.
- Sou administrador de identidade federado de um espaço do CodeCatalyst que oferece suporte à federação de identidades. Ao adicionar um novo usuário aos usuários e grupos de SSO no Centro de Identidade do IAM, recebo uma mensagem de que o usuário não é válido.

Possíveis correções: o primeiro usuário a entrar no CodeCatalyst ou ser adicionado ao espaço como um usuário SSO com um determinado nome de usuário truncado será bem-sucedido. Qualquer usuário que entrar com um ID do builder AWS ou for adicionado ao Centro de Identidade do IAM depois disso não conseguirá fazer login porque o nome parecerá duplicado. Dependendo do tipo de espaço, siga um destes procedimentos:

- Para entrar no espaço do ID do builder AWS, cadastre-se com um nome de usuário diferente.
- Para adicionar o novo usuário no Centro de Identidade do IAM, adicione-o com um nome de usuário diferente.

### Note

Embora o nome de usuário pareça truncado, o CodeCatalyst mapeia a identidade de uma forma que não é afetada por nomes de usuário com nomes truncados. No entanto, se for criado um nome de usuário igual a um nome de usuário truncado, esse nome de usuário não

estará disponível se outro usuário associado (com o mesmo espaço ou aplicação do Centro de Identidade do IAM) já tiver entrado no CodeCatalyst com esse nome de usuário truncado.

## Adicionar um usuário do SSO ao meu espaço federado como um novo usuário criou um usuário duplicado

Problema: usuários do SSO do CodeCatalyst adicionados a um espaço do CodeCatalyst e depois removidos podem ter um problema na tentativa de reutilização do nome de usuário. Isso pode resultar em um erro semelhante ao seguinte, em que o usuário não pode ser duplicado, mesmo que tenha sido recriado.

### Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Conta da AWS that is the billing account for the space.

Possíveis correções: se um usuário existente do IDC for excluído e um novo usuário for recriado com o mesmo nome de usuário, o novo usuário não poderá fazer login devido ao conflito do nome de usuário com o usuário antigo. Depois que um nome de usuário é adicionado a um espaço como um usuário do SSO, o nome não pode ser usado novamente. Qualquer usuário que entrar com um ID do builder AWS ou for adicionado ao Centro de Identidade do IAM depois disso não conseguirá fazer login porque o nome parecerá duplicado.

Dependendo do tipo de espaço, siga um destes procedimentos:

- Para entrar no espaço do ID do builder AWS, cadastre-se com um nome de usuário diferente.
- Para adicionar o novo usuário no Centro de Identidade do IAM, adicione-o com um nome de usuário diferente.

## Quero enviar feedback no CodeCatalyst

Problema: encontrei um bug no CodeCatalyst e quero enviar um feedback.

Possíveis correções: envie o feedback diretamente no CodeCatalyst.

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.

2. No painel de navegação, selecione Enviar feedback.
3. Selecione o tipo de feedback no menu suspenso e insira seu feedback.

## Solucionar problemas com repositórios de origem

As informações a seguir podem ajudar a solucionar problemas comuns com repositórios de origem no CodeCatalyst.

### Tópicos

- [Atingi o máximo de armazenamento do meu espaço e são exibidos avisos ou erros](#)
- [Eu recebo um erro ao tentar clonar ou enviar para um repositório de origem do Amazon CodeCatalyst](#)
- [Eu recebo um erro ao tentar confirmar ou enviar para um repositório de origem do Amazon CodeCatalyst](#)
- [Preciso de um repositório de origem para meu projeto](#)
- [Meu repositório de origem é totalmente novo, mas contém uma confirmação](#)
- [Quero uma ramificação diferente como minha ramificação padrão](#)
- [Estou recebendo e-mails sobre atividades em solicitações pull](#)
- [Esqueci meu token de acesso pessoal \(PAT\)](#)
- [Uma solicitação pull não exibe as alterações esperadas](#)
- [Uma solicitação pull mostra o status Não mesclável](#)

### Atingi o máximo de armazenamento do meu espaço e são exibidos avisos ou erros

Problema: quero enviar o código para um ou mais repositórios de origem no CodeCatalyst, mas um erro é exibido. No console, vejo uma mensagem na página de repositório de origem informando que atingi o limite de armazenamento do espaço.

Possíveis correções: dependendo do seu perfil no projeto ou no espaço, você pode reduzir o tamanho de um ou mais repositórios de origem, excluir repositórios de origem não usados ou alterar seu nível de faturamento para um que tenha mais armazenamento.

- Para reduzir o tamanho de um repositório de origem em um projeto, exclua ramificações não usadas. Para obter mais informações, consulte [Exclusão de uma ramificação](#) e [Perfil de colaborador](#).
- Para reduzir o armazenamento geral de um espaço, exclua repositórios de origem não usados. Para obter mais informações, consulte [Exclusão de um repositório de origem](#) e [Perfil de administrador do projeto](#).
- Para aumentar a quantidade de armazenamento disponível para seu espaço, altere seu nível de faturamento para um com mais armazenamento. Para ter mais informações, consulte [Changing your CodeCatalyst billing tier](#) no Guia do administrador do Amazon CodeCatalyst.

## Eu recebo um erro ao tentar clonar ou enviar para um repositório de origem do Amazon CodeCatalyst

Problema: quando tento clonar um repositório de origem em um computador local ou em um Ambiente de Desenvolvimento integrado (IDE), recebo um erro de permissões.

Possíveis correções: talvez você não tenha um token de acesso pessoal (PAT) para o ID do builder AWS, talvez não tenha configurado seu sistema de gerenciamento de credenciais com seu PAT ou seu PAT pode ter expirado. Tente uma ou mais das seguintes soluções:

- Crie um token de acesso pessoal (PAT). Para obter mais informações, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).
- Verifique se você aceitou um convite para o projeto que contém o repositório de origem e se você ainda é membro desse projeto. Você não pode clonar um repositório de origem se não for um membro ativo desse projeto. Entre no console e tente navegar até o espaço e o projeto em que você está tentando clonar um repositório de origem. Se você não consegue ver o projeto na lista de projetos do espaço, significa que não é membro desse projeto ou não aceitou um convite para esse projeto. Para obter mais informações, consulte [Aceitando um convite e criando um AWS Builder ID](#).
- Verifique se o comando de clone está formatado corretamente e inclui o ID do builder AWS. Por exemplo:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

- Use o AWS CLI para garantir que você tenha um PAT associado ao ID do builder AWS e que ele não esteja expirado. Se você não tiver um ou o PAT estiver expirado, crie um. Para obter

mais informações, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).

- Tente criar um Ambiente de Desenvolvimento para trabalhar com o código no repositório de origem em vez de cloná-lo em um repositório ou IDE local. Para obter mais informações, consulte [Criar um Ambiente de Desenvolvimento](#).

## Eu recebo um erro ao tentar confirmar ou enviar para um repositório de origem do Amazon CodeCatalyst

Problema: quando tento enviar para um repositório de origem, recebo um erro de permissões.

Possíveis correções: talvez você não tenha um perfil no projeto que permita confirmar e enviar alterações de código para o projeto. Veja seu perfil no projeto em que você está tentando enviar alterações para um repositório de origem. Para obter mais informações, consulte [Obtenção de uma lista dos membros e perfis no projeto](#) e [Concessão de acesso com perfis de usuário](#).

Se você tem um perfil que permite confirmar e enviar alterações, a ramificação em que você está tentando confirmar as alterações pode ter uma regra de ramificação configurada para ela que impede o envio de alterações de código para essa ramificação. Em vez disso, tente criar uma ramificação e enviar seu código para ela. Para obter mais informações, consulte [Gerenciar as ações permitidas para uma ramificação com regras de ramificação](#).

## Preciso de um repositório de origem para meu projeto

Problema: meu projeto não tem um repositório de origem ou eu preciso de outro repositório de origem para meu projeto.

Possíveis correções: alguns projetos são criados sem nenhum recurso. Se você for membro do projeto, poderá criar repositórios de origem para esse projeto no CodeCatalyst. Se alguém com o perfil de administrador do espaço instalar os repositórios do GitHub e conectá-los a uma conta do GitHub, você poderá vincular aos repositórios disponíveis do GitHub para adicioná-los ao seu projeto, se você tiver o perfil de administrador do projeto. Para ter mais informações, consulte [Criação de um repositório de origem](#) e [Vinculação de um repositório de origem](#).



## Meu repositório de origem é totalmente novo, mas contém uma confirmação

**Problema:** acabei de criar um repositório de origem. Ele deve estar vazio, mas contém uma confirmação, uma ramificação e um arquivo README .md.

**Possíveis correções:** esse é o comportamento esperado. Todos os repositórios de origem no CodeCatalyst incluem uma confirmação inicial que define a ramificação padrão como `main` e inclui um código de amostra (se o repositório tiver sido criado para um projeto usando um esquema que inclua um código de amostra) ou um arquivo de marcação modelo para um arquivo README do repositório. Você pode criar ramificações adicionais no console e nos clientes Git. Você pode criar e editar arquivos no console e excluir arquivos em Ambientes de Desenvolvimento e clientes Git.

## Quero uma ramificação diferente como minha ramificação padrão

**Problema:** meu repositório de origem veio com uma ramificação padrão chamada `main`, mas quero uma ramificação diferente como minha ramificação padrão.

**Possíveis correções:** você não pode alterar ou excluir a ramificação padrão nos repositórios de origem no CodeCatalyst. Você pode criar ramificações adicionais e usar essas ramificações nas ações de origem nos fluxos de trabalho. Você também pode optar por vincular os repositórios do GitHub e usá-los como repositórios para o seu projeto.

## Estou recebendo e-mails sobre atividades em solicitações pull

**Problema:** não me inscrevi nem configurei notificações por e-mail sobre a atividade de solicitação pull, mas as estou recebendo mesmo assim.

**Possíveis correções:** as notificações por e-mail são enviadas automaticamente sobre a atividade da solicitação pull. Para obter mais informações, consulte [Análise de código com pull requests na Amazon CodeCatalyst](#).

## Esqueci meu token de acesso pessoal (PAT)

**Problema:** estou usando um PAT para clonar, enviar e extrair código para repositórios de origem, mas perdi o valor do meu token e não consigo encontrá-lo no console do CodeCatalyst.

**Possíveis correções:** a maneira mais rápida de resolver esse problema é criar outro PAT e configurar seu gerenciador de credenciais ou IDE para usar esse novo PAT. Só exibimos o valor de um PAT

quando você o cria. Se você perder esse valor, ele não poderá ser recuperado. Para obter mais informações, consulte [Conceder aos usuários acesso ao repositório com tokens de acesso pessoal](#).

## Uma solicitação pull não exibe as alterações esperadas

Problema: criei uma solicitação pull, mas não vejo as mudanças esperadas entre as ramificações de origem e de destino.

Possíveis correções: isso pode ser causado por vários problemas. Tente uma ou mais das seguintes soluções:

- Você pode estar revisando as alterações entre as revisões mais antigas ou pode não estar vendo as alterações mais recentes. Atualize seu navegador e verifique se você escolheu a comparação entre as revisões que deseja visualizar.
- Nem todas as alterações em uma solicitação pull podem ser exibidas no console. Por exemplo, você não pode visualizar os submódulos do Git no console, portanto, não pode ver as diferenças em um submódulo em uma solicitação pull. Algumas diferenças podem ser muito grandes para serem exibidas. Para obter mais informações, consulte [Cotas para repositórios de origem em CodeCatalyst](#) e [Visualizar um arquivo](#).
- As solicitações pull exibem as diferenças entre a base de mesclagem e qualquer revisão que você escolher. Ao criar uma solicitação pull, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a solicitação pull é criada, a diferença exibida é entre a revisão e sua base de mesclagem. A base de mesclagem é a confirmação que era a ponta da ramificação de destino quando a revisão foi criada. A base de mesclagem pode mudar entre as revisões. Para ter mais informações sobre diferenças e bases de mesclagem no Git, consulte [git-merge-base](#) na documentação do Git.

## Uma solicitação pull mostra o status Não mesclável

Problema: quero mesclar uma solicitação pull, mas seu status é exibido como Não mesclável.

Possíveis correções: isso pode ser causado por um ou mais problemas:

- Todos os revisores necessários para sua solicitação pull devem aprovar uma solicitação pull antes que ela possa ser mesclada. Revise a lista de revisores obrigatórios para qualquer revisor com um ícone de relógio ao lado do nome. Um ícone de relógio indica que o revisor não aprovou a solicitação pull.

**Note**

Se um revisor obrigatório tiver sido removido do seu projeto antes de aprovar a solicitação pull, ela não poderá ser mesclada. Feche a solicitação pull e crie uma nova.

- Pode haver um conflito de mesclagem entre a ramificação de origem e a ramificação de destino. O CodeCatalyst não é compatível com todas as estratégias e opções possíveis de mesclagem do Git. Você pode avaliar as ramificações em busca de conflitos de mesclagem em um Ambiente de Desenvolvimento ou clonar o repositório e usar um IDE ou ferramentas Git para localizar e resolver conflitos de mesclagem. Para obter mais informações, consulte [Mesclar uma solicitação pull](#).

## Solução de problemas de projetos e esquemas

Esta seção pode ajudar você a solucionar alguns problemas comuns que pode encontrar ao trabalhar com projetos e esquemas no Amazon CodeCatalyst.

### API Java com dependências ausentes do esquema AWS Fargate para apache-maven-3.8.6

Problema: para um projeto criado a partir da API Java com o esquema AWS Fargate, o fluxo de trabalho falha com um erro por dependências ausentes de apache-maven-3.8.6. O fluxo de trabalho falha com uma saída semelhante ao seguinte exemplo:

```
Step 8/25 : RUN wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-
maven-3.8.6-bin.tar.gz -P /tmp
---> Running in 1851ce6f4d1b
[91m--2023-03-10 01:24:55-- https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving dlcdn.apache.org (dlcdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443...
[0m[91mconnected.
[0m[91mHTTP request sent, awaiting response... [0m[91m404 Not Found
2023-03-10 01:24:55 ERROR 404: Not Found.
[0mThe command '/bin/sh -c wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz -P /tmp' returned a non-zero code: 8
[Container] 2023/03/10 01:24:55 Command failed with exit status 8
```

Solução: atualize o Dockerfile do esquema usando as etapas a seguir.

1. Na barra de pesquisa, insira `apache-maven-3.8.6` para localizar o `dockerfile` dentro do projeto criado com a API Java com o esquema AWS Fargate.
2. Atualize o Dockerfile (`/static-assets/app/Dockerfile`) para usar `maven:3.9.0-amazoncorretto-11` como imagem base e remova a dependência do pacote `apache-maven-3.8.6`.
3. (Recomendado) Também recomendamos atualizar o tamanho da pilha do Maven para 6 GB.

Confira abaixo um exemplo de Dockerfile.

```
FROM maven:3.9.0-amazoncorretto-11 AS builder

COPY ./pom.xml ./pom.xml
COPY src ./src/

ENV MAVEN_OPTS='-Xmx6g'

RUN mvn -Dmaven.test.skip=true clean package

FROM amazoncorretto:11-alpine

COPY --from=builder target/Customerservice-0.0.1.jar Customerservice-0.0.1.jar
EXPOSE 80
CMD ["java", "-jar", "-Dspring.profiles.active=prod", "/Customerservice-0.0.1.jar", "-server.port=80"]
```

## O fluxo de trabalho moderno de três camadas do blueprint de aplicativos web OnPullRequestfalha com erro de permissões para a Amazon CodeGuru

Problema: quando tento executar um fluxo de trabalho para meu projeto, o fluxo de trabalho não é executado com a seguinte mensagem:

```
Failed at codeguru_codereview: The action failed during runtime. View the action's logs for more details.
```

**Solução:** uma possível causa dessa falha de ação pode ser devido à falta de permissões na política de função do IAM, em que sua versão da função de serviço usada por CodeCatalyst in the connected não tem as permissões necessárias para que a ação `codeguru_codereview` Conta da AWS seja executada com êxito. Para corrigir esse problema, a função de serviço deve ser atualizada com as permissões necessárias ou você deve alterar a função de serviço usada para o fluxo de trabalho para uma que tenha as permissões necessárias para a Amazon CodeGuru e o Amazon CodeGuru Reviewer. Realizando as etapas a seguir, encontre seu perfil e atualize as permissões da política de perfil para que o fluxo de trabalho seja executado com êxito.

 **Note**

Essas etapas se aplicam aos seguintes fluxos de trabalho em: CodeCatalyst

- O `OnPullRequest` fluxo de trabalho fornecido para projetos criados com o modelo moderno de aplicativo web de três camadas em: CodeCatalyst
- Fluxos de trabalho adicionados aos projetos CodeCatalyst com ações que acessam a Amazon CodeGuru ou o Amazon CodeGuru Reviewer.

Cada projeto contém fluxos de trabalho com ações que usam uma função e um ambiente fornecidos pelos Conta da AWS conectados ao seu projeto em CodeCatalyst. O fluxo de trabalho com as ações e a política designada é armazenado em seu repositório de origem no diretório `/.codecatalyst/workflows`. Não é necessário modificar o YAML do fluxo de trabalho, a menos que você esteja adicionando um novo ID de perfil ao fluxo de trabalho existente. Para ter mais informações sobre elementos de modelo e formatação do YAML, consulte [Definição do YAML do fluxo de trabalho](#).

Essas são as etapas detalhadas a serem seguidas para editar sua política de perfil e verificar o YAML do fluxo de trabalho.

Para referenciar o nome do perfil no YAML do fluxo de trabalho e atualizar a política

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço. Navegue até o projeto.
3. Selecione CI/CD e, depois, selecione Fluxos de trabalho.
4. Escolha o fluxo de trabalho intitulado `OnPullRequest`. Escolha a guia Definição.
5. No YAML do fluxo de trabalho, no campo `Role`: abaixo da ação `codeguru_codereview`, anote o nome do perfil. Esse é o perfil com a política que você modificará no IAM. O exemplo a seguir mostra o nome do perfil.

**CI/CD** Workflows Environments Compute Secrets Change tracking

**OnPullRequest** mysfitscvb63 test

Latest state Definition Runs

Latest run Run-9ab13 Run mode Queued Latest commit 14035d52 Workflow definition Valid

WorkflowSource mysfitscvb63 test

codeguru\_codereview  
aws/build@v1  
Environment: development

```

1
2 Name: OnPullRequest
3 SchemaVersion: "1.0"
4 Triggers:
5 - Type: PULLREQUEST
6 Branches:
7 - main
8 Events:
9 - OPEN
10 - REVISION
11 Actions:
12 codeguru_codereview:
13 Identifier: aws/build@v1
14 Inputs:
15 Sources:
16 - WorkflowSource
17 Variables:
18 - Name: AWS_DEFAULT_REGION
19 Value: us-west-2
20 Outputs:
21 Artifacts:
22 - Name: codereview
23 Files:
24 - ./code-guru/*
25 Configuration:
26 Steps:
27 - Run: curl -OL https://github.com/aws/aws-codeguru-cl
28 - Run: unzip aws-codeguru-cli.zip
29 - Run: export PATH=$PATH:./aws-codeguru-cli/bin
30 - Run: aws-codeguru-cli --root-dir ./src --no-prompt -
31 Environment:
32 Name: development
33 Connections:
34 - Name: connection-11-30
35 Role: CodeCatalystPreviewDevelopmentAdministrator-j

```

## 6. Execute um destes procedimentos:

- (Recomendado) Atualize a função de serviço conectada ao seu projeto com as permissões necessárias para a Amazon CodeGuru e o Amazon CodeGuru Reviewer. O perfil terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName` com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-\*spaceName\*](#). Continue com as próximas etapas para atualizar a política no IAM.

### Note

Você deve ter acesso de AWS administrador ao Conta da AWS com a função e a política.

- Altere a função de serviço usada para o fluxo de trabalho para uma que tenha as permissões necessárias para a Amazon CodeGuru e o Amazon CodeGuru Reviewer ou crie uma nova função com as permissões necessárias.
7. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.  
  
No console do IAM, encontre o perfil na etapa 5, como CodeCatalystPreviewDevelopmentRole.
  8. No perfil da etapa 5, altere a política de permissão para incluir as permissões `codeguru-reviewer:*` e `codeguru:*`. Depois de adicionar essas permissões, a política de permissões deve ser semelhante à seguinte:
  9. Depois de fazer as correções de política, retorne CodeCatalyst e inicie a execução do fluxo de trabalho novamente.

## Ainda está tentando resolver seu problema?

Você pode acessar o [Amazon CodeCatalyst](#) ou preencher um [Formulário de feedback de suporte](#).

Na seção Solicitar informações, em Como podemos ajudar, inclua que você é um cliente do Amazon CodeCatalyst. Forneça o máximo de detalhes possível para que possamos resolver seu problema da maneira mais eficiente.

## Solucionar problemas com Ambientes de Desenvolvimento

Consulte as seções a seguir para solucionar problemas relacionados a Ambientes de Desenvolvimento. Para ter mais informações sobre Ambientes de Desenvolvimento, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

### Tópicos

- [A criação do meu Ambiente de Desenvolvimento não foi bem-sucedida devido a um problema com cotas](#)
- [Não consigo enviar alterações do meu Ambiente de Desenvolvimento para uma ramificação específica em um repositório](#)
- [Meu Ambiente de Desenvolvimento não foi retomado](#)
- [Meu Ambiente de Desenvolvimento foi desconectado](#)
- [Meu Ambiente de Desenvolvimento conectado à VPC falhou](#)

- [Não consigo encontrar em qual diretório meu projeto está](#)
- [Não consigo me conectar ao Ambiente de Desenvolvimento via SSH](#)
- [Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque minha configuração SSH local está ausente](#)
- [Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque estou com problemas com AWS Config para o perfil codecatalyst](#)
- [Não consigo criar um Ambiente de Desenvolvimento quando estou conectado ao CodeCatalyst usando uma conta de login único](#)
- [Solução de problemas com IDEs](#)
- [Solução de problemas com devfiles](#)

## A criação do meu Ambiente de Desenvolvimento não foi bem-sucedida devido a um problema com cotas

Problema: quero criar um Ambiente de Desenvolvimento no CodeCatalyst, mas vejo um erro. No console, vejo uma mensagem na página Ambientes de Desenvolvimento informando que atingi o limite de armazenamento do espaço.

Possíveis correções: dependendo do perfil no projeto ou no espaço, você pode excluir um ou mais dos seus próprios Ambientes de Desenvolvimento ou, se for o administrador do espaço, poderá excluir Ambientes de Desenvolvimento não utilizados criados por outros usuários. Você também pode decidir alterar o nível de faturamento para um que inclua mais armazenamento.

- Para ver os limites de armazenamento, consulte a guia Faturamento do espaço do Amazon CodeCatalyst para ver se as cotas de Uso atingiram o máximo permitido. Se as cotas tiverem atingido o máximo, entre em contato com alguém com o perfil de administrador do espaço para remover ambientes de desenvolvimento desnecessários ou considerar a alteração do nível de faturamento.
- Para remover qualquer Ambiente de Desenvolvimento que você criou e que não precisa mais, consulte [Excluir um Ambiente de Desenvolvimento](#).

Se o problema persistir e você receber um erro no IDE, confira se você tem um perfil do CodeCatalyst que permite criar um Ambiente de Desenvolvimento. Os perfis de Administrador do espaço, Administrador do projeto e Colaborador têm permissão para criar ambientes de



desenvolvimento. Para obter mais informações, consulte [Concessão de acesso com perfis de usuário](#).

## Não consigo enviar alterações do meu Ambiente de Desenvolvimento para uma ramificação específica em um repositório

**Problema:** quero confirmar e enviar alterações de código no meu Ambiente de Desenvolvimento para uma ramificação em um repositório de origem, mas vejo um erro.

**Possíveis correções:** dependendo do seu perfil no projeto ou no espaço, talvez você não tenha permissões para enviar o código para os repositórios de origem do projeto. Os perfis de Administrador do espaço, Administrador do projeto e Colaborador têm permissão para enviar código aos repositórios do projeto.

Se você for Colaborador, mas não conseguir enviar o código para uma ramificação específica, pode haver uma regra configurada para a ramificação específica que impede que os usuários com esse perfil enviem o código para essa ramificação específica. Tente enviar suas alterações para uma ramificação diferente ou criar uma ramificação e, depois, enviar seu código para essa ramificação. Para obter mais informações, consulte [Gerenciar as ações permitidas para uma ramificação com regras de ramificação](#).

## Meu Ambiente de Desenvolvimento não foi retomado

**Problema:** meu Ambiente de Desenvolvimento não foi retomado depois que eu o interrompi.

**Possíveis correções:** para corrigir o problema, consulte a guia Faturamento do espaço do Amazon CodeCatalyst para ver se as cotas de Uso atingiram os limites máximos. Se as cotas tiverem atingido o limite máximo, entre em contato com o administrador do espaço para aumentar o nível de faturamento.

## Meu Ambiente de Desenvolvimento foi desconectado

**Problema:** meu Ambiente de Desenvolvimento foi desconectado enquanto eu o estava usando.

**Possíveis correções:** para resolver o problema, verifique a conexão com a Internet. Se você não estiver conectado à Internet, conecte-se e continue trabalhando no Ambiente de Desenvolvimento.

## Meu Ambiente de Desenvolvimento conectado à VPC falhou

**Problema:** associei uma conexão VPC ao meu Ambiente de Desenvolvimento e ela está apresentando erros.

**Possíveis correções:** o Docker usa um dispositivo de camada de link, chamado rede de ponte, que permite que os contêineres conectados à mesma rede de ponte se comuniquem. A ponte padrão geralmente usa a sub-rede 172.17.0.0/16 para rede de contêineres. Se a sub-rede da VPC para a instância do ambiente usar o mesmo intervalo de endereços que já foi usado pelo Docker, poderá ocorrer um conflito de endereço IP. Para resolver um conflito de endereço IP causado pela Amazon VPC e Docker usando o mesmo bloco de endereço IPv4 CIDR, configure um bloco CIDR diferente de 172.17.0.0/16.

### Note

Não é possível alterar o intervalo de endereços IP de uma VPC ou sub-rede existente.

## Não consigo encontrar em qual diretório meu projeto está

**Problema:** não consigo encontrar em qual diretório meu projeto está.

**Possíveis correções:** para localizar seu projeto, altere o diretório para `/projects`. Esse é o diretório onde você pode encontrar seus projetos.

## Não consigo me conectar ao Ambiente de Desenvolvimento via SSH

Para solucionar problemas de conexão com seu Ambiente de Desenvolvimento via SSH, você pode executar o comando `ssh` com a opção `-vvv` de mostrar mais informações sobre como resolver seu problema:

```
ssh -vvv codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

## Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque minha configuração SSH local está ausente

Se sua configuração SSH local (`~/.ssh/config`) estiver ausente ou o conteúdo da seção `Host codecatalyst-dev-env*` estiver desatualizado, você não conseguirá se conectar ao Ambiente de

Desenvolvimento via SSH. Para solucionar esse problema, exclua a seção `Host codecatalyst-dev-env*` e execute o primeiro comando do modal Acesso SSH novamente. Para obter mais informações, consulte [Estabelecer conexão com um Ambiente de Desenvolvimento usando SSH](#).

## Não consigo me conectar ao Ambiente de Desenvolvimento via SSH porque estou com problemas com AWS Config para o perfil **codecatalyst**

Verifique se o AWS Config (`~/.aws/config`) para o perfil `codecatalyst` corresponde ao descrito em [Configurando para usar o AWS CLI com CodeCatalyst](#). Caso contrário, exclua o perfil de `codecatalyst` e execute o primeiro comando do modal Acesso SSH novamente. Para obter mais informações, consulte [Estabelecer conexão com um Ambiente de Desenvolvimento usando SSH](#).

## Não consigo criar um Ambiente de Desenvolvimento quando estou conectado ao CodeCatalyst usando uma conta de login único

Problema: quando estou conectado ao console do CodeCatalyst como usuário de SSO, recebo um erro de exceção desconhecido quando escolho criar um Ambiente de Desenvolvimento no espaço. Quando escolho criar um Ambiente de Desenvolvimento e escolho o IDE para acesso, por exemplo, o AWS Cloud9, tenho problemas semelhantes aos seguintes:

- A página Ambientes de desenvolvimento no console do CodeCatalyst mostra o Ambiente de Desenvolvimento na lista com um estado `FAILED`.
- Uma mensagem de erro semelhante à mensagem abaixo é exibida:

### **An unknown exception happened**

We encountered an unknown exception when launching your Dev Environment. Mention your Dev Environment id `error_message_ID` if you want to report or need any help.

Correções possíveis:

Os Ambientes de Desenvolvimento não estão disponíveis para usuários em espaços em que o Active Directory é usado como provedor de identidade. Os administradores do espaço podem usar um provedor de identidade alternativo para acessar ambientes de desenvolvimento, como o IAM Identity Center. Para ter mais informações sobre o planejamento de um espaço que comporte a federação

de identidades, consulte [Planejamento do espaço que ofereça suporte à federação de identidades](#) no Guia do administrador do CodeCatalyst.

## Solução de problemas com IDEs

Consulte as seções a seguir para solucionar problemas relacionados aos IDEs no CodeCatalyst. Para ter mais informações sobre IDEs, consulte [Criação de um Ambiente de Desenvolvimento em um IDE](#).

### Tópicos

- [Eu tenho versões de imagem de runtime incompatíveis no AWS Cloud9](#)
- [Não consigo acessar meus arquivos em /projects/projects no AWS Cloud9](#)
- [Não consigo iniciar meu Ambiente de Desenvolvimento no AWS Cloud9 usando um devfile personalizado](#)
- [Estou com problemas no AWS Cloud9](#)
- [No JetBrains, não é possível conectar aos ambientes de desenvolvimento por meio do CodeCatalyst](#)
- [Não consigo instalar AWS Toolkit para o meu IDE](#)
- [No meu IDE, não consigo iniciar meus ambientes de desenvolvimento](#)

### Eu tenho versões de imagem de runtime incompatíveis no AWS Cloud9

O AWS Cloud9 está usando versões diferentes do ativo de frontend e da imagem de runtime do backend. Usar versões diferentes pode fazer com que a extensão Git e AWS Toolkit funcionem incorretamente. Para corrigir o problema, navegue até o painel do Ambiente de Desenvolvimento, interrompa o Ambiente de Desenvolvimento e inicie-o novamente. Para corrigir o problema usando APIs, use a API `UpdateDevEnvironment` para atualizar o runtime. Para ter mais informações, consulte [UpdateDevEnvironment](#) na Referência da API do Amazon CodeCatalyst.

### Não consigo acessar meus arquivos em **/projects/projects** no AWS Cloud9

O editor do AWS Cloud9 não consegue acessar os arquivos no diretório `/projects/projects`. Para corrigir o problema, use o terminal do AWS Cloud9 para acessar seus arquivos ou movê-los para um diretório diferente.

## Não consigo iniciar meu Ambiente de Desenvolvimento no AWS Cloud9 usando um devfile personalizado

Sua imagem do devfile pode não ser compatível com o AWS Cloud9. Para corrigir o problema, revise o devfile do seu repositório e do Ambiente de Desenvolvimento correspondente e crie outro para continuar.

## Estou com problemas no AWS Cloud9

Para outros problemas, consulte a seção de solução de problemas no [Guia do usuário do AWS Cloud9](#).

## No JetBrains, não é possível conectar aos ambientes de desenvolvimento por meio do CodeCatalyst

Para corrigir o problema, confira se você tem somente a versão mais recente do JetBrains instalada. Se você tiver várias versões, desinstale as versões mais antigas e registre seu manipulador de protocolo novamente fechando o IDE e o navegador. Em seguida, abra o JetBrains e registre o manipulador de protocolo novamente.

## Não consigo instalar AWS Toolkit para o meu IDE

Para corrigir esse problema no VS Code, instale manualmente AWS Toolkit for Visual Studio Code no [GitHub](#).

Para corrigir esse problema no JetBrains, instale manualmente AWS Toolkit for JetBrains no [GitHub](#).

## No meu IDE, não consigo iniciar meus ambientes de desenvolvimento

Para corrigir esse problema no VS Code, confira se você tem a versão mais recente do VS Code e AWS Toolkit for Visual Studio Code instalada. Se você não tiver a versão mais recente, atualize e inicie seu Ambiente de Desenvolvimento. Para ter mais informações, consulte [Amazon CodeCatalyst para VS Code](#).

Para corrigir esse problema no JetBrains, confira se você tem a versão mais recente do JetBrains e AWS Toolkit for JetBrains instalada. Se você não tiver a versão mais recente, atualize e inicie seu Ambiente de Desenvolvimento. Para ter mais informações, consulte [Amazon CodeCatalyst para JetBrains](#).

## Solução de problemas com devfiles

Consulte as seções a seguir para solucionar problemas relacionados aos devfiles no CodeCatalyst. Para ter mais informações sobre devfiles, consulte [Configuração de um devfile para um Ambiente de Desenvolvimento](#).

### Tópicos

- [Meu Ambiente de Desenvolvimento está usando o devfile universal padrão, embora eu tenha implementado uma imagem personalizada em um devfile personalizado](#)
- [Meu projeto não está sendo criado em meu Ambiente de Desenvolvimento com o devfile universal padrão](#)
- [Quero mover um repositório devfile para um Ambiente de Desenvolvimento](#)
- [Estou com um problema ao iniciar meu devfile](#)
- [Não tenho certeza de como verificar o status do meu devfile](#)
- [Meu devfile não é compatível com as ferramentas fornecidas na imagem mais recente](#)

Meu Ambiente de Desenvolvimento está usando o devfile universal padrão, embora eu tenha implementado uma imagem personalizada em um devfile personalizado

Se o CodeCatalyst encontrar erros ao iniciar um Ambiente de Desenvolvimento que esteja usando um devfile personalizado, o Ambiente de Desenvolvimento usará como padrão o devfile universal padrão. Para corrigir o problema, você pode verificar o erro exato nos logs em `/aws/mde/logs/devfile.log`. Você também pode verificar se a execução de `postStart` foi bem-sucedida em seus logs: `/aws/mde/logs/devfileCommand.log`.

Meu projeto não está sendo criado em meu Ambiente de Desenvolvimento com o devfile universal padrão

Para corrigir o problema, verifique se você não está usando um devfile personalizado. Se você não estiver usando um devfile personalizado, visualize o arquivo `devfile.yaml` no repositório de origem do projeto para localizar e corrigir quaisquer erros.

Quero mover um repositório devfile para um Ambiente de Desenvolvimento

Você pode mover o devfile padrão em `/projects/devfile.yaml` para o seu repositório de código-fonte. Para atualizar a localização do devfile, use o seguinte comando: `/aws/mde/mde start --location repository-name/devfile.yaml`.

## Estou com um problema ao iniciar meu devfile

Se houver algum problema ao iniciar seu devfile, ele entrará no modo de recuperação para que você ainda possa se conectar ao ambiente e corrigir o devfile. No modo de recuperação, a execução de `/aws/mde/mde status` não conterá a localização do devfile.

```
{
 "status": "STABLE"
}
```

Você pode verificar o erro nos logs em `/aws/mde/logs`, corrigir o devfile e tentar executar `/aws/mde/mde start` novamente.

## Não tenho certeza de como verificar o status do meu devfile

Você pode verificar o status do seu devfile executando `/aws/mde/mde status`. Depois de executar esse comando, você provavelmente verá um dos seguintes:

- `{"status": "STABLE", "location": "devfile.yaml" }`

Isso indica que seu devfile está correto.

- `{"status": "STABLE" }`

Isso indica que seu devfile não foi iniciado e entrou no modo de recuperação.

Você pode verificar o erro exato nos logs em `/aws/mde/logs/devfile.log`.

Você também pode verificar se a execução de `postStart` foi bem-sucedida em seus logs: `/aws/mde/logs/devfileCommand.log`.

Para obter mais informações, consulte [Especificar imagens de devfile universais para um Ambiente de Desenvolvimento](#).

## Meu devfile não é compatível com as ferramentas fornecidas na imagem mais recente

No Ambiente de Desenvolvimento, `devfile` ou `devfile postStart` poderá falhar se as ferramentas `latest` não tiverem as ferramentas necessárias para um projeto específico. Para corrigir o problema, faça o seguinte:

1. Navegue até o seu devfile.

2. No devfile, atualize para uma versão de imagem granular em vez de latest. Ele pode ser similar ao seguinte:

```
components:
 - container:
 image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Crie um Ambiente de Desenvolvimento usando o devfile atualizado.

## Solucionar problemas com fluxos de trabalho

Consulte as seções a seguir para solucionar problemas relacionados aos fluxos de trabalho no Amazon CodeCatalyst. Para obter mais informações sobre fluxos de trabalho, consulte [Compilação, teste e implantação com fluxos de trabalho](#).

### Tópicos

- [Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?](#)
- [Como faço para corrigir os erros “A definição do fluxo de trabalho tem n erros”?](#)
- [Como faço para corrigir os erros “Não é possível localizar credenciais” e “ExpiredToken”?](#)
- [Como faço para corrigir os erros “Não foi possível conectar-se ao servidor”?](#)
- [Por que os campos do CodeDeploy estão ausentes no editor visual?](#)
- [Como faço para corrigir erros de recursos do IAM?](#)
- [Como faço para corrigir erros de “instalação npm”?](#)
- [Por que vários fluxos de trabalho têm o mesmo nome?](#)
- [Posso armazenar meus arquivos de definição de fluxo de trabalho em outra pasta?](#)
- [Como adiciono ações em sequência ao meu fluxo de trabalho?](#)
- [Por que meu fluxo de trabalho é validado com sucesso, mas falha em runtime?](#)
- [A descoberta automática não descobre nenhum relatório da minha ação](#)
- [Minha ação falha em relatórios descobertos automaticamente depois que eu configuro os critérios de sucesso](#)
- [A descoberta automática gera relatórios que eu não quero](#)
- [A descoberta automática gera diversos pequenos relatórios para uma única estrutura de teste](#)
- [Os fluxos de trabalho listados em CI/CD não correspondem aos do repositório de origem](#)
- [Não consigo criar ou atualizar fluxos de trabalho](#)



## Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?

Problema: no console do CodeCatalyst, em CI/CD, Fluxos de trabalho, seu fluxo de trabalho é exibido com a seguinte mensagem:

`Workflow is inactive.`

Essa mensagem indica que o arquivo de definição do fluxo de trabalho contém um gatilho que não se aplica à ramificação em que você está atualmente. Por exemplo, seu arquivo de definição de fluxo de trabalho pode conter um gatilho `PUSH` que faz referência à sua ramificação `main`, mas você está em uma ramificação de recursos. Como as alterações que você está fazendo em sua ramificação de recursos não se aplicam a `main` e não iniciarão a execução do fluxo de trabalho em `main`, o CodeCatalyst descomissiona o fluxo de trabalho na ramificação e o marca como `Inactive`.

Correções possíveis:

Se quiser iniciar um fluxo de trabalho na ramificação de recursos, faça o seguinte:

- Em sua ramificação de recursos, no arquivo de definição do fluxo de trabalho, remova a propriedade `Branches` da seção `Triggers` para que fique assim:

```
Triggers:
- Type: PUSH
```

Essa configuração faz com que o gatilho seja ativado ao enviar para qualquer ramificação, incluindo sua ramificação de recursos. Se o gatilho for ativado, o CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e os arquivos de origem em qualquer ramificação para a qual você esteja enviando.

- Em sua ramificação de recursos, no arquivo de definição do fluxo de trabalho, remova a seção `Triggers` e execute o fluxo de trabalho manualmente.
- Em sua ramificação de recursos, no arquivo de definição do fluxo de trabalho, altere a seção `PUSH` para que ela faça referência à sua ramificação de recursos em vez de outra ramificação (como `main`, por exemplo).

### Important

Tome cuidado para não confirmar essas alterações se você não pretende mesclá-las de volta à sua ramificação `main`.

Para ter mais informações sobre como editar o arquivo de definição de fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).

Para ter mais informações sobre gatilhos, consulte [Início da execução automática de um fluxo de trabalho usando gatilhos](#).

## Como faço para corrigir os erros “A definição do fluxo de trabalho tem *n* erros”?

Problema: você vê uma das seguintes mensagens de erro:

Erro 1:

Na página CI/CD, Fluxos de trabalho, abaixo do nome do seu fluxo de trabalho, é possível visualizar:

Workflow definition has *n* errors

Erro 2:

Ao editar um fluxo de trabalho, selecione o botão Validar para que a seguinte mensagem seja exibida na parte superior do console do CodeCatalyst:

The workflow definition has errors. Fix the errors and choose Validate to verify your changes.

Erro 3:

Depois de navegar até a página de detalhes do seu fluxo de trabalho, o seguinte erro será exibido no campo Definição do fluxo de trabalho:

*n* errors

Correções possíveis:

- Selecione CI/CD, Fluxos de trabalho e o nome do fluxo de trabalho que tem o erro. No campo Definição do fluxo de trabalho próximo à parte superior, selecione o link para o erro. Os detalhes sobre o erro são exibidos na parte inferior da página. Siga as dicas de solução de problemas no erro para corrigir o problema.
- Verifique se o arquivo de definição do fluxo de trabalho é um arquivo YAML.
- Verifique se as propriedades YAML no arquivo de definição do fluxo de trabalho estão aninhadas no nível correto. Para ver como as propriedades devem ser aninhadas no arquivo de definição do

fluxo de trabalho, consulte [Definição do YAML do fluxo de trabalho](#) ou a documentação da sua ação, vinculada de [Adição de uma ação a um fluxo de trabalho](#).

- Verifique se os asteriscos (\*) e outros caracteres especiais escaparam corretamente. Para escapar deles, adicione aspas simples ou duplas. Por exemplo:

```
Outputs:
 Artifacts:
 - Name: myartifact
 Files:
 - "**/*"
```

Para ter mais informações sobre caracteres especiais no arquivo de definição de fluxo de trabalho, consulte [Diretrizes e convenções de sintaxe](#).

- Verifique se as propriedades YAML no arquivo de definição do fluxo de trabalho fazem uso correto de letras maiúsculas e minúsculas. Para ter mais informações sobre regras de uso de maiúsculas e minúsculas, consulte [Diretrizes e convenções de sintaxe](#). Para determinar o uso de maiúsculas e minúsculas correto de cada propriedade, consulte [Definição do YAML do fluxo de trabalho](#) ou a documentação da sua ação, vinculada de [Adição de uma ação a um fluxo de trabalho](#).
- Verifique se a propriedade `SchemaVersion` está presente e definida para a versão correta no arquivo de definição do fluxo de trabalho. Para obter mais informações, consulte [SchemaVersion](#).
- Certifique-se de que a seção `Triggers` no arquivo de definição do fluxo de trabalho inclua todas as propriedades necessárias. Para determinar as propriedades necessárias, selecione o gatilho no [editor visual](#) e procure os campos com informações ausentes ou consulte a documentação de referência do gatilho em [Triggers](#).
- Certifique-se de que a propriedade `DependsOn` no arquivo de definição do fluxo de trabalho esteja configurada corretamente e não introduza dependências circulares. Para obter mais informações, consulte [Sequenciar ações](#).
- Certifique-se de que a seção `Actions` no arquivo de definição do fluxo de trabalho inclua pelo menos uma ação. Para obter mais informações, consulte [Ações](#).
- Assegure-se de que cada ação inclua todas as propriedades necessárias. Para determinar as propriedades necessárias, selecione a ação no [editor visual](#) e procure por campos com informações ausentes ou consulte a documentação da sua ação, vinculada de [Adição de uma ação a um fluxo de trabalho](#).
- Assegure-se de que todos os artefatos de entrada tenham artefatos de saída correspondentes. Para obter mais informações, consulte [Definir um artefato de saída](#).

- Assegure-se de que as variáveis definidas em uma ação sejam exportadas para que possam ser usadas em outras ações. Para obter mais informações, consulte [Exportação de uma variável para que outras ações possam usá-la](#).

## Como faço para corrigir os erros “Não é possível localizar credenciais” e “ExpiredToken”?

Problema: durante o trabalho em [Tutorial: Implantar uma aplicação no Amazon EKS](#), você vê uma ou as duas mensagens de erro a seguir na janela do terminal da sua máquina de desenvolvimento:

Unable to locate credentials. You can configure credentials by running "aws configure".

ExpiredToken: The security token included in the request is expired

Correções possíveis:

Esses erros indicam que as credenciais sendo usadas para acessar os serviços da AWS expiraram. Nesse caso, não execute o comando `aws configure`. Em vez disso, use as instruções a seguir para atualizar sua chave de acesso e token de sessão da AWS.

Como atualizar sua chave de acesso e token de sessão da AWS

1. Verifique se você tem o URL do portal de acesso da AWS, o nome de usuário e a senha que você está usando no tutorial completo do Amazon EKS (codecatalyst-eks-user). Você deve ter configurado esses itens ao concluir [Etapa 1: configurar sua máquina de desenvolvimento](#) do tutorial.

### Note

Se você não tiver essas informações, acesse a página de detalhes do codecatalyst-eks-user no IAM Identity Center, selecione Redefinir senha, Gerar uma senha de uso único [...] e Redefinir senha novamente para exibir as informações na tela.

2. Execute um destes procedimentos:

- Cole a URL do portal de acesso da AWS na barra de endereço do seu navegador.

Ou

- Atualize a página do portal de acesso da AWS, se ela já estiver carregada.
3. Faça login com o nome de usuário e senha de `codecatalyst-eks-user`, caso ainda não tenha feito.
  4. Selecione Conta da AWS e, depois, escolha o nome da Conta da AWS a que você atribuiu o usuário `codecatalyst-eks-user` e o conjunto de permissões.
  5. Ao lado do nome do conjunto de permissões (`codecatalyst-eks-permission-set`), selecione Linha de comando ou acesso programático.
  6. Copie os comandos no meio da página. Eles são semelhantes a:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
export AWS_SESSION_TOKEN="session-token"
```

... onde o *token de sessão* é uma longa string aleatória.

7. Cole os comandos no prompt do terminal em sua máquina de desenvolvimento e pressione Enter.

As novas chaves e o token da sessão são carregados.

Você atualizou suas credenciais. Os comandos `AWS CLI`, `eksctl` e `kubectl` agora devem funcionar.

## Como faço para corrigir os erros “Não foi possível conectar-se ao servidor”?

Problema: ao trabalhar no tutorial descrito em [Tutorial: Implantar uma aplicação no Amazon EKS](#), você vê uma mensagem de erro semelhante à seguinte na janela do terminal da sua máquina de desenvolvimento:

```
Unable to connect to the server: dial tcp: lookup long-string.gr7.us-west-2.eks.amazonaws.com on 1.2.3.4:5: no such host
```

Correções possíveis:

Esse erro geralmente indica que as credenciais que o utilitário `kubectl` está usando para se conectar ao seu cluster Amazon EKS expiraram. Para solucionar o problema, atualize as credenciais inserindo o comando a seguir no prompt do terminal:

```
aws eks update-kubeconfig --name codecatalyst-eks-cluster --region us-west-2
```

Em que:

- *codecatalyst-eks-cluster* é substituído pelo nome do seu cluster Amazon EKS.
- *us-west-2* é substituído pela região da AWS em que seu cluster está implantado.

## Por que os campos do CodeDeploy estão ausentes no editor visual?

Problema: você está usando uma ação [Implantar no Amazon ECS](#) e não está vendo os campos do CodeDeploy, como CodeDeploy AppSpec, no editor visual do fluxo de trabalho. Esse problema pode ocorrer porque o serviço Amazon ECS especificado no campo Serviço não está configurado para realizar implantações azuis/verdes.

Correções possíveis:

- Escolha um serviço do Amazon ECS diferente na guia Configuração da ação Implantar no Amazon ECS. Para obter mais informações, consulte [Implantação no Amazon ECS com um fluxo de trabalho](#).
- Configure o serviço do Amazon ECS selecionado para executar implantações azuis/verdes. Para ter mais informações sobre como configurar implantações azuis/verdes, consulte [Implantações azuis/verdes com o CodeDeploy](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

## Como faço para corrigir erros de recursos do IAM?

Problema: você está usando uma ação [Implantar pilha do CloudFormation](#) e vê `##[error] requires capabilities: [capability-name]` nos logs da ação Implantar pilha do CloudFormation.

Possíveis correções: conclua o procedimento a seguir para adicionar o recurso ao arquivo de definição do fluxo de trabalho. Para ter mais informações sobre os recursos do IAM, consulte [Como reconhecer recursos do IAM em modelos do CloudFormation](#) no Guia do usuário do IAM.

Visual

Como adicionar um recurso do IAM usando o editor visual

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.

2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione Visual.
7. No diagrama do fluxo de trabalho, selecione sua ação Implantar pilha CloudFormation.
8. Escolha a guia Configuração.
9. Na parte inferior, selecione Avançado – opcional.
10. Na lista suspensa Capacidades, marque a caixa de seleção ao lado do recurso mencionado na mensagem de erro. Se o recurso não estiver disponível na lista, use o editor YAML para adicioná-lo.
11. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.
13. Se a execução de um novo fluxo de trabalho não for iniciada automaticamente, execute o fluxo de trabalho manualmente para ver se as alterações corrigem o erro. Para ter mais informações sobre como executar manualmente um fluxo de trabalho, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## YAML

Como adicionar um recurso do IAM usando o editor YAML

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
4. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
5. Escolha Editar.
6. Selecione YAML.

7. Na ação Implantar pilha CloudFormation, adicione uma propriedade `capabilities`, como esta:

```
DeployCloudFormationStack:
 Configuration:
 capabilities: capability-name
```

Substitua *capability-name* pelo nome do recurso do IAM exibido na mensagem de erro. Use vírgulas e nenhum espaço para listar vários recursos. Para ter mais informações, consulte a descrição da propriedade `capabilities` em [Ação “Implantar CloudFormation pilha” YAML](#).

8. (Opcional) Selecione Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Selecione Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.
10. Se a execução de um novo fluxo de trabalho não for iniciada automaticamente, execute o fluxo de trabalho manualmente para ver se as alterações corrigem o erro. Para ter mais informações sobre como executar manualmente um fluxo de trabalho, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Como faço para corrigir erros de “instalação npm”?

Problema: sua [ação de implantar AWS CDK](#) ou [ação de bootstrap AWS CDK](#) falha com o erro `npm install`. Esse erro pode ocorrer porque você está armazenando as dependências da aplicação AWS CDK em um registro privado do gerenciador de pacotes de nós (npm) que não pode ser acessado pela ação.

Possíveis correções: use as instruções a seguir para atualizar o arquivo `cdk.json` arquivo da aplicação AWS CDK com informações adicionais de registro e autenticação.

Antes de começar

1. Crie segredos para suas informações de autenticação. Você referenciará esses segredos no arquivo `cdk.json` em vez de fornecer os equivalentes em texto não criptografado. Para criar os segredos:
  - a. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
  - b. Selecione o projeto.



- c. No painel de navegação, escolha CI/CD e Segredos.
- d. Crie dois segredos com as seguintes propriedades:

| Primeiro segredo                                                                                                               | Segundo segredo                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nome: npmUsername                                                                                                              | Nome: npmAuthToken                                                                                                                                                                                                                                               |
| Valor: <i>npm-username</i> , onde <i>npm-username</i> é o nome de usuário usado para autenticação no seu registro npm privado. | Valor: <i>npm-auth-token</i> , onde <i>npm-auth-token</i> é o token de acesso usado para autenticação no seu registro npm privado. Para ter mais informações sobre tokens de acesso npm, consulte <a href="#">Sobre tokens de acesso</a> na documentação do npm. |
| (Opcional) Descrição: The username used to authenticate to the private npm registry.                                           | (Opcional) Descrição: The access token used to authenticate to the private npm registry.                                                                                                                                                                         |

Para ter mais informações sobre segredos, consulte [Mascarar dados usando segredos](#).

2. Adicione os segredos como variáveis de ambiente à sua ação AWS CDK. A ação substituirá as variáveis por valores reais quando for executada. Para adicionar os segredos:
  - a. No painel de navegação, escolha CI/CD e Fluxos de trabalho.
  - b. Selecione o nome do fluxo de trabalho. É possível filtrar pelo nome do repositório ou da ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome ou o status do fluxo de trabalho.
  - c. Escolha Editar.
  - d. Selecione Visual.
  - e. No diagrama do fluxo de trabalho, escolha sua ação AWS CDK.
  - f. Escolha a guia Entradas.
  - g. Adicione duas variáveis com as seguintes propriedades:

| Primeira variável              | Segunda variável                |
|--------------------------------|---------------------------------|
| Nome: NPMUSER                  | Nome: NPMTOKEN                  |
| Valor: \${Secrets.npmUsername} | Valor: \${Secrets.npmAuthToken} |

Agora você tem duas variáveis contendo referências a segredos.

O código YAML do arquivo de definição de fluxo de trabalho deve ser semelhante ao seguinte:

 Note

O exemplo de código a seguir é de uma ação de AWS CDKbootstrap; uma ação de AWS CDKimplantação será semelhante.

```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
 CDKBootstrapAction:
 Identifier: aws/cdk-bootstrap@v2
 Inputs:
 Variables:
 - Name: NPMUSER
 Value: ${Secrets.npmUsername}
 - Name: NPMTOKEN
 Value: ${Secrets.npmAuthToken}
 Sources:
 - WorkflowSource
 Environment:
 Name: Dev2
 Connections:
 - Name: account-connection
 Role: codecatalystAdmin
 Configuration:
 Parameters:
 Region: "us-east-2"
```

Agora, você está pronto para usar as variáveis NPMUSER e NPMTOKEN em seu arquivo `cdk.json`. Vá para o próximo procedimento.

Como atualizar o arquivo `cdk.json`

1. Vá para o diretório raiz do seu projeto AWS CDK e abra o arquivo `cdk.json`.
2. Encontre a propriedade `"app":` e altere-a para incluir o código exibido em *itálico vermelho*:

 Note

O código de exemplo a seguir é de um projeto TypeScript. Se você estiver usando um projeto JavaScript, o código parecerá semelhante, mas não idêntico.

```
{
 "app": "npm set registry=https://your-registry/folder/CDK-package/ --
userconfig .npmrc && npm set //your-registry/folder/CDK-package/:always-auth=true
--userconfig .npmrc && npm set //your-registry/folder/CDK-package/:_authToken=
\"${NPMUSER}\"\":\"${NPMTOKEN}\" && npm install && npx ts-node --prefer-ts-exts bin/
hello-cdk.ts/js",
 "watch": {
 "include": [
 "***"
],
 "exclude": [
 "README.md",
 "cdk*.json",
 "**/*.d.ts",
 "**/*.js",
 "tsconfig.json",
 "package*.json",
]
 },
 ...
}
```

3. No código destacado em *itálico vermelho*, substitua:
  - *your-registry/folder/CDK-package/* pelo caminho para as dependências do seu projeto AWS CDK em seu registro privado.

- `hello-cdk.ts|.js` pelo nome do seu arquivo de ponto de entrada. Ele pode ser um arquivo `.ts` (TypeScript) ou `.js` (JavaScript), dependendo da linguagem que você está usando.

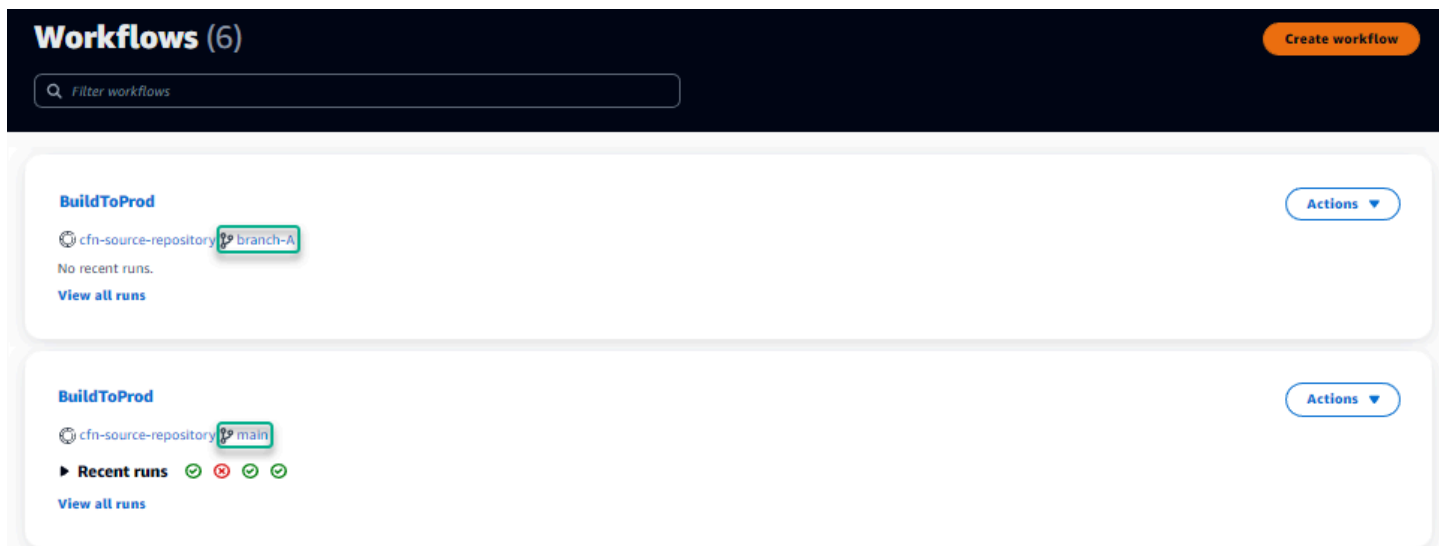
 Note

A ação substituirá as variáveis `NPMUSER` e `NPMTOKEN` pelo nome de usuário e token de acesso npm que você especificou em Segredos.

4. Salve o arquivo `cdk.json`.
5. Execute mais uma vez a ação manualmente para ver se as alterações corrigem o erro. Para ter mais informações sobre como executar ações manualmente, consulte [Iniciar um fluxo de trabalho executado manualmente](#).

## Por que vários fluxos de trabalho têm o mesmo nome?

Os fluxos de trabalho são armazenados por ramificação, por repositório. Dois fluxos de trabalho diferentes podem ter o mesmo nome se existirem em ramificações diferentes. Na página Fluxos de trabalho, você pode diferenciar fluxos de trabalho com o mesmo nome observando o nome da ramificação. Para obter mais informações, consulte [Organizando seu trabalho de código-fonte com filiais na Amazon CodeCatalyst](#).



The screenshot shows the 'Workflows (6)' section of the Amazon CodeCatalyst interface. At the top, there is a search bar labeled 'Filter workflows' and a 'Create workflow' button. Below this, two workflow cards are displayed. Both cards are titled 'BuildToProd'. The first card shows it is for the 'cfn-source-repository' repository and the 'branch-A' branch. It indicates 'No recent runs' and has a 'View all runs' link. The second card also shows it is for the 'cfn-source-repository' repository but for the 'main' branch. It displays 'Recent runs' with four status icons (green, red, green, green) and a 'View all runs' link. Both cards have an 'Actions' button in the top right corner.

## Posso armazenar meus arquivos de definição de fluxo de trabalho em outra pasta?

Não, você deve armazenar todos os arquivos de definição de fluxo de trabalho na pasta `.codecatalyst/workflows` ou nas subpastas dessa pasta. Se você estiver usando um repositório mono com vários projetos lógicos, coloque todos os seus arquivos de definição de fluxo de trabalho na pasta `.codecatalyst/workflows` ou em uma de suas subpastas e, depois, use o campo Arquivos alterados (editor visual) ou a propriedade `FilesChanged` (editor YAML) dentro de um gatilho para acionar o fluxo de trabalho automaticamente em um caminho de projeto especificado. Para obter mais informações, consulte [Adição de gatilhos aos fluxos de trabalho](#) e [Exemplo: um gatilho com um envio, ramificações e arquivos](#).

## Como adiciono ações em sequência ao meu fluxo de trabalho?

Por padrão, quando você adiciona uma ação ao seu fluxo de trabalho, ela não terá dependências e será executada paralelamente a outras ações.

Se quiser organizar as ações em sequência, defina uma dependência em outra ação configurando o campo `DependsOn`. Você também pode configurar uma ação para consumir artefatos ou variáveis que são saídas de outras ações. Para obter mais informações, consulte [Sequenciar ações](#).

## Por que meu fluxo de trabalho é validado com sucesso, mas falha em runtime?

Se você validou seu fluxo de trabalho usando o botão `Validate`, mas ele falhou mesmo assim, pode haver uma limitação no validador.

Qualquer erro em referência a um recurso do CodeCatalyst, como segredos, ambientes ou frotas, na configuração do fluxo de trabalho não será registrado durante uma confirmação. Se alguma referência não válida for usada, o erro só será identificado quando um fluxo de trabalho for executado. Da mesma forma, se houver algum erro na configuração da ação, como falta de um campo obrigatório ou erros de digitação nos atributos da ação, eles serão identificados somente quando o fluxo de trabalho for executado. Para obter mais informações, consulte [Criação de um fluxo de trabalho](#).

## A descoberta automática não descobre nenhum relatório da minha ação

Problema: configurei a descoberta automática para uma ação que executa testes, mas nenhum relatório é descoberto pelo CodeCatalyst.

Possíveis correções: isso pode ser causado por vários problemas. Tente uma ou mais das seguintes soluções:

- Certifique-se de que a ferramenta usada para executar testes produza saídas em um dos formatos que o CodeCatalyst entende. Por exemplo, se você quiser que o `pytest` permita que o CodeCatalyst descubra relatórios de teste e cobertura de código, inclua os seguintes argumentos:

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Para obter mais informações, consulte [Tipos de relatório de qualidade](#).

- Certifique-se de que a extensão do arquivo para as saídas seja consistente com o formato escolhido. Por exemplo, ao configurar `pytest` para produzir resultados no formato JUnitXML, verifique se a extensão do arquivo é `.xml`. Para obter mais informações, consulte [Tipos de relatório de qualidade](#).
- Certifique-se de que `IncludePaths` esteja configurado para incluir todo o sistema de arquivos (`**/*`), a menos que você esteja excluindo determinadas pastas propositalmente. Da mesma forma, certifique-se de que `ExcludePaths` não exclua os diretórios onde você espera que seus relatórios estejam localizados.
- Se você configurou manualmente um relatório para usar um arquivo de saída específico, ele será excluído da descoberta automática. Para obter mais informações, consulte [Exemplo de relatórios de qualidade YAML](#).
- A descoberta automática pode não encontrar relatórios porque a ação falhou antes que qualquer saída fosse gerada. Por exemplo, a compilação pode ter falhado antes da execução de qualquer teste de unidade.

## Minha ação falha em relatórios descobertos automaticamente depois que eu configuro os critérios de sucesso

Problema: quando eu habilito a descoberta automática e configuro os critérios de sucesso, alguns dos relatórios não atendem aos critérios de sucesso e, como resultado, a ação falha.

Possíveis correções: para resolver isso, tente uma ou mais das seguintes soluções:

- Modifique `IncludePaths` ou `ExcludePaths` para excluir relatórios nos quais você não está interessado.

- Atualize os critérios de sucesso para permitir que todos os relatórios sejam aprovados. Por exemplo, se dois relatórios foram descobertos com um com cobertura de linha de 50% e outro de 70%, ajuste a cobertura mínima da linha para 50%. Para obter mais informações, consulte [Critérios de sucesso](#).
- Transforme o relatório de falha em um relatório configurado manualmente. Isso permite que você configure diferentes critérios de sucesso para esse relatório específico. Para obter mais informações, consulte [Configuração de critérios de sucesso para relatórios](#).

## A descoberta automática gera relatórios que eu não quero

Problema: quando eu habilito a descoberta automática, ela gera relatórios que eu não quero. Por exemplo, o CodeCatalyst gera relatórios de cobertura de código para arquivos incluídos nas dependências da minha aplicação armazenadas em `node_modules`.

Possíveis correções: você pode ajustar a configuração de `ExcludePaths` para excluir arquivos indesejados. Por exemplo, para excluir `node_modules`, adicione `node_modules/**/*.*`. Para obter mais informações, consulte [Inclusão/exclusão de caminhos](#).

## A descoberta automática gera diversos pequenos relatórios para uma única estrutura de teste

Problema: quando uso determinadas estruturas de relatórios de cobertura de código e teste, percebi que a descoberta automática gera um grande número de relatórios. Por exemplo, ao usar o [plug-in Maven Surefire](#), a descoberta automática produz um relatório diferente para cada classe de teste.

Possíveis correções: sua estrutura pode ser capaz de agregar saídas em um único arquivo. Por exemplo, se você estiver usando o plug-in Maven Surefire, use `npx junit-merge` para agregar os arquivos manualmente. A expressão completa pode ser semelhante a:

```
mvn test; cd test-package-path/surefire-reports && npx junit-merge -d ./ && rm
*Test.xml
```

## Os fluxos de trabalho listados em CI/CD não correspondem aos do repositório de origem

Problema: os fluxos de trabalho exibidos na página CI/CD, Fluxos de trabalho não correspondem aos da pasta `~/.codecatalyst/workflows/` em seu [repositório de origem](#). Você poderá ver as seguintes incompatibilidades:

- Um fluxo de trabalho é exibido na página Fluxos de trabalho, mas um arquivo de definição de fluxo de trabalho correspondente não existe no seu repositório de origem.
- Existe um arquivo de definição de fluxo de trabalho no seu repositório de origem, mas um fluxo de trabalho correspondente não é exibido na página Fluxos de trabalho.
- Existe um fluxo de trabalho no repositório de origem e na página Fluxos de trabalho, mas eles são diferentes.

Esse problema pode ocorrer se a página Fluxos de trabalho não estiver atualizada ou se uma cota de fluxo de trabalho tiver excedido.

Correções possíveis:

- Espere. Normalmente, você precisa esperar dois ou três segundos após uma confirmação na origem antes de ver a alteração na página Fluxos de trabalho.
- Se você excedeu uma cota de fluxo de trabalho, siga um destes procedimentos:

### Note

Para determinar se uma cota de fluxo de trabalho foi excedida, revise [Cotas para fluxos de trabalho em CodeCatalyst](#) e compare as cotas documentadas com os fluxos de trabalho em seu repositório de origem ou na página Fluxos de trabalho. Não há mensagem de erro que indique que uma cota foi excedida, então você terá que investigar por conta própria.

- Se você excedeu a cota de Número máximo de fluxos de trabalho por espaço, exclua alguns fluxos de trabalho e, depois, realize um teste de confirmação no arquivo de definição do fluxo de trabalho. Um exemplo de confirmação de teste pode ser adicionado em um espaço no arquivo.
- Se você excedeu a cota de Tamanho máximo do arquivo de definição de fluxo de trabalho, altere o arquivo de definição de fluxo de trabalho para reduzir seu tamanho.



- Se você excedeu a cota de Número máximo de arquivos de fluxo de trabalho processados em um único evento de origem, realize várias confirmações de teste. Modifique menos do que o número máximo de fluxos de trabalho em cada confirmação.

## Não consigo criar ou atualizar fluxos de trabalho

Problema: quero criar ou atualizar um fluxo de trabalho, mas vejo um erro quando tento confirmar a alteração.

Possíveis correções: dependendo do seu perfil no projeto ou no espaço, talvez você não tenha permissões para enviar o código para os repositórios de origem do projeto. Os arquivos YAML para fluxos de trabalho são armazenados em repositórios. Para obter mais informações, consulte [Arquivos de definição de fluxo de trabalho](#). Os perfis de Administrador do espaço, Administrador do projeto e Colaborador têm permissão para confirmar e enviar o código aos repositórios do projeto.

Se você tiver o perfil de Colaborador, mas não consegue criar ou confirmar alterações no fluxo de trabalho YAML em uma ramificação específica, pode haver uma regra de ramificação configurada para essa ramificação que impede que usuários com esse perfil enviem código para essa ramificação específica. Tente criar um fluxo de trabalho ou confirmar suas alterações em uma ramificação diferente. Para obter mais informações, consulte [Gerenciar as ações permitidas para uma ramificação com regras de ramificação](#).

## Solução de problemas

As informações a seguir podem ajudar a solucionar problemas comuns no CodeCatalyst.

### Tópicos

- [Não consigo selecionar um destinatário para o meu problema](#)

## Não consigo selecionar um destinatário para o meu problema

Problema: ao criar um problema, a lista de destinatários fica vazia.

Possíveis correções: a lista de destinatários está diretamente vinculada aos usuários do CodeCatalyst listados como membros do projeto. Para verificar se o acesso ao perfil do usuário está funcionando corretamente, selecione o ícone do perfil e Perfil do usuário. Se as informações do perfil do usuário não forem preenchidas, confira se há incidentes no relatório de integridade. Se ele for preenchido, registre um tíquete de serviço.

# Solucionar problemas com pesquisa no CodeCatalyst

Consulte as seções a seguir para solucionar problemas relacionados a pesquisas no CodeCatalyst. Para obter mais informações sobre fluxos de trabalho, consulte [Pesquisa do código, problemas, projetos e usuários no CodeCatalyst](#).

## Tópicos

- [Não consigo encontrar um usuário no meu projeto](#)
- [Não vejo o que estou procurando no meu projeto ou espaço](#)
- [O número de resultados de pesquisa continua mudando quando eu navego pelas páginas](#)
- [Minha consulta de pesquisa não está sendo concluída](#)

## Não consigo encontrar um usuário no meu projeto

Problema: quando tento visualizar os detalhes de um usuário, não vejo suas informações no projeto.

Possíveis correções: no momento, a pesquisa não é compatível com a pesquisa de usuários em um projeto. Para pesquisar usuários com acesso ao seu espaço, alterne para Este espaço na Pesquisa rápida ou remova quaisquer filtros de projeto que você possa ter especificado usando a linguagem de consulta avançada.

## Não vejo o que estou procurando no meu projeto ou espaço

Problema: os resultados não aparecem quando tento pesquisar informações específicas.

Possíveis correções: é provável que as atualizações de conteúdo levem alguns segundos para serem atualizadas nos resultados da pesquisa. Atualizações grandes podem levar alguns minutos.

Para recursos que não foram atualizados recentemente, talvez seja necessário refinar sua pesquisa. Você pode refinar adicionando mais palavras-chave ou usando a linguagem de consulta avançada. Para ter mais informações sobre como refinar suas consultas, consulte [Refinamento da sua consulta de pesquisa](#).

## O número de resultados de pesquisa continua mudando quando eu navego pelas páginas

Problema: o número de resultados da pesquisa parece mudar quando eu vou para a próxima página, então não está claro quantos resultados totais existem.

Possíveis correções: ao navegar pelas páginas dos resultados da pesquisa, você pode ver uma alteração no número de resultados de pesquisa que correspondem à sua consulta. O número de resultados pode ser atualizado para refletir um número mais preciso de correspondências descobertas conforme você navega pelas páginas.

Ao navegar pelos resultados, você pode ver a seguinte mensagem: Nenhum resultado para “teste”. Você receberá a mensagem caso não tenha acesso aos demais resultados.

## Minha consulta de pesquisa não está sendo concluída

Problema: os resultados da minha consulta de pesquisa não estão aparecendo e parece que estão demorando muito.

Possíveis correções: sua pesquisa pode não ser concluída quando há muitas pesquisas sendo feitas ao mesmo tempo no espaço, seja de forma programática ou devido à alta atividade da equipe. Se você estiver executando pesquisas programáticas, pause-as ou diminua-as. Caso contrário, tente novamente em alguns segundos.

## Solução de problemas com extensões

Consulte as seções a seguir para solucionar problemas relacionados às extensões no CodeCatalyst. Para ter mais informações sobre extensões, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

### Tópicos

- [Não consigo ver as alterações em um repositório de terceiros vinculado nem pesquisar os resultados dessas alterações](#)

## Não consigo ver as alterações em um repositório de terceiros vinculado nem pesquisar os resultados dessas alterações

Problema: as alterações no meu repositório de terceiros não são exibidas no CodeCatalyst.

Possíveis correções: o CodeCatalyst não comporta a detecção de alterações na ramificação padrão para repositórios vinculados. Para alterar a ramificação padrão de um repositório vinculado, primeiro você deve desvinculá-la do CodeCatalyst, alterar a ramificação padrão e vinculá-la novamente.

Para obter mais informações, consulte [Vinculando GitHub repositórios, repositórios do Bitbucket, repositórios de GitLab projetos e projetos do Jira no CodeCatalyst](#).

# Solução de problemas de contas associadas ao seu espaço

No CodeCatalyst, você pode adicionar uma Conta da AWS ao seu espaço para conceder permissões aos recursos e para fins de faturamento. As informações a seguir podem ajudar a solucionar problemas comuns com contas associadas no CodeCatalyst.

## Tópicos

- [Minha solicitação de conexão da Conta da AWS recebe um erro de token inválido](#)
- [Meu fluxo de trabalho do projeto Amazon CodeCatalyst falha com um erro na conta configurada, no ambiente ou no perfil do IAM](#)
- [Preciso de uma conta, perfil e ambiente associados para criar um projeto](#)
- [Não consigo acessar a página de espaços do Amazon CodeCatalyst no Console de gerenciamento da AWS](#)
- [Quero uma conta diferente como minha conta de faturamento](#)
- [O fluxo de trabalho do meu projeto falha com um erro de nome de conexão](#)

## Minha solicitação de conexão da Conta da AWS recebe um erro de token inválido

Problema: ao criar uma solicitação de conexão com um token de conexão, a página não aceita o token e mostra um erro informando que o token não é válido.

Possíveis correções: forneça o ID da conta que você deseja adicionar ao seu espaço. Você deve ter permissões administrativas para sua Conta da AWS ou ser capaz de trabalhar com seu administrador para adicionar a conta.

Quando você optar por verificar a conta, uma nova janela do navegador será aberta no Console de gerenciamento da AWS. É necessário que a mesma conta esteja logada no lado do console. Tente novamente depois de verificar o seguinte:

- Você está logado no Console de gerenciamento da AWS com a mesma Conta da AWS que deseja adicionar ao seu espaço.
- Você está logado no Console de gerenciamento da AWS com a região definida corretamente para o seu espaço.

- Se você chegou da página de faturamento e deseja adicionar a Conta da AWS como uma conta de faturamento especificada ao seu espaço, verifique se a conta não atingiu a cota como conta de faturamento para outro espaço ou espaços.

## Meu fluxo de trabalho do projeto Amazon CodeCatalyst falha com um erro na conta configurada, no ambiente ou no perfil do IAM

Problema: quando o fluxo de trabalho é executado e não encontra uma conta configurada ou perfis do IAM associados ao seu espaço, você deve preencher os campos de perfil, conexão e ambiente manualmente no YAML do fluxo de trabalho. Visualize a ação do fluxo de trabalho com falha e observe se as mensagens de erro são as seguintes:

- O perfil não está disponível para uso com a conexão associada ao ambiente.
- A ação não foi bem-sucedida. Status: FALHA; O valor fornecido para a conexão da conta ou do ambiente não é válido. Verifique se a conexão está associada ao seu espaço e se o ambiente está associado ao seu projeto.
- A ação não foi bem-sucedida. Status: FALHA; O valor fornecido para o perfil do IAM não é válido. Verifique se o nome existe, se o perfil do IAM foi adicionado à conexão da sua conta e se a conexão já está associada ao seu espaço do Amazon CodeCatalyst

Possíveis correções: verifique se os campos YAML do fluxo de trabalho têm valores precisos para [Ambiente](#), [Conexões](#) e [Perfil](#). As ações do fluxo de trabalho do CodeCatalyst que exigem um ambiente são ações de compilação ou implantação que executam recursos da AWS ou geram pilhas de recursos da AWS.

Selecione o bloco de ação do fluxo de trabalho com falha e Visual. Escolha a guia Configuração. Se os campos Ambiente, Nome da conexão e Nome do perfil não estiverem preenchidos, você precisará atualizar manualmente o fluxo de trabalho. Use as etapas a seguir para editar seu YAML de fluxo de trabalho:

- Expanda o diretório `/.codecatalyst` e, depois, expanda o diretório `/workflows`. Abra o arquivo YAML do fluxo de trabalho. Certifique-se de que os perfis do IAM e as informações da conta estejam especificadas no YAML configurado para seu fluxo de trabalho. Exemplo:

```
Actions:
 cdk_bootstrap:
 Identifier: action-@v1
```

```
Inputs:
 Sources:
 - WorkflowSource
 Environment:
 Name: Staging
 Connections:
 - Name: account-connection
 Role: build-role
```

As propriedades Ambiente, Conexão e Perfil são necessárias para executar ações de compilação e implantação do fluxo de trabalho do CodeCatalyst com recursos da AWS. Para ver um exemplo, consulte os parâmetros YAML de referência da ação de criação do CodeCatalyst para [Ambiente](#), [Conexões](#) e [Perfil](#).

- Certifique-se de que seu espaço tenha uma conta adicionada a ele e de que a conta tenha o perfil ou perfis do IAM apropriados adicionados à conta. Você pode ajustar ou adicionar contas se tiver o perfil de administrador do espaço. Para obter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

## Preciso de uma conta, perfil e ambiente associados para criar um projeto

Problema: nas opções de criação do projeto, meu projeto não tem uma conta adicional disponível no meu espaço ou eu preciso adicionar outra conta ao meu espaço para uso do meu projeto.

Possíveis correções: para o seu espaço, você pode adicionar Contas da AWS autorizadas para adicioná-las ao seu projeto se tiver o perfil de administrador do espaço. Você também deve ter uma Conta da AWS com permissões administrativas ou que pode trabalhar com seu administrador da AWS.

Para garantir que uma conta e um perfil estejam disponíveis na tela de criação do projeto, primeiro adicione a conta e os perfis. Para obter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Você tem a opção de escolher criar um perfil de serviço com uma política de perfil chamada política de perfil de CodeCatalystWorkflowDevelopmentRole-**spaceName**. O perfil terá um nome CodeCatalystWorkflowDevelopmentRole-**spaceName** com um identificador exclusivo anexado. Para ter mais informações sobre o perfil e a política de perfis, consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole-\*\*spaceName\*\*](#). Para as etapas necessárias para criar o perfil, consulte [Criar o perfil CodeCatalystWorkflowDevelopmentRole-\*\*spaceName\*\* para a conta e o](#)

[espaço](#). O perfil é adicionado à sua conta e está disponível nas páginas de criação de projetos no CodeCatalyst.

## Não consigo acessar a página de espaços do Amazon CodeCatalyst no Console de gerenciamento da AWS

Problema: quando tento acessar a página do Amazon CodeCatalyst no Console de gerenciamento da AWS para adicionar uma conta ao meu espaço do CodeCatalyst ou adicionar perfis a uma conta na AWS, recebo um erro de permissão.

Correções possíveis:

Para o seu espaço, você pode adicionar Contas da AWS autorizadas para adicioná-las ao seu projeto se tiver o perfil de administrador do espaço. Você também deve ter uma Conta da AWS com permissões administrativas ou que pode trabalhar com seu administrador da AWS. Primeiro, certifique-se de estar conectado ao Console de gerenciamento da AWS com a mesma conta que deseja gerenciar. Depois de fazer login no Console de gerenciamento da AWS, abra o console e tente novamente.

Abra a página do Amazon CodeCatalyst no Console de gerenciamento da AWS em <https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#/>.

## Quero uma conta diferente como minha conta de faturamento

Problema: quando configurei meu login do CodeCatalyst, concluí várias etapas para configurar meu espaço e associar uma Conta da AWS autorizada. Agora, quero autorizar uma conta diferente para faturamento.

Possíveis correções: para seu espaço, você pode autorizar contas de faturamento se tiver a função de administrador do espaço. Você também deve ter uma Conta da AWS com permissões administrativas ou que pode trabalhar com seu administrador da AWS.

Para ter mais informações, consulte [Managing billing](#) no Guia do administrador do Amazon CodeCatalyst.

## O fluxo de trabalho do meu projeto falha com um erro de nome de conexão

Problema: ao criar um projeto e depois executar o fluxo de trabalho do projeto, o fluxo de trabalho falha e mostra um erro de que o nome da conexão não é válido, da seguinte forma:

Falha em <action\_name>: o nome da conexão não é válido.

Possíveis correções: forneça o ID da conta que você deseja adicionar ao seu espaço e certifique-se de que a conta não esteja habilitada para conexões de contas restritas ao projeto. Se a conta estiver habilitada para conexões de conta restritas ao projeto, talvez seja necessário atualizar a conexão da conta habilitando o acesso ao novo projeto. Para ter mais informações, consulte [Configuração de conexões de contas restritas ao projeto](#).

## Solução de problemas entre o Amazon CodeCatalyst e os SDKs da AWS ou AWS CLI

As informações a seguir podem ajudar você a solucionar problemas comuns ao trabalhar com o CodeCatalyst e os SDKs de AWS CLI ou da AWS.

### Tópicos

- [Eu recebo uma mensagem de erro quando entro em aws codecatalyst em uma linha de comando ou terminal dizendo que é uma escolha inválida](#)
- [Eu recebo um erro de credenciais quando executo comandos aws codecatalyst](#)

### Eu recebo uma mensagem de erro quando entro em aws codecatalyst em uma linha de comando ou terminal dizendo que é uma escolha inválida

Problema: quando tento usar o AWS CLI com o CodeCatalyst, um ou mais dos comandos aws codecatalyst não são reconhecidos como válidos.

Solução: a causa mais comum desse problema é que você está usando uma versão do AWS CLI que não contém as atualizações mais recentes dos últimos serviços e comandos. Atualize a instalação do AWS CLI e tente de novo. Para obter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

### Eu recebo um erro de credenciais quando executo comandos aws codecatalyst

Problema: quando tento usar o AWS CLI com o CodeCatalyst, recebo uma mensagem informando `You can configure credentials by running "aws configure". ou Unable to locate authorization token.`



Solução: você deve configurar um perfil do AWS CLI para trabalhar com os comandos do CodeCatalyst. Para obter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

## Noções básicas sobre o status atual do serviço com o relatório de integridade do CodeCatalyst

O relatório de integridade do Amazon CodeCatalyst é um painel público que fornece aos usuários uma lista agregada de notificações atualizadas sobre o desempenho dos recursos e a disponibilidade de serviços no CodeCatalyst que têm um impacto generalizado. Você pode ver quais recursos estão com problemas e podem afetar as aplicações no CodeCatalyst. Isso permite que você acompanhe interrupções e outros tempos de inatividade de recursos em todo o sistema. Quando ocorre um incidente, um indicador azul aparece no ícone do relatório de integridade. Além disso, o CodeCatalyst envia automaticamente um alerta e uma notificação por e-mail a todos os usuários com o perfil de Administrador do espaço no projeto, fornecendo detalhes e um histórico do incidente quase em tempo real.

O painel fornece uma lista de todos os eventos ativos e um registro de até 100 incidentes anteriores ocorridos nos últimos 30 dias. Você pode organizar a lista de incidentes com base na data em que o incidente foi atualizado. Você também pode atualizar a lista de incidentes para receber atualizações de última hora.

Confira um possível fluxo de trabalho para usar o relatório de integridade do CodeCatalyst:

Mateo Jackson é desenvolvedor no Budding Space com permissões de administrador do espaço. Ao tentar criar uma solicitação pull, ele continua recebendo uma mensagem de erro. Ele verifica o e-mail e descobre que recebeu um e-mail de incidente do sistema gerado automaticamente pelo CodeCatalyst, fornecendo um histórico detalhado sobre o problema do sistema que afeta o espaço. Ele seleciona Exibir atualização e é levado ao relatório de integridade do CodeCatalyst, onde pode visualizar todos os incidentes relatados pelo sistema. Ele escolhe o incidente na lista para ter mais informações. É aberta uma tela dividida que fornece um registro de data e hora de última atualização, histórico, recursos afetados, hora de início e status atual do incidente. Ele também pode ver que o problema está em andamento, mas a equipe de serviço começou a trabalhar nele. Sempre que há uma atualização no histórico ou no status do incidente, ele recebe um e-mail. Caso não tenha acesso ao e-mail, ele pode escolher o ícone de sino no painel superior para acessar o relatório de integridade do CodeCatalyst.

# Conceitos do relatório de integridade do CodeCatalyst

Aprender os conceitos a seguir ajudará você a entender o relatório de integridade do CodeCatalyst e como eles permitem que você acompanhe a integridade das aplicações, dos serviços e dos recursos.

## O incidente

Incidente é o evento do sistema que está afetando aplicações e recursos no CodeCatalyst. Você pode escolher o incidente para ver um histórico detalhado do evento, incluindo a hora em que ele começou e se a equipe de serviço está trabalhando para resolvê-lo.

## Status

O status é o status em tempo real do incidente. Ele será exibido como Em andamento ou Resolvido.

## Recursos afetados

Os recursos afetados são os recursos ou as aplicações afetados pelo incidente. Um único incidente pode afetar várias áreas do sistema, incluindo solicitações pull, problemas, fluxos de trabalho, teste, implantação e origem.

## Atualizado em

Atualizado em fornece um registro de data e hora da última atualização do incidente.

# Suporte para o Amazon CodeCatalyst

Ao criar um espaço no, você deve conectar uma Conta da AWS e designá-la como a conta de faturamento do seu espaço. A Conta da AWS que você designa como sua conta de faturamento também é onde você acessa seu plano do Suporte para o Amazon CodeCatalyst. Se precisar de suporte, crie casos de suporte a partir dessa Conta da AWS designada.

Os usuários do CodeCatalyst em um espaço usam o Suporte para a página do Amazon CodeCatalyst no CodeCatalyst para gerenciar casos de suporte. Você pode atualizar para um plano do Suporte, como suporte empresarial ou corporativo, para criar e gerenciar casos de suporte técnico do CodeCatalyst no CodeCatalyst. O suporte está disponível por telefone, Web ou chat para casos de suporte.

Somente casos específicos do serviço e dos recursos do CodeCatalyst podem receber suporte por meio do Suporte do Amazon CodeCatalyst. Os recursos do CodeCatalyst incluem recursos implantados no CodeCatalyst e por usuários no CodeCatalyst, mas não incluem recursos implantados para outros serviços da AWS ou de terceiros. Se precisar de suporte para qualquer outro serviço da AWS, você deve abri-lo por meio do Console de gerenciamento da AWS.

Para alterar o plano de suporte, consulte [Alteração dos planos de suporte](#).

## Note

Os planos do Suporte ao desenvolvedor não são projetados para ambientes de produção. Se uma conta de faturamento de espaço tiver um plano de Suporte ao desenvolvedor, esse plano não será distribuído a todos os administradores e membros do espaço no Suporte no CodeCatalyst.

# Cobrança do Suporte para o Amazon CodeCatalyst

Quando você cria um espaço no CodeCatalyst, os usuários no espaço podem criar e gerenciar casos de suporte do Suporte para o Amazon CodeCatalyst. Você pode criar dois tipos de casos de clientes:

- Casos de suporte de conta e faturamento estão disponíveis para todos os usuários do CodeCatalyst no espaço. Você pode receber ajuda com perguntas sobre faturamentos e contas com base em suas permissões no CodeCatalyst.

- Casos de suporte técnico conectam você a um engenheiro de suporte técnico para obter ajuda com problemas técnicos relacionados ao serviço e extensões para aplicações de terceiros. Se você tiver o suporte Básico, não poderá criar um caso de suporte técnico.

A Conta da AWS designada como faturamento para o espaço deve ter um plano Business Support ou Enterprise Support para o espaço a usar o Suporte para o CodeCatalyst em casos técnicos.

 Note

Se o seu espaço usar o Suporte para o Amazon CodeCatalyst a partir de uma conta que não tenha um plano Business Support ou Enterprise Support, você ainda poderá usar o Suporte para o Amazon CodeCatalyst para casos de conta e faturamento.

Para receber suporte técnico, você deve abrir todos os casos por meio do console do CodeCatalyst. Você não pode criar casos de suporte técnico para o CodeCatalyst a partir do [Suporte](#) no Console de gerenciamento da AWS.

 Note

As solicitações de aumento do limite de serviço não estão disponíveis do Suporte para o Amazon CodeCatalyst. Essas solicitações só podem ser enviadas pelo usuário-raiz da conta de faturamento do espaço no AWS Support Center Console.

Suporte para o Amazon CodeCatalyst tem os mesmos acordos de suporte que o Suporte, com as seguintes considerações:

- Listas de gravidade, tempos de resposta e SLAs no Suporte se aplicam a casos de suporte no Suporte para o CodeCatalyst, conforme detalhado em [Escolher uma gravidade](#).
- Administradores e membros do espaço não podem usar as APIs do Suporte, SDK da AWS ou a aplicação do Suporte no Slack para criar casos para o CodeCatalyst. Os casos de suporte do CodeCatalyst só podem ser enviados pelo CodeCatalyst.

 Note

O CodeCatalyst não está totalmente integrado com a Detecção e resposta a incidentes de AWS Trusted Advisor ou AWS. Valide como o CodeCatalyst é integrado para garantir que suas práticas comerciais estejam alinhadas com a integração atual.

Você deve ser um usuário no espaço onde deseja solicitar suporte.

 Note

Se tiver mais de um construtor em seu espaço, recomendamos comprar um plano Business Support ou Enterprise Support. Esses planos fornecem suporte técnico para o espaço para até 5.000 construtores.

A Conta da AWS designada como conta de faturamento do espaço usa o perfil de `AWSRoleForCodeCatalystSupport` e a política gerenciada [AmazonCodeCatalystSupportAccess](#). Isso permite que os usuários do CodeCatalyst em um espaço acessem o Suporte para a página do Amazon CodeCatalyst. Para ter mais informações sobre este perfil e política, consulte [AmazonCodeCatalystSupportAccess](#). Para outras considerações sobre faturamento, consulte [Managing billing](#) no Guia do administrador do Amazon CodeCatalyst.

Veja a seguir um fluxo possível para um construtor criar um caso de suporte no CodeCatalyst:

Mateo Jackson é desenvolvedor de um projeto no CodeCatalyst. Depois de assinar a Conta da AWS que gerencia o faturamento com o Suporte no Amazon CodeCatalyst e atualizar para um plano Business Support, todos os criadores no espaço podem criar casos de suporte técnico. Mateo envia um caso de suporte técnico para uma falha no fluxo de trabalho em seu projeto. Mateo usa o Suporte para a página do Amazon CodeCatalyst para preencher o formulário e criar um caso, fornecendo o ID do fluxo de trabalho e outros detalhes na solicitação. O caso é criado com um ID de caso e inclui o ID da conta da Conta da AWS designada como conta de faturamento e associada ao plano de suporte do espaço.

Embora todos os criadores possam criar casos de suporte no Suporte para o CodeCatalyst, você não é cobrado por cada caso criado. Você pode abrir casos e contatos praticamente ilimitados com base no plano Suporte Premium adquirido em sua conta de faturamento de espaço.

 Note

A conta de faturamento de espaço é a Conta da AWS cobrada pelos usuários e recursos do CodeCatalyst. Se você tiver implantado em Contas da AWS adicionais, entre em contato com o Suporte por meio do Console de gerenciamento da AWS para receber assistência com recursos implantados em outros serviços.

Você pode identificar a Conta da AWS implantada a partir do fluxo de trabalho.

## Configuração do espaço para o Suporte para o Amazon CodeCatalyst

Suporte para o Amazon CodeCatalyst gerencia casos de suporte como parte da integração da API do Suporte com o CodeCatalyst.

O perfil de `AWSRoleForCodeCatalystSupport` é um perfil de serviço usado para casos de suporte em seu espaço. O perfil deve ser adicionado à conta de faturamento designada para o espaço. Para ter mais informações ou para criar o perfil, consulte [Criação do perfil `AWSRoleForCodeCatalystSupport` para a conta e o espaço](#).

 Note

Para um espaço criado antes de 20 de abril de 2023, você deve criar o perfil para que o suporte ao CodeCatalyst funcione em seu espaço. Se você criar um espaço após 20 de abril de 2023, poderá criar o perfil durante a criação do espaço, na página de detalhes de faturamento no CodeCatalyst ou clicando no link do banner de suporte no CodeCatalyst.

Para configurar o suporte para seu espaço

1. Ao criar um espaço do CodeCatalyst, você é instruído a conectar uma conta de faturamento. A conta de faturamento designada para o espaço será cobrada pela AWS. Para ter mais informações sobre como criar um espaço, consulte [Criar um espaço e perfil de desenvolvimento \(começando sem um convite\)](#).
2. Quando você cria um espaço do CodeCatalyst, a opção está disponível para criar o perfil de serviço `AWSRoleForCodeCatalystSupport` que permite que os usuários do CodeCatalyst acessem o suporte. Esse perfil usa a política gerenciada de

AmazonCodeCatalystSupportAccess. O perfil deve ser adicionado à Conta da AWS designada para o espaço. Para obter mais informações sobre a criação do perfil, consulte [Criação do perfil AWSRoleForCodeCatalystSupport para a conta e o espaço](#).

3. Para a conta de faturamento designada para o espaço, recomendamos que o administrador do espaço adquira um plano Business Support ou Enterprise Support para a Conta da AWS. Todos os membros do espaço poderão gerenciar casos de suporte do Suporte para o Amazon CodeCatalyst, e os canais de suporte estarão alinhados ao plano do Suporte adquirido quando as integrações forem concluídas.
4. Para criar e gerenciar casos de suporte no CodeCatalyst, consulte [Criação de um caso de suporte do CodeCatalyst no CodeCatalyst](#).

## Acesso ao suporte para o CodeCatalyst no Console de gerenciamento da AWS

Se a conta de faturamento habilitada para suporte de um espaço for desconectada, os casos do Suporte associados à conta de faturamento do espaço anterior e ao plano de suporte associado não estarão mais visíveis no Suporte para o Amazon CodeCatalyst. O usuário-raiz dessa conta de faturamento pode visualizar e resolver casos antigos a partir do Console de gerenciamento da AWS e pode configurar permissões do IAM para o Suporte para que outros usuários visualizem e resolvam casos antigos. Você ainda poderá aproveitar os benefícios do seu plano de suporte do Console de gerenciamento da AWS para todos os outros Serviços da AWS e concluir quaisquer casos de suporte do CodeCatalyst que não tenham sido resolvidos anteriormente.

Para ter mais informações, consulte [Updating, resolving, and reopening your case](#) no Guia do usuário do Suporte.

Os casos de suporte para informações gerais sobre o CodeCatalyst também podem ser abertos no Console de gerenciamento da AWS, mas nenhum suporte técnico pode ser recebido por meio desse canal para o CodeCatalyst. Para ter mais informações, consulte [Creating support cases and case management](#) no Guia do usuário do Suporte.

Veja a seguir um fluxo possível para um usuário resolver um caso de suporte para o CodeCatalyst no Console de gerenciamento da AWS:

Embora todos os criadores possam criar casos de suporte com o Suporte para o Amazon CodeCatalyst, as solicitações de suporte são cobradas da conta designada como a conta de faturamento do espaço. Mateo Jackson é desenvolvedor de um projeto no CodeCatalyst e abriu um



caso de suporte técnico para uma falha no fluxo de trabalho em seu projeto. No entanto, a conta de faturamento do espaço que foi cadastrada com o Suporte para o Amazon CodeCatalyst e comprou um plano Business Support foi desconectada do espaço. A única maneira de Mateo visualizar a comunicação mais recente e resolver os casos abertos para o CodeCatalyst é gerenciar o ID do caso a partir do Centro do Suporte no Console de gerenciamento da AWS. Para fazer isso, Mateo recebe permissões de IAM do usuário-raiz da conta de faturamento do espaço anterior anexada ao caso de suporte e resolve o caso por meio do Suporte no console.

#### Important

Se você alterar a conta de faturamento designada para seu espaço, seu plano do Suporte ainda estará acessível até o final do mês por meio do único Console de gerenciamento da AWS. Você precisará recomprar o Suporte na conta de faturamento atualizada para continuar acessando seus casos de suporte criados anteriormente no CodeCatalyst. Recomendamos aguardar até que você tenha resolvido todos os seus casos de suporte para alterar as contas de faturamento do espaço, a fim de evitar qualquer impacto no acesso aos seus casos de suporte por meio do Suporte para o Amazon CodeCatalyst.

## Criação de um caso de suporte do CodeCatalyst no CodeCatalyst

É possível abrir um caso de suporte na página Suporte do Amazon CodeCatalyst.

Um ID do builder AWS só pode receber suporte para o alias com o qual está autenticado e somente para recursos com base nas permissões. As opções de conta e de faturamento estão disponíveis para todos os administradores e membros do espaço. No entanto, os usuários só podem receber suporte para os recursos aos quais têm acesso no CodeCatalyst e não em relação ao gerenciamento do faturamento da conta.

Você pode criar um caso de conta e faturamento ou um caso de suporte técnico para seus recursos do CodeCatalyst usando a página Suporte do CodeCatalyst do seu espaço.

#### Note

Somente casos específicos do serviço e dos recursos do CodeCatalyst podem receber suporte por meio do Suporte do Amazon CodeCatalyst. Os recursos do CodeCatalyst incluem recursos implantados no CodeCatalyst e por usuários no CodeCatalyst, mas não incluem recursos implantados para outros serviços da AWS ou de terceiros. Se precisar

de suporte para qualquer outro serviço da AWS, você deve abri-lo por meio do Console de gerenciamento da AWS.

Para criar um caso de suporte no CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Acesse o espaço no CodeCatalyst.

 **Tip**

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Na parte superior da página, selecione o ícone ? e, depois, selecione Suporte.
4. Escolha Criar caso.
5. Escolha uma das seguintes opções:
  - Conta e faturamento
  - Técnico

 **Note**

No Suporte do Amazon CodeCatalyst, se um plano Business Support ou Enterprise Support for adicionado à conta de faturamento do espaço, o suporte técnico de casos do CodeCatalyst estará disponível para todos os administradores e membros do espaço. Para obter informações sobre a solução de problemas, consulte [Não consigo criar casos de suporte técnico para meu espaço](#).

Os planos de Suporte não se estendem por espaços. Se você for membro de vários espaços, o administrador do espaço precisará comprar um plano de Suporte Premium para cada espaço para receber suporte técnico em todos os espaços.

6. Selecione o Service (Serviço), a Category (Categoria) e a Severity (Severidade). Para ter informações sobre como escolher uma gravidade, consulte [Escolha de uma gravidade](#).
  - Orientações gerais
  - Sistema prejudicado

- Sistema de produção prejudicado
  - Sistema de produção desativado
  - Sistema crítico para os negócios inativo
7. Selecione Próxima etapa: informações adicionais.
  8. Na página Additional information (Informações adicionais), em Subject (Assunto), insira um título sobre o problema.
  9. Em Description (Descrição), siga as instruções para descrever seu caso, como segue:
    - Informações de solução de problemas específicas do CodeCatalyst, como ID do fluxo de trabalho, logs ou capturas de tela
    - Mensagens de erro recebidas
    - Etapas de resolução de problemas que você seguiu

 Note

Não compartilhe informações confidenciais em correspondências de casos, como credenciais, dados de cartão de crédito, URLs assinados ou informações de identificação pessoal.

10. (Opcional) Escolha Attach files (Anexar arquivos) para adicionar arquivos relevantes ao seu caso, como registros de erros ou capturas de tela. É possível anexar até três arquivos. Cada arquivo pode ter até 5 MB.
11. Em Nome do espaço, o nome do seu espaço é exibido.
12. Em Nome do construtor, o nome completo associado ao ID do builder AWS é preenchido automaticamente.
13. (Opcional) Escolha o projeto em Nome do projeto (se aplicável).

 Note

Você só verá os projetos para os quais tem permissão. Se você precisar acessar outro projeto, peça ao administrador que lhe forneça acesso antes de criar um caso de suporte.

14. Selecione Próxima etapa: entre em contato conosco.

15. Em Idioma de contato preferencial, escolha o padrão. Somente Inglês está disponível no momento.
16. Escolha a opção Web, Telefone ou Chat como método de contato.
17. Revise os detalhes do caso e escolha Enviar. O número do ID do caso e o resumo são exibidos.

O caso de suporte é criado no nível do espaço e pode ser visualizado por todos os membros com acesso ao espaço e ao projeto (se selecionados) definidos no caso de suporte. Não há como omitir um caso de suporte de usuários individuais no momento.

## Resolução de um caso de suporte no CodeCatalyst

Você pode resolver casos de suporte abertos de Suporte para a página Amazon CodeCatalyst.

Você deve ter um perfil de administrador de espaço ou membro de espaço no espaço onde deseja resolver um caso de suporte. Se você não tiver o perfil de administrador de espaço ou se um projeto foi selecionado quando o caso foi criado, você também precisará ser membro do projeto para visualizar e resolver o caso.

Para resolver um caso de suporte aberto no CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Acesse o espaço no CodeCatalyst.

### Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Na parte superior da página, selecione o ícone ? e, depois, selecione Suporte para Amazon CodeCatalyst.
4. Selecione o link para o caso de suporte que você deseja gerenciar. Escolha Resolve case (Resolver caso).

## Reabertura de um caso de suporte no CodeCatalyst

Você pode usar a reabertura de um caso de suporte resolvido de Suporte para a página Amazon CodeCatalyst.

 Note

É possível reabrir seu caso de suporte até 14 dias a partir do momento em que o problema foi resolvido. No entanto, não é possível reabrir um caso que esteja inativo há mais de 14 dias. Se você não conseguir reabrir seu caso, abra um novo caso e inclua o ID do caso anterior como referência.

Se você reabrir um caso existente que tenha informações diferentes do problema atual, o atendente do suporte poderá solicitar que você crie um novo caso.

Para reabrir um caso de suporte no CodeCatalyst

1. Abra o console do CodeCatalyst em <https://codecatalyst.aws/>.
2. Acesse o espaço no CodeCatalyst.

 Tip

Se você pertencer a mais de um espaço, selecione um espaço na barra de navegação superior.

3. Na parte superior da página, selecione o ícone ? e, depois, selecione AWS Support para CodeCatalyst.
4. Selecione o link para o caso de suporte que você deseja gerenciar. Selecione Reopen (Reabrir). Selecione OK na tela de confirmação e, depois, selecione Enviar.
5. Preencha a Descrição com as informações mais recentes sobre o mesmo problema. Não compartilhe informações confidenciais em correspondências de casos, como credenciais, dados de cartão de crédito, URLs assinados ou informações de identificação pessoal.

# Cotas para o CodeCatalyst

As tabelas a seguir descrevem as cotas e os limites no Amazon CodeCatalyst. É possível encontrar informações adicionais sobre aspectos específicos do CodeCatalyst nos seguintes tópicos:

- [Cotas para repositórios de origem em CodeCatalyst](#)
- [Cotas para identidade, permissão e acesso em CodeCatalyst](#)
- [Cotas para fluxos de trabalho em CodeCatalyst](#)
- [Cotas para ambientes de desenvolvimento em CodeCatalyst](#)
- [Cotas para projetos](#)
- [Cotas para plantas em CodeCatalyst](#)
- [Cotas para espaços](#)
- [Cotas para edições em CodeCatalyst](#)

|                                                                                                                                 |       |
|---------------------------------------------------------------------------------------------------------------------------------|-------|
| Número máximo de espaço em uma conta                                                                                            | 5     |
| Número máximo de espaços que um usuário pode criar em um mês civil                                                              | 5     |
| Número mínimo de Contas da AWS para um espaço                                                                                   | 1     |
| Número máximo de conexões de conta para um espaço                                                                               | 5.000 |
| Número máximo de Contas da AWS como conta de faturamento de um espaço                                                           | 1     |
| Número máximo de espaços no nível gratuito para as quais uma única Conta da AWS pode ser especificada como conta de faturamento | 3     |
| Número máximo de espaços no nível pago para as quais uma única Conta da AWS pode ser especificada como conta de faturamento     | 15    |

|                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Número máximo de conexões VPC para um espaço                  | 100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Número máximo de projetos em um espaço                        | 100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Número máximo de projetos aos quais um usuário pode pertencer | 1.000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Descrições do espaço                                          | <p>As descrições do espaço são opcionais. Se especificado, elas devem ter entre 0 e 200 caracteres de comprimento. Elas podem conter qualquer combinação de letras, números, espaços, pontos, sublinhados, vírgulas, traços e os caracteres especiais a seguir:</p> <p>? &amp; \$ % + = / \ ; : \n \t \r</p>                                                                                                                                                                                     |
| Nomes de espaço                                               | <p>Os nomes de espaço devem ser exclusivos no CodeCatalyst. Você não pode reutilizar nomes de espaços excluídos.</p> <p>Os nomes de espaço deve ter entre 3 e 63 caracteres de comprimento. Eles também devem começar com um caractere alfanumérico. Os nomes dos espaços podem conter qualquer combinação de letras, números, pontos, sublinhados e traços. Eles não podem conter nenhum dos seguinte caracteres:</p> <p>! ? @ # \$ % ^ &amp; * ( ) + = { } [ ]   / \ &gt; &lt; ~ ` ' " ; :</p> |

# Histórico de documentos da Amazon CodeCatalyst

A tabela a seguir descreve o histórico da documentação e as atualizações da documentação geral do CodeCatalyst.

| Alteração                                                                        | Descrição                                                                                                                                                                                          | Data                   |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">A Amazon não CodeCatalyst está mais aberta a novos clientes</a>      | A Amazon não CodeCatalyst está mais aberta a novos clientes. Para obter mais informações, consulte <a href="#">Como migrar da Amazon CodeCatalyst</a> .                                            | 7 de novembro de 2025  |
| <a href="#">Novo conteúdo: nova imagem universal do devfile</a>                  | Adição da documentação da nova imagem universal do devfile (Universal image 5.0).                                                                                                                  | 30 de julho de 2025    |
| <a href="#">Conteúdo atualizado: privacidade de dados</a>                        | A documentação foi atualizada para incluir informações adicionais sobre a privacidade de dados relacionada ao IAM Identity Center. Consulte <a href="#">Privacidade de dados em CodeCatalyst</a> . | 28 de julho de 2025    |
| <a href="#">Conteúdo atualizado: suspensão de uso do CodeCatalyst blueprints</a> | O conteúdo AWS do Project Development Kit (AWS PDK) foi removido da tabela de <a href="#">esquemas disponíveis</a> .                                                                               | 06 de dezembro de 2024 |
| <a href="#">Novo conteúdo: nova imagem universal do devfile</a>                  | Adição da documentação da nova imagem universal do devfile (Universal image 4.0).                                                                                                                  | 26 de agosto de 2024   |



|                                                                                                                                                                 |                                                                                                                                                                                                                                                  |                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <a href="#">Novo conteúdo: Imagem de tempo de execução usada pela ação 'AWS CDK implantar', Imagem de tempo de execução usada pela ação 'AWS CDK bootstrap'</a> | Adição de dois tópicos: <a href="#">Imagem de tempo de execução usada pela ação 'AWS CDK implantar'</a> e <a href="#">Imagem de tempo de execução usada pela ação 'AWS CDK bootstrap'</a> .                                                      | 20 de agosto de 2024 |
| <a href="#">Novo conteúdo: imagens de devfile universais obsoletas</a>                                                                                          | Adição da documentação de imagens de devfile universais obsoletas (Universal image 1.0 e Universal image 2.0).                                                                                                                                   | 16 de agosto de 2024 |
| <a href="#">Conteúdo atualizado: suspensão de uso do CodeCatalyst blueprints</a>                                                                                | O modelo de aplicativo sem servidor (SAM) e o conteúdo do serviço Video-on-demand web foram removidos da tabela de <a href="#">blueprints disponíveis</a> .                                                                                      | 14 de agosto de 2024 |
| <a href="#">Conteúdo atualizado: Imagens ativas</a>                                                                                                             | O tópico <a href="#">Imagens ativas</a> foi atualizado para indicar que determinadas ações podem usar uma imagem de novembro de 2022 ou março de 2024. Também houve a atualização da documentação de cada ação para indicar qual imagem é usada. | 9 de agosto de 2024  |

|                                                                                                                                                             |                                                                                                                                                                                                                                                                            |                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: solução de problemas com nome de usuário truncado</a>                                                                      | Os tópicos de solução de problemas foram atualizados para incluir informações sobre a solução de problemas de nomes de usuário que podem estar truncados em determinados casos.                                                                                            | 31 de julho de 2024 |
| <a href="#">Conteúdo atualizado: Criar um ambiente</a>                                                                                                      | A <a href="#">Criar um ambiente</a> seção foi atualizada para mencionar CodeCatalyst funções.                                                                                                                                                                              | 25 de julho de 2024 |
| <a href="#">Conteúdo atualizado: Compartilhar artefatos e arquivos entre ações</a>                                                                          | O subtópico <a href="#">Referência de arquivos em um artefato</a> foi atualizado para incluir informações sobre grupos de ação. Também foi adicionado o subtópico <a href="#">Exemplo: referência a um arquivo em um artefato quando um grupo de ações está presente</a> . | 18 de julho de 2024 |
| <a href="#">Conteúdo atualizado: Compilação, teste e implantação com fluxos de trabalho</a>                                                                 | A documentação foi atualizada para indicar que a extensão do arquivo <code>.yaml</code> ou <code>.yml</code> da definição do fluxo de trabalho deve estar em minúsculas.                                                                                                   | 15 de julho de 2024 |
| <a href="#">Conteúdo atualizado: solução de problemas adicionada para usuários de SSO ao criar um ambiente de desenvolvimento em um espaço CodeCatalyst</a> | Adição de um tópico de solução de problemas para incluir informações sobre problemas encontrados por usuários de SSO conectados em um espaço com federação de identidades ao criar ambientes de desenvolvimento.                                                           | 12 de julho de 2024 |

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                |                     |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: Criação de um fluxo de trabalho</a>              | O tópico foi atualizado para indicar que o arquivo de definição do fluxo de trabalho pode ser armazenado em um subdiretório de <code>~/codecatalyst/workflows/</code> .                                                                                                                                                                                        | 12 de julho de 2024 |
| <a href="#">Conteúdo atualizado: Estados do fluxo de trabalho em CodeCatalyst</a> | Foram adicionadas instruções sobre como ocultar fluxos de trabalho inativos no CodeCatalyst console.                                                                                                                                                                                                                                                           | 12 de julho de 2024 |
| <a href="#">Novo conteúdo: Configurar gatilhos somente manuais</a>                | Adição de um tópico <a href="#">Configurar gatilhos somente manuais</a> .                                                                                                                                                                                                                                                                                      | 26 de junho de 2024 |
| <a href="#">Conteúdo atualizado: Criar um ambiente</a>                            | A seção <a href="#">Criar um ambiente</a> e várias outras foram atualizadas para refletir um novo recurso que permita especificar um perfil do IAM padrão ao criar um ambiente de CI/CD. Esse perfil será atribuído a todas as ações do fluxo de trabalho que estão associadas ao ambiente. Você não precisa mais atribuir perfis do IAM diretamente às ações. | 21 de junho de 2024 |
| <a href="#">Novo conteúdo: Tutorial: extrair de um repositório de pacotes</a>     | Foi adicionado um tutorial que descreve como configurar um fluxo de trabalho para extrair dependências de um repositório de CodeCatalyst pacotes.                                                                                                                                                                                                              | 20 de junho de 2024 |

[Conteúdo atualizado: use repositórios de GitLab projetos com plantas](#)

Documentação atualizada para a capacidade de criar um repositório de GitLab projetos ao [Criar um projeto com um esquema](#) [criar um blueprint personalizado](#). [Adicionar um esquema a um projeto para integrar recursos](#)

19 de junho de 2024

[Conteúdo atualizado: Tutorial: Usar recursos de IA generativa](#)

O tutorial foi atualizado para refletir a capacidade de usar o Amazon Q para escolher um esquema ao criar um projeto ou adicionar um esquema a um projeto existente.

18 de junho de 2024

[Conteúdo atualizado: criação de um repositório de origem](#)

A documentação foi atualizada para incluir informações sobre como criar repositórios vazios.

18 de junho de 2024

[Novo conteúdo: Criando um projeto CodeCatalyst com o Amazon Q](#)

Foram adicionadas novas seções a [Como criar um projeto](#) chamadas [Práticas recomendadas ao usar o Amazon Q para criar projetos ou adicionar funcionalidades com esquemas](#) e [Práticas recomendadas para usar esquemas em projetos](#) com instruções para criar projetos e adicionar esquemas com o Amazon Q.

18 de junho de 2024

|                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                     |
|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Novo conteúdo: clonar um repositório Git existente em um repositório de origem</a> | Foi adicionado um novo tópico sobre como usar o Git para enviar um clone espelho ou um repositório local para um repositório de origem vazio em. CodeCatalyst                                                                                                                                                                                                                                                                                         | 18 de junho de 2024 |
| <a href="#">Novo conteúdo: Support para os tipos de pacotes Maven NuGet,, e Python</a>         | Foi adicionada documentação para usar os pacotes Maven NuGet,, e Python no. CodeCatalyst                                                                                                                                                                                                                                                                                                                                                              | 17 de junho de 2024 |
| <a href="#">Atualizações nos espaços e nas conexões da conta</a>                               | Agora você pode conectar o Conta da AWS especificado como conta de cobrança a vários CodeCatalyst espaços. Para espaços configurados para federação de identidades, uma instância do Identity Center pode ser conectada a vários espaços. Consulte <a href="#">Permitindo acesso a AWS recursos com conexão Contas da AWS</a> , <a href="#">Solução de problemas de contas associadas ao seu espaço</a> e <a href="#">Cotas para o CodeCatalyst</a> . | 13 de junho de 2024 |
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                       | Atualização da documentação dos perfis para incluir permissões para usar o Amazon Q para recomendar e criar tarefas para um problema.                                                                                                                                                                                                                                                                                                                 | 13 de junho de 2024 |

|                                                                                  |                                                                                                                                                                                                                                                                |                     |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                         | A documentação dos perfis foi atualizada para incluir permissões para criar, atualizar e remover links entre problemas.                                                                                                                                        | 13 de junho de 2024 |
| <a href="#">Conteúdo atualizado: Tutorial: Usar recursos de IA generativa</a>    | O tutorial foi atualizado para incluir informações sobre o uso do Amazon Q para recomendar tarefas para um problema.                                                                                                                                           | 13 de junho de 2024 |
| <a href="#">Conteúdo atualizado: gerenciamento de tarefas sobre problemas</a>    | Foram adicionadas informações sobre o uso do Amazon Q para recomendar tarefas para um problema em CodeCatalyst.                                                                                                                                                | 13 de junho de 2024 |
| <a href="#">Novo conteúdo: vincular um problema a outro</a>                      | Foi adicionado um novo tópico sobre como vincular problemas a outros problemas em CodeCatalyst.                                                                                                                                                                | 13 de junho de 2024 |
| <a href="#">Conteúdo atualizado: YAML da ação “Implantação do AWS CDK ”</a>      | Correção da descrição da propriedade Region.                                                                                                                                                                                                                   | 12 de junho de 2024 |
| <a href="#">Conteúdo atualizado: usar repositórios de terceiros com esquemas</a> | Documentação atualizada para a capacidade de criar um GitHub repositório quando <a href="#">Criar um projeto com um esquema</a> Adicionar um <a href="#">esquema a um projeto para integrar recursos</a> , ou <a href="#">criar um esquema personalizado</a> . | 6 de junho de 2024  |

|                                                                                        |                                                                                                                                                                                                                     |                    |
|----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <a href="#">Novo conteúdo: etapas adicionadas para trabalhar com conexões pessoais</a> | Foram adicionadas etapas para criar e excluir conexões pessoais. As conexões pessoais permitem que você gerencie GitHub recursos para projetos e plantas na Amazon CodeCatalyst.                                    | 6 de junho de 2024 |
| <a href="#">Novo conteúdo: extensão de repositórios do Bitbucket</a>                   | Novo conteúdo adicionado para usar a extensão de repositórios Bitbucket em. CodeCatalyst                                                                                                                            | 5 de junho de 2024 |
| <a href="#">Novo conteúdo: Tipos de ação</a>                                           | O <a href="#">Ações de terceiros</a> tópico foi atualizado para mencionar a ação SonarCloud Escanear.                                                                                                               | 29 de maio de 2024 |
| <a href="#">Conteúdo atualizado: YAML da ação “Implantação do AWS CDK ”</a>            | Correção do exemplo CdkRootRootPath .                                                                                                                                                                               | 28 de maio de 2024 |
| <a href="#">Conteúdo atualizado</a>                                                    | Títulos de tópicos atualizados e conteúdo reorganizado para melhorar a legibilidade e facilitar a descoberta. Se você quiser fornecer feedback sobre essas mudanças, use o <a href="#">link Fornecer feedback</a> . | 17 de maio de 2024 |
| <a href="#">Novo conteúdo: visualizar o histórico de alterações em um arquivo</a>      | A documentação foi atualizada para refletir a nova funcionalidade de visualização do histórico de alterações em um arquivo em um repositório de origem.                                                             | 1º de maio de 2024 |

|                                                                                         |                                                                                                                                                                                                                        |                     |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: Tutorial: Usar recursos de IA generativa</a>           | O tutorial foi atualizado para refletir a integração com o Amazon Q Developer.                                                                                                                                         | 29 de abril de 2024 |
| <a href="#">Conteúdo atualizado: Tutorial: Usar recursos de IA generativa</a>           | O tutorial foi atualizado para refletir a possibilidade de o Amazon Q analisar problemas em busca de complexidade, sugerir e criar tarefas e trabalhar em tarefas em um problema.                                      | 22 de abril de 2024 |
| <a href="#">Novo conteúdo: Bloquear uma execução de fluxo de trabalho</a>               | Adição de <a href="#">Bloquear uma execução de fluxo de trabalho</a> , <a href="#">Solicitar aprovações em execuções de fluxo de trabalho</a> e vários outros tópicos relacionados às aprovações do fluxo de trabalho. | 22 de abril de 2024 |
| <a href="#">Novo conteúdo: usar tarefas para dividir problemas em objetivos menores</a> | Conteúdo adicionado para apoiar o lançamento de tarefas em problemas. As tarefas podem ser adicionadas aos problemas para detalhar, organizar e acompanhar ainda mais o trabalho desse problema.                       | 4 de abril de 2024  |



[Conteúdo atualizado: Compartilhar artefatos e arquivos entre ações](#)

O tópico [Compartilhar artefatos e arquivos entre ações](#) foi atualizado para incluir dois novos subtópicos: [Posso compartilhar artefatos sem especificá-los como saídas e entradas?](#) e [Posso compartilhar artefatos entre fluxos de trabalho?](#)

2 de abril de 2024

[Conteúdo atualizado: GitHub Limitações das ações em CodeCatalyst](#)

O [GitHub Limitações das ações em CodeCatalyst](#) tópico foi atualizado para indicar que GitHub as ações são executadas em uma imagem Docker de um ambiente de tempo de execução mais antigo.

2 de abril de 2024

[Novo conteúdo: YAML da ação “Implantação do AWS CDK ”](#)

Adição de uma nova CloudAssemblyRootPath propriedade a [YAML da ação “Implantação do AWS CDK ”](#).

1º de abril de 2024

[Conteúdo atualizado: Especificação de imagens de ambiente de runtime](#)

O tópico [Especificação de imagens de ambiente de runtime](#) foi atualizado para incluir informações sobre a nova imagem do ambiente de runtime de março de 2024.

26 de março de 2024

|                                                                                          |                                                                                                                                                                                                                                                                          |                     |
|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                 | Informações consolidadas de permissão de perfil em uma única tabela. A tabela está em um novo tópico <a href="#">Visualização das permissões disponíveis para cada perfil</a> .                                                                                          | 18 de março de 2024 |
| <a href="#">Novo conteúdo: Visualização de todos os espaços e projetos de um usuário</a> | Foram adicionadas informações sobre a visualização de um anúncio na página inicial do usuário que mostra cada CodeCatalyst espaço ou projeto para o usuário conectado. CodeCatalyst Consulte <a href="#">Visualização de todos os espaços e projetos de um usuário</a> . | 18 de março de 2024 |
| <a href="#">Novo conteúdo: Exemplo: um gatilho com uma extração e ramificações</a>       | Foi adicionado um exemplo de gatilho de solicitação pull. Pequenas correções no tópico <a href="#">Início da execução automática de um fluxo de trabalho usando gatilhos</a> .                                                                                           | 11 de março de 2024 |
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                 | A documentação dos perfis foi atualizada para incluir permissões para criar, excluir e visualizar ambientes.                                                                                                                                                             | 4 de março de 2024  |
| <a href="#">Conteúdo atualizado: Tutorial: Usar recursos de IA generativa</a>            | O tutorial foi atualizado para refletir as mudanças ao criar e atribuir problemas ao Amazon Q.                                                                                                                                                                           | 4 de março de 2024  |

|                                                                     |                                                                                                                                                                                                                                                         |                         |
|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <a href="#">Novo conteúdo: componentes de problemas</a>             | Adição de um novo conteúdo sobre como trabalhar com componentes de problema como desenvolvedor de esquemas personalizados.                                                                                                                              | 27 de fevereiro de 2024 |
| <a href="#">Conteúdo atualizado: Tipos de ação</a>                  | O <a href="#">CodeCatalyst Ações do Labs</a> tópico foi atualizado para incluir uma lista de ações do CodeCatalyst Labs.                                                                                                                                | 21 de fevereiro de 2024 |
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a> | A documentação foi atualizada para refletir a nova funcionalidade com regras de aprovação e requisitos de substituição para mesclar uma solicitação pull.                                                                                               | 15 de fevereiro de 2024 |
| <a href="#">Conteúdo atualizado: mesclagem de solicitações pull</a> | Adição da documentação para solicitações pull para incluir informações sobre a substituição dos requisitos de mesclagem para mesclar uma solicitação pull que ainda não recebeu aprovações dos revisores necessários ou atendeu às regras de aprovação. | 15 de fevereiro de 2024 |
| <a href="#">Novo conteúdo: gerenciar regras de aprovação</a>        | Foi adicionada documentação para solicitações pull de forma a incluir informações sobre a criação e o gerenciamento de regras de aprovação.                                                                                                             | 15 de fevereiro de 2024 |

|                                                                                                                              |                                                                                                                                                                                                                        |                         |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                                                     | A documentação dos perfis foi atualizada para incluir permissões para trabalhar com regras de aprovação e solicitações pull.                                                                                           | 14 de fevereiro de 2024 |
| <a href="#">Conteúdo atualizado: Como faço para corrigir os erros “A definição do fluxo de trabalho tem <i>n</i> erros”?</a> | A seção <a href="#">Como faço para corrigir os erros “A definição do fluxo de trabalho tem <i>n</i> erros”?</a> foi atualizada para incluir mais dicas de solução de problemas.                                        | 9 de fevereiro de 2024  |
| <a href="#">Novo conteúdo: Visualização do status de um fluxo de trabalho</a>                                                | Foi adicionada uma seção que descreve os estados do fluxo de trabalho.                                                                                                                                                 | 9 de fevereiro de 2024  |
| <a href="#">Novo conteúdo: Visualização do status de um fluxo de trabalho</a>                                                | Foi adicionada uma seção que descreve os estados do fluxo de trabalho.                                                                                                                                                 | 9 de fevereiro de 2024  |
| <a href="#">Atualização de conteúdo: Cotas para fluxos de trabalho em CodeCatalyst</a>                                       | O tópico <a href="#">Cotas para fluxos de trabalho em CodeCatalyst</a> foi atualizado com as cotas Número máximo de ações por fluxo de trabalho e Número máximo de ambientes associados a uma Conta da AWS por espaço. | 7 de fevereiro de 2024  |
| <a href="#">Conteúdo atualizado: Criar um ambiente</a>                                                                       | A seção <a href="#">Criar um ambiente</a> foi atualizada para indicar que você pode usar no máximo uma conexão de conta por ambiente.                                                                                  | 31 de janeiro de 2024   |

|                                                                                                                       |                                                                                                                                             |                        |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Novo conteúdo: repositório de plantas GitHub personalizadas</a>                                           | Novo conteúdo adicionado ao GitHub repositório que é disponibilizado publicamente.                                                          | 10 de janeiro de 2024  |
| <a href="#">Conteúdo atualizado: Configurando o npm com CodeCatalyst</a>                                              | Instruções gerais de configuração atualizadas para usar o npm com CodeCatalyst e maior clareza sobre a <code>always-auth=true</code> opção. | 5 de janeiro de 2024   |
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a>                                                   | A documentação foi atualizada para refletir a nova funcionalidade com os recursos generativos de IA em CodeCatalyst.                        | 28 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: criação de um problema</a>                                                           | A documentação foi atualizada para refletir a nova funcionalidade com os recursos generativos de IA em CodeCatalyst.                        | 28 de novembro de 2023 |
| <a href="#">Novo conteúdo: Tutorial: Usar recursos de IA generativa</a>                                               | Foi adicionado um tutorial para usar os recursos generativos de IA na Amazon CodeCatalyst.                                                  | 28 de novembro de 2023 |
| <a href="#">Novo conteúdo: esquemas personalizados e gerenciamento do ciclo de vida</a>                               | Foi adicionado novo conteúdo para usar o plano personalizado e os recursos de gerenciamento do ciclo de vida na Amazon. CodeCatalyst        | 27 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: Tutorial: Criar um projeto com o esquema de aplicação web moderna de três níveis</a> | Atualização do tutorial com correções e informações sobre solução de problemas.                                                             | 22 de novembro de 2023 |

[Conteúdo atualizado: Início da execução automática de um fluxo de trabalho usando gatilhos](#)

Alguns exemplos e descrições relacionados aos gatilhos de solicitação pull foram corrigidos. Adição de uma seção [Diretrizes para o uso de gatilhos e ramificações](#).

22 de novembro de 2023

[Novo conteúdo: Fazer login com SSO](#)

Foram adicionadas informações sobre como entrar com o Single Sign-On (SSO) e links para informações sobre como configurar e gerenciar um CodeCatalyst espaço que ofereça suporte à federação de identidades. Consulte [Configuração e login no CodeCatalyst](#) e [Fazer login com SSO](#).

17 de novembro de 2023

[Conteúdo atualizado: trabalho com perfis](#)

A documentação dos perfis foi atualizada para incluir permissões para trabalhar com equipes, conexões VPC, login único e recursos de máquina.

16 de novembro de 2023

[Conteúdo atualizado: trabalho com solicitações pull](#)

A documentação foi atualizada para refletir as alterações na forma como as alterações de uma solicitação pull são exibidas.

16 de novembro de 2023

[Conteúdo atualizado: Cotas para CodeCatalyst](#)

O tópico [Cotas para o CodeCatalyst](#) foi atualizado com a cota Número máximo de conexões VPC para um espaço.

16 de novembro de 2023

|                                                                                                                                   |                                                                                                                                                                                                         |                        |
|-----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Novo tópico de migração para CodeCatalyst</a>                                                                         | Novo tópico sobre migração.                                                                                                                                                                             | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: Gerenciamento de equipes para um espaço e para CodeCatalyst projetos</a>                               | Foram adicionadas informações sobre o uso de equipes com espaços. Consulte <a href="#">Permitir o acesso ao espaço usando equipes</a> e <a href="#">Permissão do acesso ao projeto usando equipes</a> . | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: gerenciar recursos de máquina para esquemas e fluxos de trabalho em um espaço</a>                      | Foram adicionadas informações sobre o uso de recursos da máquina com espaços. Consulte <a href="#">Permissão do acesso ao espaço para recursos da máquina</a> .                                         | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: Gerenciamento de recursos de máquina para esquemas e fluxos de trabalho em um projeto CodeCatalyst</a> | Foram adicionadas informações sobre o uso de recursos da máquina com CodeCatalyst projetos. Consulte <a href="#">Permissão do acesso ao projeto para recursos da máquina</a> .                          | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: associar uma conexão VPC a um ambiente</a>                                                             | Adição da documentação para associar uma conexão VPC a um ambiente, que pode ser usada em um fluxo de trabalho.                                                                                         | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: associar uma conexão VPC a um Ambiente de Desenvolvimento</a>                                          | Adição da documentação para usar Ambientes de Desenvolvimento com uma conexão VPC.                                                                                                                      | 16 de novembro de 2023 |

|                                                                                                                                         |                                                                                                                                                                                                     |                        |
|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Novo conteúdo</a>                                                                                                           | Publicação inicial do <a href="#">Amazon CodeCatalyst Administrator Guide</a> .                                                                                                                     | 16 de novembro de 2023 |
| <a href="#">Novo conteúdo: YAML da ação “Implantação do AWS CDK ”</a>                                                                   | Adição de uma nova <code>CdkCliVersion</code> propriedade de a <a href="#">YAML da ação “Implantação do AWS CDK ”</a> ou <a href="#">YAML da ação “Inicialização do AWS CDK ”</a> .                 | 14 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                                                                | A documentação dos perfis foi atualizada para incluir permissões para trabalhar com regras de ramificação.                                                                                          | 13 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: solução de problemas com repositórios de origem, fluxos de trabalho e ambientes de desenvolvimento</a> | Os tópicos de solução de problemas foram atualizados para incluir informações sobre como trabalhar com regras de ramificação.                                                                       | 13 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: Ações de criação e de teste YAML</a>                                                                   | Atualização da documentação da propriedade <code>Environment</code> . Agora é um campo opcional para ações de criação e de teste.                                                                   | 13 de novembro de 2023 |
| <a href="#">Novo conteúdo: gerenciar regras de ramificação</a>                                                                          | Adição de documentação de ramificações para incluir informações sobre como visualizar quaisquer regras de ramificação em um repositório de origem, além de criar e gerenciar regras de ramificação. | 13 de novembro de 2023 |



|                                                                                                       |                                                                                                                                                                                                      |                        |
|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a>                                   | A documentação foi atualizada para refletir as alterações na forma como as informações sobre uma solicitação pull são exibidas.                                                                      | 10 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: Armazenar arquivos em cache entre execuções de fluxo de trabalho</a> | A documentação foi atualizada para incluir limitações de armazenamento em cache de arquivos.                                                                                                         | 10 de novembro de 2023 |
| <a href="#">Conteúdo atualizado: Tutorial: Implantar uma aplicação no Amazon EKS</a>                  | A documentação foi atualizada para mencionar o esquema de implantação de aplicações EKS.                                                                                                             | 9 de novembro de 2023  |
| <a href="#">Novo conteúdo: Pacotes em CodeCatalyst</a>                                                | Documentação adicionada para usar pacotes em CodeCatalyst.                                                                                                                                           | 1º de novembro de 2023 |
| <a href="#">Conteúdo novo e atualizado: trabalho com perfis</a>                                       | Atualizou a documentação para quatro novas funções em CodeCatalyst: usuário avançado, acesso limitado, revisor e somente leitura.                                                                    | 1º de novembro de 2023 |
| <a href="#">Conteúdo atualizado: Exportação dos parâmetros de saída do GitHub</a>                     | Os exemplos foram atualizados para usar o arquivo de ambiente GITHUB_OUTPUT em vez do comando set-output . Usar arquivos de ambiente GitHub é o método recomendado para definir parâmetros de saída. | 24 de outubro de 2023  |

[Novo conteúdo: Início da execução automática de um fluxo de trabalho usando gatilhos](#)

Adição da documentação de gatilhos de programação.

16 de outubro de 2023

[Conteúdo atualizado: YAML da ação “Implantar no cluster do Kubernetes”](#)

Adição de informações sobre o uso do perfil CodeCatalystWorkflowDevelopmentRole- *spaceName* aos tópicos [YAML da ação “Implantar no cluster do Kubernetes”](#) e [Tutorial: Implantar uma aplicação no Amazon EKS](#).

22 de setembro de 2023

[Conteúdo atualizado: novo nome do perfil e política do perfil CodeCatalystWorkflowDevelopmentRole- \*spaceName\*](#)

As etapas e as descrições dos perfis foram atualizadas para a alteração do nome do perfil de desenvolvedor para CodeCatalystWorkflowDevelopmentRole- *spaceName* . A função de desenvolvedor agora usa a política AdministratorAccess AWS gerenciada. Consulte [Noções básicas sobre o perfil de serviço CodeCatalystWorkflowDevelopmentRole- \*spaceName\*](#) e [Criar um espaço e perfil de desenvolvimento \(começando sem um convite\)](#).

20 de setembro de 2023

|                                                                                                             |                                                                                                                                                                                                                            |                        |
|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Conteúdo atualizado: <u>Uso de variáveis em fluxos de trabalho</u></a>                          | Introdução de dois novos conceitos: variáveis definidas pelo usuário e variáveis predefinidas. Esses conceitos devem facilitar a leitura e a compreensão da seção <a href="#">Uso de variáveis em fluxos de trabalho</a> . | 19 de setembro de 2023 |
| <a href="#">Conteúdo atualizado: <u>trabalhar com confirmações</u></a>                                      | A documentação foi atualizada para refletir a alteração nas informações exibidas e fornecer detalhes sobre a visualização de confirmações com várias principais.                                                           | 7 de setembro de 2023  |
| <a href="#">Novo conteúdo: <u>Início da execução automática de um fluxo de trabalho usando gatilhos</u></a> | O exemplo a seguir foi adicionado ao tópico <a href="#">Início da execução automática de um fluxo de trabalho usando gatilhos</a> : <a href="#">Exemplo: um gatilho simples “enviar para principal”</a>                    | 6 de setembro de 2023  |
| <a href="#">Conteúdo atualizado: <u>trabalho com solicitações pull</u></a>                                  | A documentação foi atualizada para refletir a alteração na ordem de exibição da ramificação de origem e da ramificação de destino ao criar uma solicitação pull.                                                           | 30 de agosto de 2023   |
| <a href="#">Novo conteúdo: <u>visualizar e alterar a ramificação padrão</u></a>                             | A documentação foi adicionada para ramificações para incluir informações sobre como visualizar e alterar a ramificação padrão de um repositório de origem.                                                                 | 30 de agosto de 2023   |

|                                                                                                            |                                                                                                                                                                                                                                                                                                                             |                      |
|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <a href="#">Conteúdo atualizado: YAML da ação “Implantar no cluster do Kubernetes”</a>                     | Adição de uma nota sobre Helm e Kustomize à descrição da propriedade <a href="#">Manifests</a> no YAML da ação “Implantar no cluster do Kubernetes”.                                                                                                                                                                        | 15 de agosto de 2023 |
| <a href="#">Novo conteúdo: gerenciar anexos de problemas</a>                                               | Foi adicionada a documentação para trabalhar e gerenciar anexos sobre problemas.                                                                                                                                                                                                                                            | 15 de agosto de 2023 |
| <a href="#">Conteúdo atualizado: Início da execução automática de um fluxo de trabalho usando gatilhos</a> | Aprimoramento e expansão da documentação relacionada aos gatilhos de fluxo de trabalho.                                                                                                                                                                                                                                     | 11 de agosto de 2023 |
| <a href="#">Novo conteúdo: solucionar problemas de permissões de perfil</a>                                | Foram adicionadas informações sobre a atualização das permissões da função para executar um fluxo de trabalho que requer acesso à Amazon CodeGuru. Consulte <a href="#">O fluxo de trabalho moderno de três camadas do blueprint de aplicativos web OnPullRequest falha com erro de permissões para a Amazon CodeGuru</a> . | 11 de agosto de 2023 |
| <a href="#">Novo conteúdo: Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?</a>    | Foi adicionado o seguinte tópico de solução de problemas: <a href="#">Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?</a>                                                                                                                                                                          | 11 de agosto de 2023 |

[Novo conteúdo: Implantar no Amazon EKS com um fluxo de trabalho](#)

Adição da documentação para a ação Implantar no cluster do Kubernetes. Para obter mais informações, consulte [Implantar no Amazon EKS com um fluxo de trabalho](#) e [Tutorial: Implantar uma aplicação no Amazon EKS](#).

27 de julho de 2023

[Atualizações sobre como os eventos de gerenciamento são registrados em um espaço CodeCatalyst](#)

Foram adicionadas informações sobre como os eventos de gerenciamento são registrados para ações específicas em um CodeCatalyst espaço com AWS CloudTrail. Foram adicionadas informações sobre como todos os eventos em um espaço podem ser visualizados com o list-events comando. Consulte [Monitoramento de eventos e chamadas de API usando o registro](#).

20 de julho de 2023

[Conteúdo atualizado: Início da execução automática de um fluxo de trabalho usando gatilhos](#)

A documentação foi atualizada para indicar que os acionadores de pull request agora são compatíveis com repositórios de GitHub origem. Anteriormente, os acionadores de pull request eram compatíveis apenas com repositórios de CodeCatalyst origem.

14 de julho de 2023

|                                                                                    |                                                                                                                                                 |                     |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: Cotas para fluxos de trabalho em CodeCatalyst</a> | Atualização do tópico <a href="#">Cotas para fluxos de trabalho em CodeCatalyst</a> com a cota Tempo máximo em que uma ação pode ser executada. | 27 de junho de 2023 |
| <a href="#">Conteúdo atualizado: Definição do YAML do fluxo de trabalho</a>        | Correção de um erro de formatação no bloco de código Compute.                                                                                   | 27 de junho de 2023 |
| <a href="#">Conteúdo atualizado: proteção de dados</a>                             | A documentação foi atualizada para incluir informações adicionais sobre replicação de dados.                                                    | 26 de junho de 2023 |
| <a href="#">Novo conteúdo: Especificação da versão da ação a ser usada</a>         | Adição de um tópico <a href="#">Especificação da versão da ação a ser usada</a> .                                                               | 21 de junho de 2023 |
| <a href="#">Conteúdo atualizado: Implantação em e Contas da AWS VPCs</a>           | Esclarecimento da seção <a href="#">Quais ações permitem que suas informações de implantação sejam exibidas CodeCatalyst?</a> .                 | 14 de junho de 2023 |
| <a href="#">Conteúdo atualizado: documentação de problemas reorganizada</a>        | Reorganização da maior parte da documentação de problemas para se alinhar melhor ao conjunto geral de documentação e aos fluxos de usuários.    | 31 de maio de 2023  |
| <a href="#">Conteúdo atualizado: alternador de visualização de problemas</a>       | Vários fluxos de usuários foram atualizados para se alinharem ao alternador de visualização de problemas atualizado.                            | 31 de maio de 2023  |

|                                                                                             |                                                                                                                                                                                                                                                                                                                               |                    |
|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <a href="#">Conteúdo atualizado: gerenciamento de notificações</a>                          | A documentação das notificações foi atualizada para incluir informações sobre a configuração de notificações pessoais do Slack.                                                                                                                                                                                               | 30 de maio de 2023 |
| <a href="#">Conteúdo atualizado: gerenciamento de notificações</a>                          | A documentação das notificações foi atualizada para incluir informações sobre a configuração de notificações pessoais do Slack.                                                                                                                                                                                               | 30 de maio de 2023 |
| <a href="#">Novo conteúdo: modelo de CodeCatalyst confiança</a>                             | Foi adicionado um novo tópico com informações sobre o modelo de confiança, que CodeCatalyst permite assumir a função de serviço no conectado Conta da AWS. Foi adicionada uma nova seção sobre os princípios de serviço definidos para CodeCatalyst. Consulte <a href="#">Entendendo o modelo de CodeCatalyst confiança</a> . | 20 de maio de 2023 |
| <a href="#">Conteúdo atualizado: Conectar repositórios de origem aos fluxos de trabalho</a> | Simplificação das instruções em <a href="#">Fazer referência a arquivos do repositório de origem</a> .                                                                                                                                                                                                                        | 10 de maio de 2023 |
| <a href="#">Conteúdo atualizado: Cotas para fluxos de trabalho em CodeCatalyst</a>          | O tópico <a href="#">Cotas para fluxos de trabalho em CodeCatalyst</a> foi atualizado com a cota Tamanho máximo de um valor variável de saída.                                                                                                                                                                                | 10 de maio de 2023 |

|                                                                              |                                                                                                                                                                                                                                                                                                                                                              |                     |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Novo conteúdo: Compartilhar artefatos e arquivos entre ações</a> | Foram adicionados dois exemplos: <a href="#">Exemplo: referência de um arquivo em um único artefato</a> e <a href="#">Exemplo: referência a um arquivo em um artefato quando uma WorkflowSource está presente</a> .                                                                                                                                          | 10 de maio de 2023  |
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a>          | A documentação de solicitações pull foi atualizada para incluir informações sobre a configuração de preferências de e-mail para eventos de solicitação pull.                                                                                                                                                                                                 | 21 de abril de 2023 |
| <a href="#">Conteúdo atualizado: gerenciamento de notificações</a>           | A documentação de notificações foi atualizada para incluir informações sobre a configuração de preferências de e-mail para eventos de solicitação pull.                                                                                                                                                                                                      | 21 de abril de 2023 |
| <a href="#">Atualizações das políticas gerenciadas</a>                       | Foram adicionadas as políticas gerenciadas <a href="#">AWS política gerenciada: AmazonCodeCatalystFullAccess</a> , <a href="#">AWS política gerenciada: AmazonCodeCatalystReadOnlyAccess</a> e <a href="#">AWS política gerenciada: AmazonCodeCatalystSupportAccess</a> . Consulte <a href="#">CodeCatalyst atualizações nas políticas AWS gerenciadas</a> . | 20 de abril de 2023 |
| <a href="#">Novo conteúdo: Remoção de um destino de implantação</a>          | Adição de um tópico <a href="#">Remoção de um destino de implantação</a> .                                                                                                                                                                                                                                                                                   | 20 de abril de 2023 |



|                                                                                                                         |                                                                                                                                                                                                                                      |                     |
|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Novo conteúdo: Tipos de ação</a>                                                                            | Adição de um tópico <a href="#">CodeCatalyst ações</a> .                                                                                                                                                                             | 20 de abril de 2023 |
| <a href="#">Atualizações para gerenciar um usuário com o perfil de administrador em um espaço</a>                       | Foram adicionadas informações sobre como remover ou alterar o perfil de um usuário que é administrador em um espaço. Consulte <a href="#">Remoção ou alteração do perfil de um usuário com o perfil de administrador do espaço</a> . | 19 de abril de 2023 |
| <a href="#">Atualizações para administrar ambientes de desenvolvimento</a>                                              | Foram adicionadas informações sobre a administração de Ambientes de Desenvolvimento como administrador do espaço. Consulte <a href="#">Administração de Ambientes de Desenvolvimento para um espaço</a> .                            | 19 de abril de 2023 |
| <a href="#">Conteúdo atualizado: encontrar e visualizar problemas</a>                                                   | Reorganização do tópico <a href="#">Descoberta e visualização de problemas</a> e dos subtópicos.                                                                                                                                     | 19 de abril de 2023 |
| <a href="#">Conteúdo atualizado: Criação de um projeto CodeCatalyst com um repositório de GitLab projetos vinculado</a> | Atualizado <a href="#">Como criar um projeto</a> para incluir GitLab integração na <a href="#">Criar um projeto com um repositório de terceiros vinculado</a> seção.                                                                 | 19 de abril de 2023 |
| <a href="#">Conteúdo atualizado: Configuração de imagens de computação e runtime</a>                                    | Adição de suporte para a arquitetura Arm64 no Amazon Linux 2.                                                                                                                                                                        | 19 de abril de 2023 |

|                                                                                                                                  |                                                                                                                                                                                                                                                                                     |                     |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Novo conteúdo: movimentar problemas em grupos</a>                                                                    | Adição da <a href="#">documentação para mover problemas em grupos</a> nas visualizações Quadro e Todos os problemas.                                                                                                                                                                | 19 de abril de 2023 |
| <a href="#">Conteúdo atualizado: Cotas para fluxos de trabalho em CodeCatalyst</a>                                               | Atualização do tópico <a href="#">Cotas para fluxos de trabalho em CodeCatalyst</a> com cotas ausentes e atualização da cota Tamanho total máximo de variáveis de saída de uma única ação para 120 KB (de 2 KB).                                                                    | 18 de abril de 2023 |
| <a href="#">Novo conteúdo: Visualizar o código-fonte de uma ação</a>                                                             | Adição de um tópico <a href="#">Visualizar o código-fonte de uma ação</a> .                                                                                                                                                                                                         | 18 de abril de 2023 |
| <a href="#">Novo conteúdo: Parada de uma execução de fluxo de trabalho</a>                                                       | Adição de um tópico <a href="#">Parada de uma execução de fluxo de trabalho</a> .                                                                                                                                                                                                   | 10 de abril de 2023 |
| <a href="#">Novo conteúdo: seções adicionadas para marcar recursos para conexões de contas entre AWS e a Amazon CodeCatalyst</a> | Foram adicionadas informações para marcar recursos de conexão da conta e gerenciar políticas do IAM para recursos de conexão. Consulte <a href="#">Uso de tags para controlar o acesso a recursos de conexão de conta</a> e <a href="#">CodeCatalyst referência de permissões</a> . | 6 de abril de 2023  |
| <a href="#">Novo conteúdo: Tipos de ação</a>                                                                                     | Adição de um tópico <a href="#">Tipos de ação</a> .                                                                                                                                                                                                                                 | 6 de abril de 2023  |

|                                                                                                       |                                                                                                                                                                                                                                     |                     |
|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: Ação “Implantar CloudFormation pilha” YAML</a>                       | Atualização da descrição da propriedade parameter-overrides . Agora ela oferece suporte a arquivos JSON.                                                                                                                            | 5 de abril de 2023  |
| <a href="#">Novo conteúdo: Criação de um projeto CodeCatalyst com um GitHub repositório vinculado</a> | Foi adicionada uma nova seção <a href="#">Como criar um projeto</a> intitulada <a href="#">Criar um projeto com um repositório de terceiros vinculado</a> com instruções para criar um projeto vinculado ao seu GitHub repositório. | 5 de abril de 2023  |
| <a href="#">Conteúdo atualizado: trabalho com notificações</a>                                        | A documentação de notificações foi atualizada para incluir informações sobre a configuração de e-mails sobre eventos de projeto.                                                                                                    | 31 de março de 2023 |
| <a href="#">Novo conteúdo</a>                                                                         | Publicação inicial do <a href="#">guia do Amazon CodeCatalyst Action Development Kit</a> .                                                                                                                                          | 31 de março de 2023 |
| <a href="#">Conteúdo atualizado: reestruturou a seção Spaces na Amazon CodeCatalyst</a>               | A seção Espaços foi atualizada removendo páginas de destino e consolidando tópicos.                                                                                                                                                 | 29 de março de 2023 |

|                                                                                                    |                                                                                                                                                                                                                                                      |                     |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <a href="#">Conteúdo atualizado: Tutorial: Implantar uma aplicação no Amazon ECS</a>               | Alterado <a href="#">Etapa 1: configurar um AWS usuário e AWS CloudShell</a> para descrever como criar um usuário em Centro de Identidade do AWS IAM vez de AWS Identity and Access Management. A criação de usuários do IAM não é mais recomendada. | 23 de março de 2023 |
| <a href="#">Conteúdo atualizado: trabalho com perfis</a>                                           | A documentação dos perfis de administrador do espaço, administrador do projeto e colaborador foi atualizada para incluir permissões para vincular problemas a solicitações pull.                                                                     | 13 de março de 2023 |
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a>                                | A documentação de solicitações pull foi atualizada para incluir informações sobre problemas de vinculação a solicitações pull.                                                                                                                       | 13 de março de 2023 |
| <a href="#">Conteúdo atualizado: trabalho com problemas</a>                                        | A documentação de problemas foi atualizada para incluir informações sobre problemas de vinculação a solicitações pull.                                                                                                                               | 13 de março de 2023 |
| <a href="#">Novo conteúdo: Visualizar o status e detalhes de todas as execuções em seu projeto</a> | Adição de uma seção que descreve a nova página de execução do fluxo de trabalho agregado.                                                                                                                                                            | 8 de março de 2023  |

|                                                                                                                        |                                                                                                                                                                                      |                         |
|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <a href="#">Novo conteúdo: Como faço para corrigir os erros “A definição do fluxo de trabalho tem <i>n</i> erros”?</a> | Adição de uma seção sobre como solucionar os erros “A definição do fluxo de trabalho tem erros”.                                                                                     | 7 de março de 2023      |
| <a href="#">Conteúdo atualizado: Criação de um fluxo de trabalho</a>                                                   | As instruções foram atualizadas para refletir a nova interface do usuário.                                                                                                           | 3 de março de 2023      |
| <a href="#">Novo conteúdo: Integrando com universal-test-runner</a>                                                    | Adição de um tópico <a href="#">Integrando com universal-test-runner</a> .                                                                                                           | 3 de março de 2023      |
| <a href="#">Conteúdo atualizado: Compilação, teste e implantação com fluxos de trabalho</a>                            | Várias seções foram atualizadas para refletir os novos filtros de nome de repositório, ramificação e fluxo de trabalho de origem na página de resumo Fluxos de trabalho.             | 2 de março de 2023      |
| <a href="#">Novo conteúdo: Rastreamento do status de implantação por confirmação</a>                                   | Adição de uma seção sobre visualização da qualidade do código e do status de implantação por confirmação.                                                                            | 27 de fevereiro de 2023 |
| <a href="#">Novo conteúdo: variáveis BranchName " e CommitId "</a>                                                     | Adição de uma nova variável BranchName predefinida.                                                                                                                                  | 16 de fevereiro de 2023 |
| <a href="#">Conteúdo atualizado: Gerenciamento de membros espaciais na Amazon CodeCatalyst</a>                         | Informações atualizadas sobre como alterar as funções dos membros, convidar membros e remover membros em duas novas tabelas com base na função atribuída ao usuário em CodeCatalyst. | 15 de fevereiro de 2023 |

|                                                                                                                       |                                                                                                                                              |                         |
|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <a href="#">Conteúdo atualizado: etapas adicionadas para o gerenciamento de PAT no console da Amazon CodeCatalyst</a> | Foram adicionadas etapas para visualização, criação e exclusão PATs no console.                                                              | 15 de fevereiro de 2023 |
| <a href="#">Conteúdo atualizado: Especificação de imagens de ambiente de runtime</a>                                  | Foram adicionadas mais ferramentas à tabela Versões da ferramenta de imagem padrão.                                                          | 10 de janeiro de 2023   |
| <a href="#">Conteúdo atualizado: Compartilhar artefatos e arquivos entre ações</a>                                    | Correção de um caminho de artefato.                                                                                                          | 3 de janeiro de 2023    |
| <a href="#">Conteúdo atualizado: GitHub Ação 'Ações' YAML</a>                                                         | Correção do snippet de código na seção Steps.                                                                                                | 3 de janeiro de 2023    |
| <a href="#">Conteúdo atualizado: Conectar repositórios de origem aos fluxos de trabalho</a>                           | Correção de um caminho de origem.                                                                                                            | 3 de janeiro de 2023    |
| <a href="#">Conteúdo atualizado: atualização de uma solicitação pull</a>                                              | A documentação foi atualizada para incluir informações sobre a atualização de revisores obrigatórios ou opcionais para uma solicitação pull. | 23 de dezembro de 2022  |
| <a href="#">Novo conteúdo: Armazenar arquivos em cache entre execuções de fluxo de trabalho</a>                       | Adição de uma página para armazenamento em cache de arquivos em um fluxo de trabalho.                                                        | 20 de dezembro de 2022  |
| <a href="#">Conteúdo atualizado: trabalho com solicitações pull</a>                                                   | A documentação de solicitações pull foi atualizada para incluir informações sobre notificações.                                              | 16 de dezembro de 2022  |

|                                                                                    |                                                                                                                                                    |                        |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Novo conteúdo: YAML da ação “Implantação do AWS CDK ”</a>              | Adição de uma nova propriedade de CdkRootPath .                                                                                                    | 16 de dezembro de 2022 |
| <a href="#">Novo conteúdo: Compartilhamento de computação entre ações</a>          | Adição de um tópico <a href="#">Compartilhamento de computação entre ações</a> .                                                                   | 14 de dezembro de 2022 |
| <a href="#">Conteúdo atualizado: Compartilhar artefatos e arquivos entre ações</a> | Correção de exemplos mostrando como especificar artefatos de entrada.                                                                              | 13 de dezembro de 2022 |
| <a href="#">Novo conteúdo: GitHub Ação 'Ações' YAML</a>                            | Foi adicionada uma página de referência dedicada para a ação GitHub Ações.                                                                         | 13 de dezembro de 2022 |
| <a href="#">Conteúdo atualizado: Cotas para projetos em CodeCatalyst</a>           | Atualização da documentação com um máximo de cem projetos em um espaço.                                                                            | 2 de dezembro de 2022  |
| <a href="#">Novo conteúdo</a>                                                      | Publicação inicial do Guia CodeCatalyst do usuário da Amazon.                                                                                      | 1º de dezembro de 2022 |
| <a href="#">Novo conteúdo: Solução de problemas com extensões</a>                  | Adição de um tópico de solução de problemas sobre possíveis problemas que os usuários podem encontrar ao usar o recurso de extensões de terceiros. | 18 de junho de 2022    |

# AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.