

Manual do usuário

Amazon CodeCatalyst



Amazon CodeCatalyst: Manual do usuário

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestígie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é a Amazon CodeCatalyst?	1
O que eu posso fazer com CodeCatalyst?	1
Como faço para começar com CodeCatalyst?	2
Saiba mais sobre CodeCatalyst	2
Conceitos	3
AWS Espaços de ID do construtor em CodeCatalyst	4
Espaços que oferecem suporte à federação de identidades em CodeCatalyst	4
Projetos	4
Esquemas	4
Conexões de conta	5
Conexões VPC	5
AWS ID do construtor	5
Perfis de usuário em CodeCatalyst	6
Repositórios de origem	6
Confirmações	7
Ambientes de desenvolvimento	7
Fluxos de trabalho	8
Ações	8
Problemas	8
Tokens de acesso pessoal (PATs)	9
Conexões pessoais	9
Funções	9
Configure e faça login em CodeCatalyst	11
Criando sua primeira função espacial e de desenvolvimento (começando sem um convite)	13
Criando seu primeiro espaço e funções do IAM	14
Aceitando um convite e criando seu AWS Builder ID	20
Aceitando um convite e criando um AWS Builder ID	21
Faça login com seu AWS Builder ID	22
Dispositivos confiáveis	22
Faça login com SSO	23
Veja todos os espaços e projetos de um usuário	23
Visualizando e gerenciando CodeCatalyst perfis	24
Visualizando seu CodeCatalyst perfil	25
Visualizando o CodeCatalyst perfil de outro usuário	25

Atualizando seu perfil	26
Alterando sua CodeCatalyst senha	27
Configurando para usar o AWS CLI com CodeCatalyst	27
Tutoriais de introdução	30
Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas	31
Pré-requisitos	33
Etapa 1: criar o projeto de aplicativo web moderno de três camadas	34
Etapa 2: convidar alguém para seu projeto	35
Etapa 3: criar problemas para colaborar e monitorar o trabalho	36
Etapa 4: Exibir seu repositório de origem	37
Etapa 5: criar um ambiente de desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código	38
Etapa 6: Visualize o fluxo de trabalho que cria o aplicativo moderno	40
Etapa 7: peça que outras pessoas revisem suas alterações	43
Etapa 8: fechar o problema	47
Limpeza de recursos	47
Referência	48
Tutorial: Começando com um projeto vazio	50
Pré-requisitos	50
Crie um projeto vazio	51
Criar um repositório de origem	51
Crie um fluxo de trabalho para criar, testar e implantar uma alteração de código	53
Convide alguém para seu projeto	53
Crie problemas para colaborar e monitorar o trabalho	54
Tutorial: Usando recursos generativos de IA	55
Pré-requisitos	56
Adicione um resumo gerado automaticamente ao criar uma pull request	56
Crie um resumo dos comentários deixados sobre alterações de código em uma pull request	59
Crie um problema e atribua-o ao Amazon Q	60
Limpar recursos	68
Tutorial: Criando um aplicativo de pilha completa com esquemas de PDK compostos	68
Pré-requisitos	70
Etapa 1: criar um projeto monorepo	70
Etapa 2: adicionar a API Type Safe ao projeto	72
Etapa 3: Adicionar um site do Cloudscape React para o projeto	73

Etapa 4: gerar a infraestrutura para implantar o aplicativo na nuvem da AWS	74
Etapa 5: configurar um DevOps fluxo de trabalho para implantar seu projeto	76
Etapa 6: confirme o fluxo de trabalho de lançamento e visualize seu site	78
Colabore e repita o projeto PDK	84
Organize recursos com espaços	101
Criando um espaço	103
Editando um espaço	106
Excluindo um espaço	106
Monitoramento de atividades para usuários e recursos em um espaço	108
Permitindo acesso a AWS recursos com conexão Contas da AWS	108
Adicionando um Conta da AWS a um espaço	109
Adicionar funções do IAM às conexões da conta	113
Adicionar a conexão da conta e as funções do IAM ao seu ambiente de implantação	115
Visualizando conexões da conta	116
Removendo uma conta de um espaço (em CodeCatalyst)	116
Configurar uma conta de cobrança para um espaço	117
Configurando funções do IAM para contas conectadas	118
Função da CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i>	119
Função da AWSRoleForCodeCatalystSupport	120
Criação de uma função do IAM e uso da política de CodeCatalyst confiança	121
Concedendo permissões de espaço aos usuários	122
Visualizando membros em um espaço	123
Convidar um usuário diretamente para um espaço	124
Cancelando um convite para um espaço	125
Alterando a função de um membro do espaço	126
Removendo um membro do espaço	127
Removendo ou alterando a função de um usuário com a função de administrador do Space	128
Permitindo o acesso ao espaço usando equipes	129
Criando uma equipe	130
Visualizando uma equipe	132
Conceder funções espaciais para uma equipe	133
Conceder funções de projeto para uma equipe no nível do espaço	133
Adicionar um usuário diretamente a uma equipe	134
Removendo um usuário de uma equipe diretamente	135
Adicionar um grupo de SSO a uma equipe	136

Excluindo uma equipe	136
Permitindo o acesso ao espaço para os recursos da máquina	137
Visualizando o acesso ao espaço para recursos da máquina	138
Desativando o acesso ao espaço para recursos da máquina	138
Habilitando o acesso ao espaço para os recursos da máquina	139
Administrando ambientes de desenvolvimento para um espaço	140
Visualizando ambientes de desenvolvimento para seu espaço	141
Editando um ambiente de desenvolvimento para seu espaço	142
Interrompendo um ambiente de desenvolvimento para seu espaço	142
Excluindo um ambiente de desenvolvimento para seu espaço	143
Cotas para espaços	144
Organize o trabalho com projetos	146
Criação de um projeto	147
Criando um projeto com um blueprint	147
Criando um projeto vazio	149
Criação de um projeto com um repositório de terceiros vinculado	149
Adicionar recursos e tarefas aos projetos criados	154
Obter uma lista de projetos	154
Visualizando tarefas do projeto e ambientes de desenvolvimento	155
Visualizando todos os projetos em um espaço	156
.....	156
Visualizando as configurações do projeto	156
Mudando para um projeto diferente em CodeCatalyst	157
Excluir um projeto	157
Conceder permissões de projeto aos usuários	158
Obter uma lista dos membros e suas funções no projeto	158
Convidar um usuário para um projeto	159
Cancelando um convite	160
Removendo um usuário do seu projeto	161
Aceitar ou recusar um convite para um projeto	162
Permitindo o acesso ao projeto usando equipes	162
Adicionar uma equipe a um projeto	162
Concessão de funções de projeto para uma equipe	163
Removendo uma função de projeto para uma equipe	164
Permitindo o acesso ao projeto para recursos da máquina	164
Visualizando o acesso ao projeto para recursos da máquina	165

Desabilitando o acesso ao projeto para recursos da máquina	166
Habilitando o acesso ao projeto para recursos da máquina	167
Cotas para projetos	167
Trabalhar com notificações	168
Como as notificações funcionam?	169
Começando a usar as notificações do Slack	170
Gerenciando notificações	174
Configure projetos com plantas	179
Criando um projeto com um blueprint	180
Aplicando um blueprint em um projeto para adicionar recursos	180
Desassociando um blueprint de um projeto	183
Alterando as versões do blueprint em um projeto	183
Editando uma descrição para um blueprint em um projeto	186
Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint	186
Usando o gerenciamento do ciclo de vida em projetos existentes	187
Usando o gerenciamento do ciclo de vida em vários esquemas em um projeto	187
Trabalhando com conflitos em pull requests do ciclo de vida	187
Optando por não participar das mudanças no gerenciamento do ciclo de vida	188
Substituindo o gerenciamento do ciclo de vida de um blueprint em um projeto	188
Criando um projeto abrangente com plantas	189
Plantas disponíveis	189
Encontrando informações sobre o plano do projeto	194
Padronizando projetos personalizados	194
Conceitos de plantas personalizadas	195
Introdução aos esquemas personalizados	199
Tutorial: Criando e atualizando um aplicativo React	204
Trabalhando com o gerenciamento do ciclo de vida como autor do plano	212
Desenvolvendo um plano personalizado para atender aos requisitos do projeto	218
Publicar um blueprint personalizado em um espaço	250
Visualizando detalhes, versões e projetos de um blueprint personalizado	254
Adicionar um blueprint personalizado a um catálogo espacial	255
Removendo um blueprint personalizado de um catálogo espacial	256
Definindo permissões de publicação para um blueprint personalizado	257
Alterando as versões do catálogo para um blueprint personalizado	257
Excluindo um blueprint ou uma versão personalizada publicada	258

Adicionar dependências, lidar com incompatibilidades e atualizar ferramentas e componentes	260
Contribua	262
Cotas para plantas	263
Armazene e colabore no código com repositórios de origem	264
Conceitos do repositório de origem	266
Projetos	4
Repositórios de origem	266
Ambientes de desenvolvimento	7
Tokens de acesso pessoal (PATs)	9
Ramificações	268
Ramificações padrão	268
Confirmações	7
Solicitações pull	269
Revisões	269
Fluxos de trabalho	8
Configuração	270
Instale o Git	271
Crie um token de acesso pessoal	271
Introdução aos repositórios de origem	272
Criando um projeto com um blueprint	272
Visualizando os repositórios de um projeto	274
Criar um ambiente de desenvolvimento	276
Criação de uma solicitação pull	277
Mesclando uma pull request	280
Visualizando o código implantado	281
Como excluir recursos	282
Armazenando o código-fonte em repositórios	282
Criando um repositório de origem	283
Vinculando um repositório de origem	284
Visualizando um repositório de origem	287
Editando as configurações de um repositório de origem	288
Clonando um repositório de origem	289
Excluindo um repositório de origem	292
Organizando seu trabalho de código-fonte com filiais	293
Criando uma filial	294

Gerenciando a ramificação padrão	295
Gerencie ações permitidas com regras de ramificação	296
Comandos Git para filiais	299
Visualizando filiais e detalhes	301
Excluindo uma ramificação	301
Gerenciando arquivos de código-fonte	302
Criando ou adicionando um arquivo	303
Visualizando um arquivo	305
Visualizando o histórico de alterações em um arquivo	306
Editar um arquivo	307
Renomeando ou excluindo um arquivo	308
Revisando o código com pull requests	308
Criação de uma solicitação pull	310
Visualizando pull requests	314
Gerenciando requisitos para fusão com regras de aprovação	316
Analisando uma pull request	318
Atualizando uma pull request	321
Mesclando uma pull request	323
Fechando uma pull request	327
Entendendo as mudanças no código-fonte com commits	327
Visualizando confirmações em uma ramificação	328
Alterando a forma como os commits são exibidos (CodeCatalystconsole)	329
Cotas para repositórios de origem	330
Escreva e modifique código com Dev Environments	335
Criar um ambiente de desenvolvimento	336
Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento	337
Criando um ambiente de desenvolvimento em CodeCatalyst	337
Criando um ambiente de desenvolvimento em um IDE	340
Interromper um ambiente de desenvolvimento	341
Retomar um ambiente de desenvolvimento	342
Editando um ambiente de desenvolvimento	344
Excluir um ambiente de desenvolvimento	345
Conectando-se a um ambiente de desenvolvimento usando SSH	346
Configurando e usando um devfile	348
Editando um devfile	349

Recursos do Devfile suportados pelo CodeCatalyst	351
Exemplo de um devfile para um ambiente de desenvolvimento	352
Solução de problemas de um devfile do repositório usando o modo de recuperação	353
Especificação de imagens de devfile universais	353
Comandos Devfile	358
Eventos Devfile	360
Componentes do Devfile	360
Associando uma conexão VPC a um ambiente de desenvolvimento	361
Cotas para ambientes de desenvolvimento	362
Publique e compartilhe pacotes de software	364
Conceitos de pacotes	365
Pacotes	365
Namespaces de pacotes	365
Versões do pacote	365
Ativos	366
Repositórios de pacotes	366
Repositórios de gateway	366
Repositórios upstream	367
Configurando e usando repositórios de pacotes	367
Criando um repositório de pacotes	367
Conectando-se a um repositório de pacotes	368
Editando um repositório de pacotes	368
Excluindo um repositório de pacotes	369
Configurando e usando repositórios upstream	369
Adicionar um repositório upstream	370
Editando a ordem de pesquisa dos repositórios upstream	371
Solicitar uma versão do pacote com repositórios upstream	372
Removendo um repositório upstream	375
Conectando-se a repositórios externos públicos	375
Repositórios de pacotes externos compatíveis e seus repositórios de gateway	376
Publicando e modificando pacotes	377
Pacotes de publicação	377
Visualizando detalhes da versão do pacote	378
Excluindo uma versão do pacote	379
Atualizando o status da versão de um pacote	379
Editar controles de origem do pacote	381

Uso de npm	387
Configurando e usando o npm	387
Tratamento de tags npm	397
Cotas para pacotes	398
Crie, teste e implante com fluxos de trabalho	399
Sobre o arquivo de definição do fluxo de trabalho	399
Usando os editores visual e YAML do CodeCatalyst console	402
Descobrimo fluxos de trabalho	403
Visualizando detalhes da execução do fluxo de trabalho	404
Próximas etapas	405
Conceitos de fluxos de trabalho	405
Fluxos de trabalho	405
Arquivos de definição de fluxo de trabalho	406
Ações	406
Grupos de ação	406
Artefatos	406
Computação	407
Ambientes do	407
Portões	407
Portões	407
Relatórios	408
Execuções	408
Origens	408
Variáveis	409
Acionadores de fluxo de trabalho	409
Introdução aos fluxos de trabalho	409
Pré-requisitos	410
Etapa 1: criar e configurar seu fluxo de trabalho	411
Etapa 2: salve seu fluxo de trabalho com um commit	412
Etapa 3: Exibir os resultados da execução	413
(Opcional) Etapa 4: Limpar	414
Construindo com fluxos de trabalho	414
Como faço para criar um aplicativo?	414
Benefícios da ação de construção	415
Alternativas à ação de construção	416
Adicionando a ação de construção	416

Visualizando os resultados da ação de compilação	418
Tutorial: Faça upload de artefatos para o Amazon S3	418
Definição YAML da ação de construção e teste	427
Testando com fluxos de trabalho	457
Tipos de relatório de qualidade	457
Adicionando a ação de teste	460
Visualizando os resultados das ações de teste	462
Ignorando testes que falharam	462
Integrando com universal-test-runner	463
Configurando relatórios de qualidade	465
Tentando novamente os casos de teste	472
Práticas recomendadas para testes	472
Propriedades do SARIF	476
Implantação com fluxos de trabalho	486
Como faço para implantar um aplicativo?	486
Lista de ações de implantação	487
Benefícios das ações de implantação	488
Alternativas para implantar ações	488
Implantação no Amazon ECS	489
Implantação no Amazon EKS	542
Implantando uma pilha CloudFormation	590
Implantação de um aplicativo AWS CDK	644
Inicializando um aplicativo AWS CDK	669
Publicar no Amazon S3	686
Implantação em Contas da AWS e VPCs	701
Exibindo o URL do aplicativo	709
Removendo um alvo de implantação	713
Rastreando o status de implantação por confirmação	714
Visualizando registros de implantação	716
Visualizando o status da implantação, confirmações, PRs e muito mais	717
Criação de um fluxo de trabalho	718
Executar um fluxo de trabalho	722
Iniciando um fluxo de trabalho executado manualmente	723
Iniciando um fluxo de trabalho executado automaticamente com gatilhos	723
Interromper uma execução de fluxo de trabalho	741
Bloqueando a execução de um fluxo de trabalho	742

Exigindo aprovações em execuções de fluxo de trabalho	745
Configurando o comportamento de enfileiramento das execuções	758
Armazenando arquivos em cache entre as execuções	765
Visualizando o status e os detalhes da execução do fluxo de trabalho	769
Configurando ações de fluxo de trabalho	774
Tipos de ação	775
Adicionando uma ação	779
Removendo uma ação	780
Desenvolvendo uma ação personalizada	782
Agrupando ações em grupos de ação	782
Configurando ações para depender de outras ações	784
Compartilhamento de dados entre ações usando artefatos	789
Especificando a versão da ação a ser usada	802
Determinar quais versões de uma ação estão disponíveis	804
Visualizando o código-fonte de uma ação	805
Integração com GitHub ações	807
Configurando imagens Docker do ambiente de computação e tempo de execução	845
Tipos de computação	846
Frotas de computação	847
Propriedades de frota sob demanda	847
Propriedades da frota provisionada	849
Criação de uma frota provisionada	850
Editando uma frota provisionada	851
Excluindo uma frota provisionada	851
Atribuindo uma frota provisionada ou computação sob demanda a uma ação	852
Compartilhamento de computação entre ações	854
Especificando imagens do Docker do ambiente de execução	861
Conectando um fluxo de trabalho a um repositório de origem	871
Especificando a fonte que armazenará o arquivo de definição do fluxo de trabalho	871
Especificando a fonte que uma ação de fluxo de trabalho usará	872
Fazendo referência a arquivos em um repositório de origem	874
Variáveis produzidas pela fonte	875
Publicação e importação de pacotes	876
Especificando repositórios de CodeCatalyst pacotes em fluxos de trabalho	877
Exemplos de especificação de repositórios de pacotes em fluxos de trabalho	879
Invocando uma função AWS Lambda	881

Quando usar essa ação	881
Exemplo de fluxo de trabalho que invoca uma função Lambda	882
Adicionando a AWS Lambda ação “invocar”	883
Variáveis produzidas pela AWS Lambda ação “invocar”	886
Definição YAML da ação “AWS Lambda invocar”	887
Modificando um arquivo de definição de tarefa	901
Quando usar essa ação	901
Como funciona a ação “Renderizar a definição de tarefas do Amazon ECS”	902
Exemplo de fluxo de trabalho que modifica um arquivo de definição de tarefas do Amazon ECS	904
Adicionando a ação “Render Amazon ECS task definition”	907
Visualizando o arquivo de definição de tarefa atualizado	909
Variáveis produzidas pela ação “Render Amazon ECS task definition”	911
Definição YAML da ação “Definição de tarefa de renderização”	911
Configurando e usando variáveis	920
Usando variáveis definidas pelo usuário	921
Usando variáveis predefinidas	933
Lista de variáveis predefinidas	937
Configurando e usando segredos	938
Criando um segredo	939
Editando um segredo	939
Usando um segredo	940
Excluindo um segredo	941
Visualizando o status do fluxo de trabalho	942
Cotas de fluxo de trabalho	943
Estados de execução do fluxo de trabalho	945
Estados do fluxo de trabalho	946
Definição de fluxo de trabalho YAML	947
Exemplo de um arquivo de definição de fluxo de trabalho	948
Diretrizes e convenções de sintaxe	949
Propriedades de nível superior	951
Acompanhe e organize o trabalho com problemas	964
Conceitos de questões	965
Problemas ativos	965
Problemas arquivados	966
Destinatário	966

Campos personalizados	966
Estimativa	966
Problema	966
Rótulo	967
Prioridade	967
Status e categorias de status	967
Tarefas	967
Visões	967
Rastreamento do trabalho com problemas	968
Criando um problema	968
Estimando um problema	973
Editando e colaborando em questões	974
Encontrando e visualizando problemas	981
Progredindo em um problema	984
Arquivando um problema	986
Problemas de exportação	987
Organizando o trabalho com listas de pendências, etiquetas e quadros	987
Categorizando o trabalho com rótulos	988
Organizando o trabalho com campos personalizados	989
Rastreamento do trabalho com status personalizados	990
Configurando a estimativa do esforço do problema	992
Ativando ou desativando vários destinatários	992
Criação de uma visualização de problemas	993
Cotas para edições	994
Configure identidade, permissões e acesso no CodeCatalyst	995
Concedendo acesso com funções de usuário	996
Entendendo as funções do usuário em espaços e projetos	996
Visualizando as permissões disponíveis para cada função	999
Visualizando e alterando funções de usuário	1043
Conceda aos usuários acesso ao repositório com tokens de acesso pessoal	1045
Criação de PATs	1045
Visualizando PATs	1047
Excluindo PATs	1049
.....	1050
Criando conexões pessoais	1051
Excluindo conexões pessoais	1052

Configure seu AWS Builder ID para fazer login com autenticação multifator (MFA)	1053
Como registrar um dispositivo para uso com autenticação multifator	1054
Aplicativos autenticadores	1056
Alterando seus dispositivos de MFA	1057
Segurança	1058
Proteção de dados	1059
CodeCatalyst e Identity and Access Management	1061
Validação de conformidade	1127
Resiliência	1128
Segurança da infraestrutura	1129
Análise de configuração e vulnerabilidade	1129
Seus dados e privacidade na Amazon CodeCatalyst	1129
Melhores práticas para ações de fluxo de trabalho	1130
Entendendo o modelo de CodeCatalyst confiança	1131
Monitoramento de eventos e chamadas de API usando o registro	1133
Monitoramento de chamadas de API Contas da AWS usando o AWS CloudTrail registro ..	1136
Acessando eventos registrados usando o registro de eventos	1144
Cotas para identidade, permissão e acesso	1147
Solução de problemas	1148
Problemas ao se inscrever	1149
Problemas ao fazer login	1149
Problemas ao finalizar a sessão	1150
Recebo um erro de função não existe para um fluxo de trabalho com falha	1151
Eu recebo um erro de função em um fluxo de trabalho com falha	1151
Preciso atualizar a função do IAM em um fluxo de trabalho do projeto	1151
Tenho uma solicitação de revisão para minha GitHub conta depois de criar uma conexão pessoal	1152
Como preencho um formulário de suporte?	1152
Adicione funcionalidade a projetos com extensões	1153
Extensões de terceiros disponíveis	1153
Integrando GitHub repositórios em CodeCatalyst	1154
Integrando repositórios do Bitbucket em CodeCatalyst	1155
Integrando problemas do Jira no CodeCatalyst	1156
Conceitos de extensões	1156
Extensões	1156
CodeCatalyst catálogo	1156

Conectando e vinculando	1157
Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos	1157
Etapa 1: instalar uma extensão de terceiros do CodeCatalyst catálogo	1159
Etapa 2: Conecte seu provedor terceirizado ao seu CodeCatalyst espaço	1160
Etapa 3: vincular seus recursos de terceiros ao seu CodeCatalyst projeto	1163
Próximas etapas	1166
Instalando uma extensão em um espaço	1166
Desinstalando uma extensão em um espaço	1167
Conectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira	1168
Desconectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira	1172
Vinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira	1174
Vinculando recursos de fornecedores terceirizados conectados	1175
Vincule um repositório de terceiros durante a criação do projeto	1180
Desvincular GitHub repositórios, repositórios do Bitbucket e projetos do Jira	1181
Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst	1183
Visualizando repositórios de terceiros no CodeCatalyst	1183
Pesquisando problemas com o Jira no CodeCatalyst	1184
Iniciando automaticamente a execução de um fluxo de trabalho após eventos de repositórios de terceiros	1184
Adicionar gatilhos para iniciar a execução do fluxo de trabalho	1185
Restringindo o acesso IP com o GitHub Enterprise Cloud	1186
Endereços IP usados pela extensão de repositórios de terceiros	1187
O bloqueio de pull requests de terceiros é mesclado quando os fluxos de trabalho falham	1187
Vinculando problemas do Jira a pull requests	1188
Visualizando CodeCatalyst eventos em edições do Jira	1189
Pesquise código, problemas, projetos e usuários	1191
Refinando sua consulta de pesquisa	1192
Refinação por tipo	1192
Refinação por campo	1193
Refinando com operadores booleanos	1193
Refinação por projeto	1194
Considerações ao trabalhar com a pesquisa	1194
Referência de campos pesquisáveis	1195
Solução de problemas	1201
Solução de problemas gerais de acesso	1201
Esqueci minha senha	1201

Parte ou toda a Amazon CodeCatalyst não está disponível	1202
Não consigo criar um projeto no CodeCatalyst	1202
Solução de problemas de suporte	1202
Recebo uma mensagem de erro ao acessar AWS Support a Amazon CodeCatalyst	1202
Não consigo criar casos de suporte técnico para meu espaço	1203
Minha conta para casos de suporte não está mais conectada ao meu espaço no CodeCatalyst	1203
Não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) na AWS Support Amazon CodeCatalyst	1204
Parte ou toda a Amazon CodeCatalyst não está disponível	1202
Não consigo criar um projeto no CodeCatalyst	1202
Quero enviar feedback em CodeCatalyst	1202
Solução de problemas de repositórios de origem	1205
Atingi o máximo de armazenamento do meu espaço e vejo avisos ou erros	1205
Eu recebo uma mensagem de erro ao tentar clonar ou enviar para um repositório de CodeCatalyst origem da Amazon	1206
Eu recebo uma mensagem de erro ao tentar confirmar ou enviar para um repositório de CodeCatalyst origem da Amazon	1207
Preciso de um repositório de origem para meu projeto	1207
Meu repositório de origem é totalmente novo, mas contém um commit	1208
Quero uma ramificação diferente como minha ramificação padrão	1208
Estou recebendo e-mails sobre atividades em pull requests	1208
Esqueci meu token de acesso pessoal (PAT)	1208
Um pull request não exibe as alterações que eu espero	1209
Uma pull request mostra o status Não mesclável	1209
Solução de problemas de projetos e plantas	1210
API Java com dependências ausentes do AWS Fargate blueprint para apache- maven-3.8.6	1210
O fluxo de trabalho moderno de três camadas do blueprint de aplicativos web OnPullRequestfalha com erro de permissões para a Amazon CodeGuru	1211
Ainda está procurando resolver seu problema?	1215
Solução de problemas de fluxos de trabalho	1215
Como faço para corrigir as mensagens “O fluxo de trabalho está inativo”?	1216
Como faço para corrigir os erros “A definição do fluxo de trabalho <i>não</i> tem erros”?	1217
Como faço para corrigir os erros “Não é possível localizar credenciais” e ExpiredToken “”?	1219

Como faço para corrigir os erros “Não é possível conectar-se ao servidor”?	1221
Por que os CodeDeploy campos estão ausentes no editor visual?	1221
Como faço para corrigir erros de recursos do IAM?	1222
Como faço para corrigir erros de “instalação do npm”?	1224
Por que vários fluxos de trabalho têm o mesmo nome?	1228
Posso armazenar meus arquivos de definição de fluxo de trabalho em outra pasta?	1228
Como adiciono ações em sequência ao meu fluxo de trabalho?	1229
Por que meu fluxo de trabalho é validado com sucesso, mas falha em tempo de execução?	1229
A descoberta automática não descobre nenhum relatório da minha ação	1229
Minha ação falha em relatórios descobertos automaticamente depois que eu configuro os critérios de sucesso	1230
A descoberta automática gera relatórios que eu não quero	1231
A descoberta automática gera muitos pequenos relatórios para uma única estrutura de teste	1231
Os fluxos de trabalho listados em CI/CD não correspondem aos do repositório de origem	1231
Não consigo criar ou atualizar fluxos de trabalho	1232
Pesquisa de solução de problemas	1233
Não consigo encontrar um usuário no meu projeto	1233
Não vejo o que estou procurando no meu projeto ou espaço	1233
O número de resultados de pesquisa continua mudando quando eu navego pelas páginas	1234
Minha consulta de pesquisa não está sendo concluída	1234
Solução de problemas de contas associadas	1234
Minha solicitação de Conta da AWS conexão recebe um erro de token inválido	1235
O fluxo de trabalho do meu CodeCatalyst projeto da Amazon falha com um erro na conta configurada, no ambiente ou na função do IAM	1235
Preciso de uma conta, função e ambiente associados para criar um projeto	1237
Não consigo acessar a página do Amazon CodeCatalyst Spaces no AWS Management Console	1237
Quero uma conta diferente como minha conta de cobrança	1208
Solução de problemas em ambientes de desenvolvimento	1238
A criação do meu ambiente de desenvolvimento não foi bem-sucedida devido a um problema com cotas	1239
Não consigo enviar alterações do meu ambiente de desenvolvimento para uma ramificação específica em um repositório	1240

Meu ambiente de desenvolvimento não foi retomado	1240
Meu ambiente de desenvolvimento desconectado	1240
Meu ambiente de desenvolvimento conectado ao VPC falhou	1240
Não consigo encontrar em qual diretório meu projeto está	1241
Não consigo me conectar ao meu ambiente de desenvolvimento via SSH	1241
Não consigo me conectar ao meu ambiente de desenvolvimento via SSH porque minha configuração SSH local está ausente	1241
Não consigo me conectar ao meu ambiente de desenvolvimento via SSH porque estou tendo problemas com AWS Config o codecatalyst meu perfil	1242
Solução de problemas de IDE	1242
Solução de problemas de devfiles	1244
Solução de problemas	1246
Não consigo escolher um responsável para o meu problema	1246
Solução de problemas AWS CLI e problemas do SDK	1247
Eu recebo uma mensagem de erro quando entro aws codecatalyst em uma linha de comando ou terminal dizendo que é uma escolha inválida	1247
Eu recebo um erro de credenciais quando executo comandos aws codecatalyst	1247
Entendendo o status atual do serviço com o relatório CodeCatalyst de saúde	1248
CodeCatalyst conceitos de relatório de saúde	1248
O incidente	1249
Status	1249
Capacidades afetadas	1249
Atualizado em	1249
AWS Support para Amazon CodeCatalyst	1250
Cobrança AWS Support para a Amazon CodeCatalyst	1250
Configurando seu espaço AWS Support para a Amazon CodeCatalyst	1253
Acessando o suporte para CodeCatalyst no AWS Management Console	1254
Criando um caso de CodeCatalyst suporte em CodeCatalyst	1255
Resolvendo um caso de suporte em CodeCatalyst	1258
Reabrindo um caso de suporte em CodeCatalyst	1259
Cotas	1260
Histórico do documento	1262
AWS Glossário	1288
.....	mcclxxxix

O que é a Amazon CodeCatalyst?

CodeCatalyst A Amazon 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. CodeCatalyst coloca as ferramentas de que você precisa em um só lugar. Você pode planejar o trabalho, colaborar no código e criar, testar e implantar aplicativos com ferramentas de integração contínua/entrega contínua (CI/CD). Você também pode integrar AWS recursos aos seus projetos Contas da AWS conectando-os ao seu CodeCatalyst espaço. Ao gerenciar todos os estágios e aspectos do ciclo de vida do seu aplicativo em uma única ferramenta, você pode fornecer software com rapidez e confiança.

Em CodeCatalyst, você cria um espaço para representar sua empresa, departamento ou grupo e, em seguida, cria projetos que contêm os recursos necessários para apoiar suas equipes e tarefas de desenvolvimento. CodeCatalyst recursos são estruturados dentro de projetos que vivem dentro de espaços. Para ajudar as equipes a começarem rapidamente, CodeCatalyst fornece esquemas de projetos baseados em linguagem ou ferramentas. Quando você cria um projeto a partir de um blueprint de projeto, o projeto vem com recursos como um repositório de origem com código de amostra, scripts de construção, ações de implantação, servidores virtuais ou recursos sem servidor e muito mais.

O que eu posso fazer com CodeCatalyst?

Você e sua equipe de desenvolvimento podem usar CodeCatalyst para realizar cada aspecto do desenvolvimento de software, desde o planejamento do trabalho até a implantação dos aplicativos. Você pode usar o CodeCatalyst para:

- Repita e colabore no código — Trabalhe de forma colaborativa com sua equipe no código com ramificações, mesclagens, pull requests 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.
- Crie, teste e implante seu aplicativo com fluxos de trabalho — Configure fluxos de trabalho com ações de criação, teste e implantação para lidar com a integração e entrega contínuas de seus aplicativos. Você pode iniciar fluxos de trabalho manualmente ou configurá-los para serem iniciados automaticamente com base em eventos como push de código ou criação ou fechamento de pull requests.

- Priorize o trabalho da sua equipe com o rastreamento de problemas — Use problemas para criar atrasos e monitorar o status das tarefas em andamento com os quadros. Criar e manter um acúmulo saudável de itens para sua equipe trabalhar é uma parte importante do desenvolvimento de software.
- Configure 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 mudanças importantes.

Como faço para começar com CodeCatalyst?

Se você não tem um espaço ou quer aprender como configurar e gerenciar um espaço, recomendamos que você comece com o [Amazon CodeCatalyst Administrator Guide](#).

Se você está começando a trabalhar em um projeto ou espaço, recomendamos que comece por:

- Analisando o [CodeCatalyst conceitos](#)
- [Criando um espaço](#)
- Criando seu primeiro projeto seguindo as etapas em [Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas](#)

Saiba mais sobre CodeCatalyst

Você pode aprender mais sobre a funcionalidade deste CodeCatalyst guia do usuário, bem como sobre os seguintes recursos:

- [AWS DevOps Artigos de blog sobre a Amazon CodeCatalyst](#)
- [Guia de referência de CodeCatalyst APIs da Amazon](#)
- [Guia do desenvolvedor do Amazon CodeCatalyst Action Development Kit](#)
- [CodeCatalyst PERGUNTAS FREQUENTES](#)
- [Testemunhos](#)

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 lançamento automatizados.

Para obter informações conceituais adicionais, consulte os seguintes tópicos:

- [Conceitos do 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](#)
- [Funções](#)

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 repositório 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 repositório de identidades por meio do aplicativo Identity Center e seu mapeamento para o ID do repositório de identidades.

Projetos

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

Para obter 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 a partir de 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 blueprint. Você sintetiza periodicamente o blueprint do projeto, o que atualiza os arquivos do projeto, como dependências de software, e regenera os

recursos. 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`, e muito mais `eslint`, com base no tipo e no idioma do seu aplicativo. Os blueprints do projeto podem gerar arquivos que suportam AWS recursos, como construções, AWS CloudFormation modelos e modelos de CDK. AWS Serverless Application Model

Para obter mais informações sobre esquemas de projetos, consulte [Criando 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.

Para obter 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 de 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 [Atualizando seu perfil](#). Para saber mais sobre o AWS Builder ID, consulte [AWS Builder ID](#) no Referência geral da AWS.

Para obter mais informações sobre como se inscrever e fazer login, consulte [Configure e faça login em 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 o. AWS CLI Seu nome de usuário é o alias que você escolheu quando se inscreveu. Você não pode 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, o endereço de e-mail e a 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 [Atualizando seu perfil](#).

Para obter mais informações sobre como se inscrever e fazer login, consulte [Configure e faça login em CodeCatalyst](#).

Repositórios de origem

Um repositório de origem é onde você armazena com segurança o código e os 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. Você pode 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 blueprint, 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 obter mais conceitos que podem ajudá-lo a trabalhar com repositórios de origem e controle de origem, consulte [Conceitos do repositório de origem](#).

Confirmações

Um commit 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. O commit inclui informações sobre a alteração, incluindo a identidade do usuário que fez a alteração, a hora e a data da alteração, o título do commit e qualquer mensagem incluída sobre a alteração. Para ter 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 aos commits para ajudar a identificar commits específicos.

Para obter mais informações sobre commits, 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 aplicativos incluídas em seu ambiente de desenvolvimento são definidas por um devfile no repositório de origem do seu projeto. Se você não tiver um devfile em seu repositório de origem, um devfile padrão será

aplicado automaticamente. O devfile 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

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Para obter mais informações sobre fluxos de trabalho, consulte [Crie, teste e implante com fluxos de trabalho em CodeCatalyst](#).

Ações

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

Para obter mais informações sobre ações, consulte [Configurando as ações que um fluxo de trabalho executa](#).

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 seu

projeto. Se você estiver usando o desenvolvimento ágil, um problema também pode descrever uma história épica ou de usuário.

Para obter 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. Os PATs representam você CodeCatalyst e você pode gerenciá-los nas 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, certifique-se de armazená-los com segurança em seu computador local. Por padrão, os PATs expiram após um ano.

Para obter mais informações sobre PATs, consulte [Conceda 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 de 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 ter mais informações, consulte [Acessando GitHub recursos com conexões pessoais](#).

Funções

Uma função define o acesso de um usuário aos recursos de um projeto ou espaço e quais ações esse usuário pode realizar. Você escolhe a função 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 uma função administrativa no nível correto pode alterar as funções atribuídas. Por exemplo, um usuário com a função de administrador do projeto tem controle total sobre esse projeto e pode alterar as funções dos usuários nesse projeto. Para obter informações sobre quais funções estão disponíveis e quais permissões cada função tem, consulte [Concedendo acesso com funções de usuário](#).

Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).

Configure e faça login em CodeCatalyst

Há dois tipos de espaço que você pode configurar CodeCatalyst: espaços que suportam usuários do AWS Builder ID e criar um espaço que ofereça suporte à federação de identidades, onde usuários e grupos de SSO são gerenciados no IAM Identity Center. Os usuários em um espaço AWS Builder ID fazem login CodeCatalyst com seu AWS Builder ID, e os usuários em um espaço corporativo CodeCatalyst fazem login usando o portal SSO da empresa associada ao espaço.

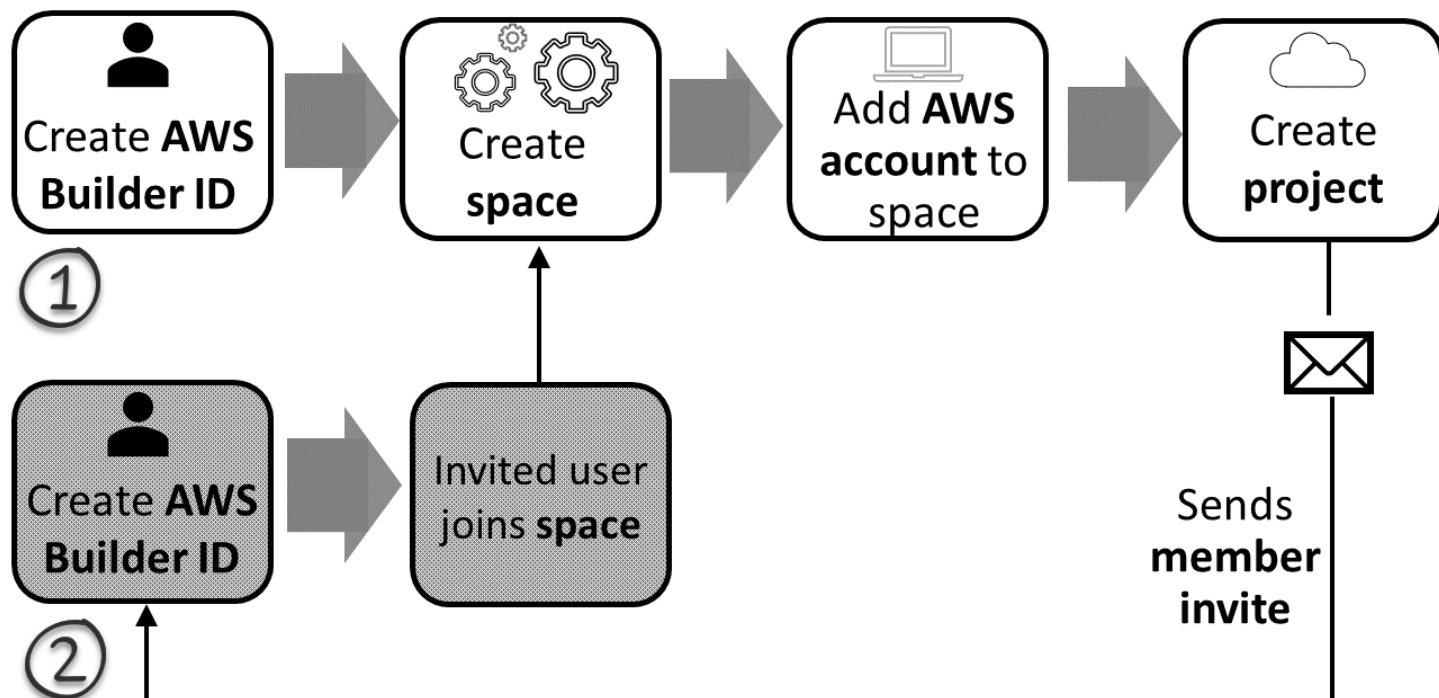
As etapas para configurar e administrar um espaço de ID do AWS Builder são fornecidas neste guia. Para trabalhar com um espaço de ID do CodeCatalyst AWS Construtor, você configurará CodeCatalyst usando as configurações do usuário e o ID do AWS Construtor que você usa para entrar CodeCatalyst.

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.

Esta seção fornece dois caminhos comuns para se configurar para trabalhar na Amazon CodeCatalyst com um espaço AWS Builder ID: 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:

Há dois caminhos comuns para se preparar para trabalhar na Amazon CodeCatalyst: 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. Um Conta da AWS deve ser fornecido para fins de cobrança, onde você ainda pode usar o nível gratuito como padrão.
2. No segundo caso, se você entrar 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 que esteja pronto para começar a trabalhar com outras pessoas.

**i** Tip

CodeCatalyst usa espaços para agrupar projetos e recursos. Ao se inscrever pela primeira vez 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 uma ID de AWS construtor que usará para fazer login CodeCatalyst. Para criar um AWS Builder ID, você fornece o nome completo, a senha e o endereço de e-mail que você usa para entrar nos AWS aplicativos. Você usa o e-mail e a senha para fazer login CodeCatalyst após esse ponto. Você também pode usar esse AWS Builder ID para fazer login em outros aplicativos que usam as credenciais do AWS Builder ID.

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

i Tip

Se você encontrar algum problema ao se inscrever no seu CodeCatalyst perfil da Amazon, siga as etapas fornecidas nessa página. Se precisar de ajuda adicional, consulte [Problemas ao se inscrever](#).

Tópicos

- [Criando sua primeira função espacial e de desenvolvimento \(começando sem um convite\)](#)
- [Aceitando um convite e criando seu AWS Builder ID](#)
- [Faça login com seu AWS Builder ID](#)
- [Faça login com SSO](#)
- [Veja todos os espaços e projetos de um usuário](#)
- [Visualizando e gerenciando CodeCatalyst perfis](#)
- [Configurando para usar o AWS CLI com CodeCatalyst](#)

Criando sua primeira função espacial e de desenvolvimento (começando sem um convite)

Você pode se inscrever na 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 de AWS construtor. Como parte da criação de um espaço, você precisará adicionar um Conta da AWS para fins de cobrança.

i Tip

Se você encontrar algum problema ao se inscrever no seu CodeCatalyst perfil da Amazon, siga as etapas fornecidas nessa página. Se precisar de ajuda adicional, consulte [Problemas ao se inscrever](#).

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

Mary Major é uma desenvolvedora que está interessada em CodeCatalyst e decide experimentá-lo. Ela navega até o CodeCatalyst console e escolhe a opção de se inscrever e criar uma ID de AWS construtor. Mary fornece um endereço de e-mail e uma senha para criar sua ID de AWS construtor. Ela poderá usar seu AWS Builder ID para fazer login em CodeCatalyst outros aplicativos. Quando solicitada a escolher um alias, ela especifica `MaryMajor` como o nome de CodeCatalyst usuário que será exibido em CodeCatalyst e que outros membros do projeto usarão para @mention Mary.

Em seguida, Mary é automaticamente direcionada para criar um espaço. Como parte desse fluxo, Mary é convidada a associar an Conta da AWS ao espaço que ela está criando para que ela possa ver o código de amostra em seu primeiro projeto de criação e implantação. Ela adiciona essas informações e cria seu espaço, onde escolhe a opção de criar uma função de desenvolvimento de pré-visualização que pode ser usada para projetos em seu novo espaço. Mary opta por criar um projeto e, em seguida, visualiza uma lista de plantas para projetos. Depois de analisar as informações dos esquemas disponíveis, ela decide experimentar o modelo moderno de aplicativo web de três camadas para seu 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 cria e implanta automaticamente esse código. Ela explora o código e o fluxo de trabalho, incluindo a visualização do aplicativo web de amostra implantado. Gostando do que vê, ela decide convidar alguns de seus colegas de trabalho para o projeto para começar a explorar. CodeCatalyst


Quando ela tem um momento, Mary configura seu AWS Builder ID para fazer login CodeCatalyst com autenticação multifator (MFA). Com o MFA configurado, Mary pode fazer login CodeCatalyst usando uma combinação de sua CodeCatalyst senha e uma senha ou token de um aplicativo de autenticação de terceiros aprovado.

Criando seu primeiro espaço e funções do IAM

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

O procedimento final cria e adiciona a função de desenvolvedor. A função de desenvolvedor é uma função AWS do IAM que permite que seus CodeCatalyst fluxos de trabalho acessem AWS recursos. A função de desenvolvedor é uma função de serviço usada para gerenciar Serviços da AWS e será criada na conta que está conectada. A função de serviço é uma [função 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 papel terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para obter

mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

 Note

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

Antes de começar, você deve estar pronto para fornecer uma Conta da AWS ID para uma conta na qual tenha privilégios administrativos. Tenha seu Conta da AWS ID de 12 dígitos em mãos. Para obter informações sobre como encontrar seu Conta da AWS ID, consulte [Seu Conta da AWS ID e seu alias](#).

Inscrever um novo usuário

1. Antes de começar no CodeCatalyst console, abra o AWS Management Console e verifique se você está conectado com o mesmo Conta da AWS que deseja usar para criar seu espaço.
2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Na página de boas-vindas, selecione Cadastrar-se. A página Criar seu ID do builder AWS é exibida. Seu AWS Builder ID é uma identidade que você cria para fazer login. Não é o mesmo que um Conta da AWS.
4. Em Seu endereço de e-mail, insira o endereço de e-mail ao qual você deseja se associar CodeCatalyst. Em seguida, escolha Próximo.
5. 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 o nome mais tarde.

Selecione Next (Próximo). A página de verificação de e-mail é exibida.

6. Um código de verificação será enviado para o e-mail que você especificou. Insira esse código em Código de verificação e escolha Verificar. Se você não receber seu código após 5 minutos e não conseguir encontrá-lo nas pastas de spam ou lixo eletrônico, escolha 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 sua concordância com o Contrato do AWS Cliente e os Termos de AWS Serviço e, em seguida, escolha Criar ID do AWS Construtor.

- Na página Criar seu CodeCatalyst 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 posteriormente.

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

Selecione Next (Próximo). A página é atualizada para mostrar a seção Crie seu CodeCatalyst espaço.

- Em Dê um nome ao seu espaço, insira o nome do seu espaço. Você não pode mudar isso mais tarde.

 Note

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

- No menu Região da AWSsuspenso, escolha a região em que você deseja armazenar seu espaço e os dados do projeto. Você não pode mudar isso mais tarde.
- Selecione Next (Próximo). A página é atualizada para mostrar a página para adicionar um Conta da AWS. Essa conta será usada como a conta de cobrança do espaço.
- Em Conta da AWS ID, insira o ID de doze dígitos da conta que você deseja conectar ao seu espaço.

No token de verificação AWS da conta, 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.

- Escolha Ir para o AWS console para verificar.


14. A página Verify Amazon CodeCatalyst Space é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

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

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

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

15. (Opcional) Em Níveis pagos autorizados, escolha Autorizar níveis pagos (Standard, Enterprise) para ativar os níveis pagos em sua conta de cobrança.

 Note

Isso não eleva o nível de cobrança 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 Você pode ativar os níveis pagos a qualquer momento. Sem fazer 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 Verify Amazon CodeCatalyst Space. Escolha o link a seguir: Para adicionar funções do IAM a esse espaço, veja os detalhes do espaço.


A página de conexões com detalhes do CodeCatalyst espaço é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

18. Volte para a CodeCatalyst página e escolha Avançar.

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

- Escolha Ignorar por enquanto.

- Escolha Criar seu primeiro projeto para o seu espaço. Para ver um tutorial que mostra como criar um projeto com um blueprint, consulte [Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas](#)

 Note


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

Para criar e adicionar o CodeCatalyst `CodeCatalystWorkflowDevelopmentRole-spaceName`

1. Antes de começar no CodeCatalyst console, abra o e AWS Management Console, 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 o 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 de Conta da AWS detalhes é exibida.
5. Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

6. Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento. O papel terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

 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. Escolha Criar função 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.

Para criar e adicionar o CodeCatalyst AWSRoleForCodeCatalystSupport

1. Antes de começar no CodeCatalyst console, abra o e AWS Management Console, em seguida, verifique se você está logado com o mesmo Conta da AWS em seu espaço.
2. Navegue até o 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 de Conta da AWS detalhes é exibida.
4. Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

5. Em detalhes do CodeCatalyst espaço, escolha Adicionar função de CodeCatalyst Support. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento de pré-visualização. A função terá um nome `AWSRoleForCodeCatalystSupport` com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função AWSRoleForCodeCatalystSupport de serviço](#).
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.

Depois de criar seu AWS Builder ID, criar seu primeiro espaço e adicionar uma conta, você poderá criar um projeto. Para ter mais informações, consulte [Como criar um projeto](#). Se esta é a primeira vez que você usa CodeCatalyst, sugerimos começar com [Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas](#).

Aceitando um convite e criando seu 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 ajuda adicional, consulte [Problemas ao se inscrever](#).

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 sua ID de AWS construtor. Saanvi poderá usar seu AWS Builder ID para fazer login em outros CodeCatalyst aplicativos. Posteriormente, ela pode editar seu perfil para alterar seu endereço de e-mail ou 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 suas funções no projeto e no espaço. Nas configurações do projeto, o alias de Saanvi aparece na lista de membros com a função atribuída 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 registro de atividades de trabalho do projeto. As edições e comentários de Saanvi mostrarão seu alias, onde outros membros do projeto poderão usá-la como @mention 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 ter mais informações, consulte [Faça login com seu AWS Builder ID](#).

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.

Para aceitar um convite e criar um AWS Builder ID

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

Tip

Seu AWS Builder ID é uma identidade que você cria para entrar. Não é o mesmo que um 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 o nome posteriormente.

Selecione Next (Próximo).

Um código de verificação será enviado para o e-mail que você especificou. Insira esse código em Código de verificação e escolha Verificar. Se você não receber seu código após 5 minutos e não conseguir encontrá-lo nas pastas de spam ou lixo eletrônico, escolha Reenviar código.

4. Depois que seu 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 seu perfil é exibido para sua atividade listada no feed de atividades, mas os membros do projeto usarão seu alias para @mention você.

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

Faça login com seu AWS Builder ID

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

Note

Você já registrou um dispositivo para autenticação multifator (MFA)? É altamente recomendável que você configure o MFA na Amazon CodeCatalyst para aumentar sua segurança. Para obter mais informações, consulte [Como registrar um dispositivo para uso com autenticação multifator](#).

Para fazer login com seu ID do AWS construtor

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Digite seu endereço de e-mail. Opcionalmente, escolha Salvar meu endereço de e-mail se quiser salvar seu endereço de e-mail para futuros logins. Escolha Continue (Continuar).
3. Digite sua senha. Escolha Sign in. Se você não se lembra da sua senha, siga as etapas em [Esqueci minha senha](#).

Dispositivos confiáveis

Depois de escolher a opção Este é um dispositivo confiável na página de login, a Amazon CodeCatalyst considera todos os logins futuros desse dispositivo como autorizados. A Amazon

não CodeCatalyst apresentará a opção de inserir um código de MFA, desde que você use esse dispositivo confiável. Algumas exceções incluem fazer login em um novo navegador ou quando seu dispositivo recebe um endereço IP desconhecido.

Faça login com SSO

Siga estas etapas para usar o SSO para fazer login na Amazon CodeCatalyst.

Em vez disso, para entrar com sua ID do AWS Construtor, consulte [Faça login com seu AWS Builder ID](#).

Para fazer login com SSO

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Em Escolha uma opção de login, escolha Usar login único (SSO).
3. Em Nome do aplicativo do AWS Identity Center, insira o nome do aplicativo fornecido pelo administrador da federação de identidades.
4. Escolha Continuar para o IAM Identity Center.

Veja 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, a função do usuário nesse espaço, como administrador do espaço, e os projetos em cada espaço em que o usuário é membro.

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

The screenshot displays the Amazon CodeCatalyst interface. At the top, there is a header bar with the text 's spaces (9)' on the left, a 'Manage AWS Builder ID' button with an external link icon in the center, and a 'Create space' button on the right. Below the header is a search bar with the placeholder text 'Filter spaces'. The main content area is divided into three sections, each representing a different space:

- EnchantedForest**: A space administrator. It contains two projects: **WildWaves** and **FracturedFairyTales**. Each project card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'. A 'Create project' button is located to the right of the space name.
- org**: A space administrator. It contains four projects: **migration**, **test**, and **12597**. Each project card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'. A 'Create project' button is located to the right of the space name.
- AnyCompany**: A space member. It contains one project: **newproject**. The project card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'. A 'Create project' button is located to the right of the space name.

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

Visualizando e gerenciando CodeCatalyst perfis

Você pode visualizar perfis de usuário na Amazon CodeCatalyst para obter informações como endereços de e-mail e CodeCatalyst aliases. Você também pode atualizar seu perfil e seu ID do AWS Construtor. Se você esquecer sua senha, poderá solicitar uma redefinição de senha.

Visualizando seu CodeCatalyst perfil

Você fornece informações no momento da inscrição que serão usadas como suas credenciais para fazer login na Amazon CodeCatalyst e que serão gerenciadas em seu perfil. Isso inclui seu nome, apelido e o endereço de e-mail que você usa para fazer CodeCatalyst login.

Note

O apelido do AWS Builder ID não é seu CodeCatalyst alias. Você selecionou seu CodeCatalyst alias na inscrição.

Para ver seu CodeCatalyst perfil

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, em seguida, escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.
3. Para atualizar seu endereço de e-mail ou senha do AWS Builder ID, ou para configurar o MFA, escolha Gerenciar ID do AWS Builder. A página AWS Builder ID é aberta.

Visualizando o CodeCatalyst perfil de outro usuário

Para ver o CodeCatalyst perfil de outro usuário

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na navegação lateral, escolha Configurações do projeto. Escolha a guia Membros. Veja a lista de membros do seu CodeCatalyst projeto.
3. Escolha o nome do membro que você deseja pesquisar ou @mention. A página Minhas configurações mostra o alias, o endereço de e-mail e o nome completo do usuário. Use o CodeCatalyst alias para os membros do projeto @mention.

Note

O apelido do AWS Builder ID de um usuário não é seu CodeCatalyst alias. Eles selecionaram seu CodeCatalyst alias no momento da inscrição.

Para ver o perfil de outro usuário em seu projeto, escolha o nome dele na lista.

Atualizando seu perfil

Em CodeCatalyst, seu perfil consiste em informações pessoais gerenciadas pelo AWSBuilder ID e configurações gerenciadas em CodeCatalyst.

- O nome completo, endereço de e-mail e senha do seu perfil são gerenciados pelo AWSBuilder ID. Você inseriu essas informações ao se inscrever. Quando você configura o MFA para usar um aplicativo autenticador para login no aplicativo, você acessa a página CodeCatalyst Builder ID. AWS
- CodeCatalyst as configurações do seu token de acesso pessoal (PAT), CodeCatalyst notificações e preferências de idioma são gerenciadas na página Minhas configurações em CodeCatalyst. Para obter mais informações, consulte [Conceda aos usuários acesso ao repositório com tokens de acesso pessoal](#).

Note

Você pode atualizar seu nome completo do AWS Builder ID (nome de CodeCatalyst exibição) e primeiro nome. No entanto, você não pode alterar seu CodeCatalyst alias.

Atualizando seu ID do AWS Builder ou endereço de e-mail

Para atualizar seu endereço ID do builder AWS ou endereço de e-mail

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, em seguida, escolha Minhas configurações. A página CodeCatalyst Minhas configurações é aberta.
3. Na página de perfil, escolha Gerenciar ID do AWS Builder. A página AWSBuilder ID é aberta.
4. No lado esquerdo da página, escolha Meus detalhes.
5. Em Informações do perfil, escolha Editar para atualizar seu nome ou apelido. Se você não especificou um apelido, o campo Apelido refletirá o primeiro nome no nome completo. Não é seu CodeCatalyst pseudônimo.

Note

Isso atualiza o nome completo e o primeiro nome do AWS Builder ID. Isso não atualiza seu CodeCatalyst alias.

Em Informações de contato, escolha Editar para atualizar seu endereço de e-mail.

Note

Isso atualiza o endereço de e-mail que você usará para fazer login CodeCatalyst.

Alterando sua CodeCatalyst senha

Para alterar sua CodeCatalyst senha

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, em seguida, escolha Perfil do usuário. A página CodeCatalyst Minhas configurações é aberta.
3. Na página de perfil, escolha Gerenciar ID do AWS Builder. A página AWS Builder ID é aberta.
4. No lado esquerdo da página, escolha Segurança.
5. Escolha 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. Instalar 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 [Conceitos básicos 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 codecatalyst` Substitua pelo nome do seu perfil, se necessário.

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

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

- [Conceda aos usuários acesso ao repositório com tokens de acesso pessoal](#)

- [Acessando eventos registrados usando o registro de eventos](#)

Tutoriais de introdução

CodeCatalyst A Amazon fornece vários modelos diferentes para ajudar você a começar seus projetos. Você também pode optar por começar com um projeto vazio e adicionar recursos a ele. Siga as etapas desses tutoriais para aprender algumas das maneiras pelas quais você pode trabalhar. CodeCatalyst

Se esta é a primeira vez que você usa CodeCatalyst, sugerimos começar com [Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas](#).

Note

Para seguir esses tutoriais, você deve primeiro concluir a configuração. Para ter mais informações, consulte [Configure e faça login em CodeCatalyst](#).

Tópicos

- [Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas](#)
- [Tutorial: começar com um projeto vazio e adicionar recursos manualmente](#)
- [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#)
- [Tutorial: Criando um aplicativo de pilha completa com esquemas de PDK compostos](#)

Para obter tutoriais adicionais que se concentram em áreas funcionais específicas em CodeCatalyst, consulte:

- [Começando a usar as notificações do Slack](#)
- [Introdução aos repositórios CodeCatalyst de origem e ao blueprint do aplicativo de página única](#)
- [Introdução aos fluxos de trabalho](#)
- [Introdução aos esquemas personalizados](#)
- [Comece com o guia do desenvolvedor de CodeCatalyst ações da Amazon](#)

Para tutoriais detalhados, consulte:

- [Tutorial: Faça upload de artefatos para o Amazon S3](#)

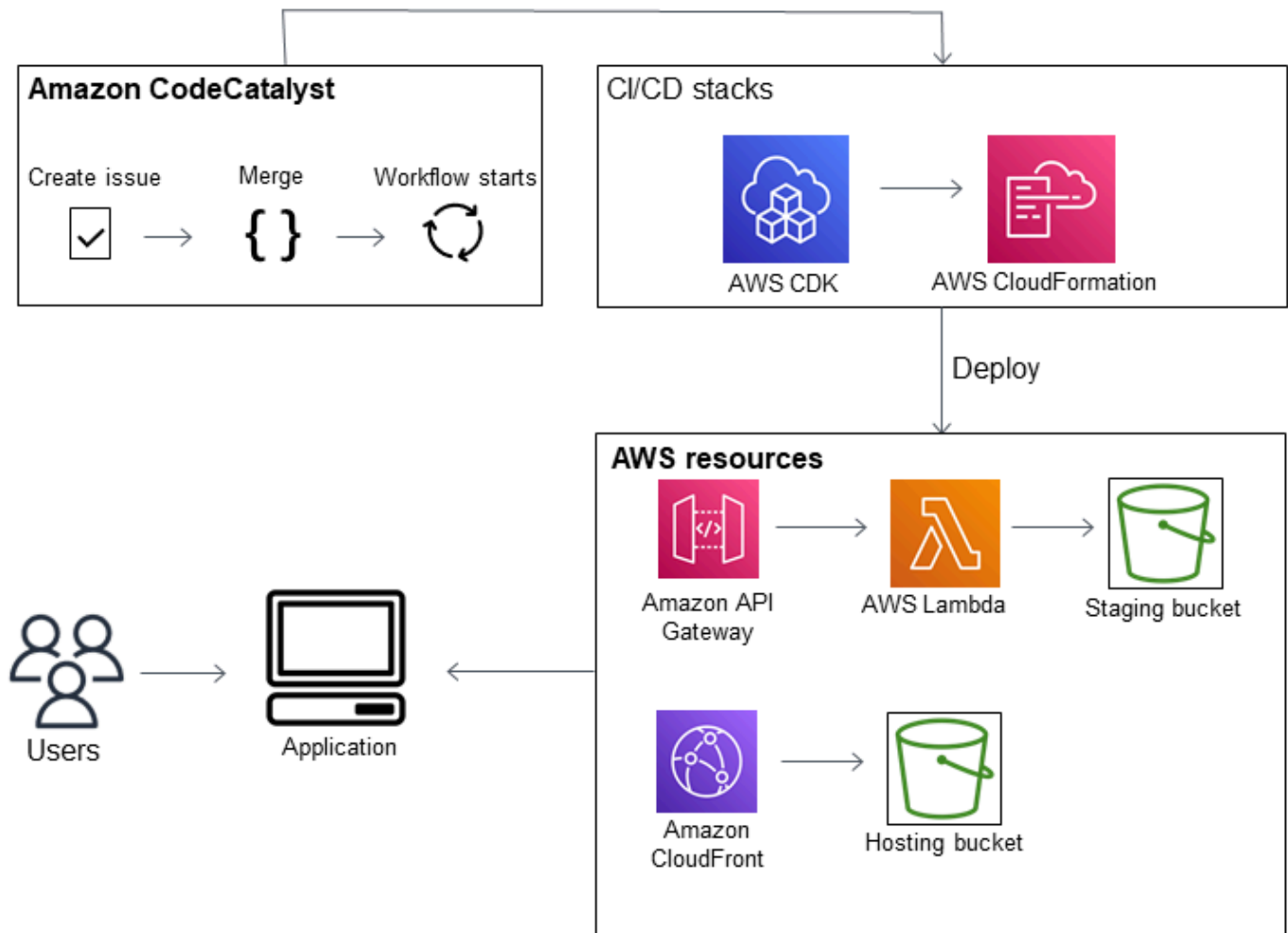
- [Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation](#)
- [Tutorial: Implantar um aplicativo no Amazon ECS](#)
- [Tutorial: Implantar um aplicativo no Amazon EKS](#)
- [Tutorial: código Lint usando uma GitHub ação em um fluxo de trabalho](#)
- [Tutorial: Criando e atualizando um aplicativo React](#)

Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas

Você pode começar mais rapidamente com o desenvolvimento de software criando um projeto com um blueprint. Um projeto criado com um blueprint inclui os recursos de que você precisa, incluindo um repositório de origem para gerenciar seu código e um fluxo de trabalho para criar e implantar o aplicativo. Neste tutorial, explicaremos como usar o modelo de aplicativo web moderno de três camadas para criar um projeto na Amazon. CodeCatalyst O tutorial também inclui visualizar a amostra implantada, convidar outros usuários para trabalhar nela e fazer alterações no código com pull requests que são automaticamente criadas e implantadas nos recursos do Connected Conta da AWS quando a pull request é mesclada. Onde CodeCatalyst cria seu projeto com relatórios, feeds de atividades e outras ferramentas, seu blueprint cria AWS recursos Conta da AWS associados ao seu projeto. Seus arquivos de blueprint permitem que você crie e teste um exemplo de aplicativo moderno e o implante na infraestrutura do Nuvem AWS.

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

As ações geram recursos no associado Conta da AWS e implantam seu aplicativo em uma AWS Lambda função sem servidor com um endpoint do API Gateway. A AWS Cloud Development Kit (AWS CDK) ação converte uma ou mais AWS CDK pilhas em AWS CloudFormation modelos e implanta pilhas em suas. Conta da AWS Os recursos em suas pilhas incluem CloudFront recursos da Amazon para distribuir conteúdo dinâmico da web, uma instância do Amazon DynamoDB para os dados do seu aplicativo e as funções e políticas que dão suporte ao aplicativo implantado.



Quando você cria um projeto com o blueprint de aplicativo web moderno de três camadas, seu projeto é criado com os seguintes recursos:

No CodeCatalyst projeto:

- Um [repositório de origem](#) com código de amostra e fluxo de trabalho YAML
- 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 uma lista de pendências que você pode usar para planejar e monitorar o trabalho
- Um pacote de relatórios de teste com relatórios automatizados incluídos no código de amostra

No associado Conta da AWS:

- Três AWS CloudFormation pilhas que criam os recursos necessários para o aplicativo.

Para obter detalhes mais detalhados sobre os recursos que serão criados em AWS e CodeCatalyst como parte deste tutorial, consulte [Referência](#).

Note

Os recursos e amostras incluídos em um projeto dependem do blueprint selecionado. A Amazon CodeCatalyst oferece vários planos de projetos que definem recursos relacionados à linguagem ou estrutura definida. Para saber mais sobre plantas, consulte [Criando um projeto abrangente com CodeCatalyst plantas](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar o projeto de aplicativo web moderno de três camadas](#)
- [Etapa 2: convidar alguém para seu projeto](#)
- [Etapa 3: criar problemas para colaborar e monitorar o trabalho](#)
- [Etapa 4: Exibir seu repositório de origem](#)
- [Etapa 5: criar um ambiente de desenvolvimento com uma ramificação de teste e fazer uma alteração rápida no código](#)
- [Etapa 6: Visualize o fluxo de trabalho que cria o aplicativo moderno](#)
- [Etapa 7: peça que outras pessoas revisem suas alterações](#)
- [Etapa 8: fechar o problema](#)
- [Limpeza de recursos](#)
- [Referência](#)

Pré-requisitos

Para criar um projeto de aplicativo moderno neste tutorial, você deve ter concluído as tarefas da [Configure e faça login em CodeCatalyst](#) seguinte forma:

- Tenha um ID de AWS construtor para fazer login CodeCatalyst.

- Pertença a um espaço e tenha a função de administrador do espaço ou usuário avançado atribuída a você nesse espaço. Para obter mais informações, consulte [Criando um espaço](#), [Concedendo permissões de espaço aos usuários](#) e [Função de administrador de espaço](#).
- Conta da AWS Associe-se ao seu espaço e tenha a função do IAM que você criou durante a inscrição. Por exemplo, durante a inscrição, você tem a opção de escolher criar uma função de serviço com uma política de função chamada política de CodeCatalystWorkflowDevelopmentRole-*spaceName*função. A função terá um nome CodeCatalystWorkflowDevelopmentRole-*spaceName* com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName*de serviço](#). Para obter as etapas para criar a função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName*função para sua conta e espaço](#).

Etapa 1: criar o projeto de aplicativo web moderno de três camadas

Depois de criá-lo, seu projeto é onde você desenvolverá e testará o código, coordenará as tarefas de desenvolvimento e visualizará as métricas do projeto. Seu projeto também contém suas ferramentas e recursos de desenvolvimento.

Neste tutorial, você usará o blueprint de aplicativo web moderno de três camadas para criar um aplicativo interativo. O fluxo de trabalho criado e executado automaticamente como parte do seu projeto criará e implantará o aplicativo. O fluxo de trabalho só é executado com êxito depois que todas as funções e informações da conta estiverem configuradas para o seu espaço. Depois que o fluxo de trabalho for executado com êxito, você poderá visitar a URL do endpoint para ver o aplicativo.

Para criar um projeto com um blueprint

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.
3. Escolha Criar projeto.
4. Escolha Começar com um blueprint.
5. Na barra de pesquisa, insira **modern**.
6. Selecione o blueprint de aplicativo web moderno de três camadas e, em seguida, escolha Avançar.
7. Em Nomear seu projeto, insira o nome do projeto. Por exemplo: .

MyExampleProject.

Note

O nome deve ser exclusivo em seu espaço.

8. Em Conta, escolha o Conta da AWS que você adicionou durante a inscrição. O blueprint instalará recursos nessa conta.
9. Em Deployment Role, escolha a função que você adicionou durante a inscrição. Para este exemplo, selecione `CodeCatalystWorkflowDevelopmentRole-spaceName`.

Se não houver funções listadas, adicione uma. Para adicionar uma função, escolha Adicionar função do IAM e adicione a função à sua Conta da AWS. Para ter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

10. Na plataforma de computação, escolha Lambda.
11. Na Opção de hospedagem de front-end, escolha Amplify Hosting. Para obter informações sobre AWS Amplify, consulte [O que é AWS Amplify hospedagem?](#) no Guia do AWS Amplify usuário.
12. Em Região de implantação, insira o código da região em que você deseja Região da AWS que o blueprint implante o aplicativo Mysfits e os recursos de suporte. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no Referência geral da AWS.
13. Em Nome do aplicativo, deixe o padrão `demysfitsstring`.
14. (Opcional) Em Gerar visualização prévia do projeto, escolha Exibir código para visualizar os arquivos de origem que o blueprint instalará. Escolha Exibir fluxo de trabalho para visualizar os arquivos de definição do fluxo de trabalho de CI/CD que o blueprint 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 seu projeto

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

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

Ao terminar de adicionar usuários, escolha Convidar.

Etapa 3: criar problemas para colaborar e monitorar o trabalho

CodeCatalyst ajuda você a monitorar recursos, tarefas, bugs e qualquer outro trabalho envolvido em seu projeto com problemas. Você pode criar problemas para monitorar o trabalho e as ideias necessárias. Por padrão, quando você cria um problema, ele é adicionado à sua lista de pendências. Você pode mover os problemas para um quadro onde você acompanha 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, escolha Problemas.
2. Escolha Criar problema.
3. Em Título da edição, forneça um nome para a edição. Opcionalmente, 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 Destinatário, escolha +Adicionar para atribuir o problema a você mesmo.
5. Escolha Criar problema. O problema agora está visível no quadro. Escolha o cartão para mover a edição para a coluna Em andamento.

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

Etapa 4: Exibir seu repositório de origem

Seu blueprint instala um repositório de origem que contém arquivos para definir e dar suporte ao seu aplicativo ou serviço. Alguns diretórios e arquivos dignos de nota no repositório de origem são:

- Diretório.cloud9 — contém arquivos de suporte para o AWS Cloud9 ambiente de desenvolvimento.
- Diretório.codecatalyst — contém o arquivo de definição do YAML fluxo de trabalho para cada fluxo de trabalho incluído no blueprint.
- Diretório.idea — Contém arquivos de suporte para os ambientes de JetBrains desenvolvimento.
- Diretório.vscode — contém arquivos de suporte para o Visual Studio Code Dev Environment.
- Diretório CDKStacks — Contém os arquivos de AWS CDK pilha que definem a infraestrutura no. Nuvem AWS
- diretório src — Contém o código-fonte do aplicativo.
- diretório de testes — contém arquivos para os testes integ e unitários que são executados como parte do fluxo de trabalho automatizado de CI/CD executado quando você cria e testa seu aplicativo.
- diretório web — Contém o código-fonte do front-end. Outros arquivos incluem arquivos de projeto, como o package.json arquivo que contém metadados importantes sobre seu projeto, a index.html página do site, o .eslintrc.cjs arquivo para código de linting e o tsconfig.json arquivo para especificar arquivos raiz e opções do compilador.
- Dockerfilearquivo — Descreve o contêiner do aplicativo.
- README.mdarquivo — Contém informações de configuração do projeto.

Para navegar até os repositórios de origem de um projeto

1. Navegue até seu projeto e faça o seguinte:
 - Na página de resumo do seu projeto, escolha o repositório que você deseja na lista e, em seguida, escolha Exibir repositório.
 - No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem. Em Repositórios de origem, escolha 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 pull requests e fluxos de trabalho. Por padrão,

o conteúdo da ramificação padrão é exibido. Você pode alterar a exibição escolhendo 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 rapidamente no código em seu repositório de origem criando um ambiente de desenvolvimento. Para este tutorial, presumimos que você:

- Crie um ambiente de AWS Cloud9 desenvolvimento.
- Escolha a opção de trabalhar em uma nova ramificação fora da ramificação principal ao criar o Dev Environment.
- Use o nome `test` dessa nova filial.

Em uma etapa posterior, você usará o Dev Environment para fazer uma alteração no código e criar uma pull request.

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. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, escolha Repositórios de origem e escolha o repositório para o qual você deseja criar um ambiente de desenvolvimento.
4. Na página inicial do repositório, escolha Create Dev Environment.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para 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 seu aplicativo de amostra gerado CodeCatalyst fazendo alterações no código com pull requests que são automaticamente criadas e implantadas nos recursos do Connected Conta da AWS quando a pull request é mesclada.

Para fazer uma alteração em seu `src/mysfit_data.json` arquivo

- Navegue até o ambiente de desenvolvimento do seu projeto. Em AWS Cloud9, expanda o menu de navegação lateral para navegar pelos arquivos. `mysfits` expanda e abra `src/mysfit_data.json`.
- No arquivo, altere o valor do "Age": campo de 6 para 12. Sua linha deve ter a seguinte aparência:

```
{
  "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 `cd /projects/mysfits` comando.

5. Adicione, confirme e envie suas alterações com os comandos `git add`, `git commit` e `git push`.

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

Etapa 6: Visualize o fluxo de trabalho que cria o aplicativo moderno

Depois de criar o projeto de aplicativo moderno, 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, CodeCatalyst criei um fluxo de trabalho e o iniciei automaticamente quando você criou seu projeto. (O fluxo de trabalho ainda pode estar em execução, dependendo de há quanto tempo você criou seu projeto.) Use os procedimentos a seguir para verificar o progresso do fluxo de trabalho, analisar os registros gerados e os relatórios de teste e, por fim, navegar até a URL do aplicativo implantado.

Para verificar o progresso do fluxo de trabalho

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

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

2. Observe a lista de fluxos de trabalho. Você deve ver quatro:
 - Os dois fluxos de trabalho na parte superior correspondem à `test` ramificação que você criou anteriormente. [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 filial. `main` O não `ApplicationDeploymentPipeline` está ativo porque está configurado para uso com a `main` ramificação. O `OnPullRequest` fluxo de trabalho não foi executado porque nenhuma solicitação de pull foi feita.
 - Os dois fluxos de trabalho na parte inferior correspondem à `main` ramificação que foi criada quando você executou o blueprint anteriormente. O `ApplicationDeploymentPipeline` fluxo de trabalho está ativo e tem uma execução em andamento (ou concluída).

Note

Se a `ApplicationDeploymentPipeline` execução falhar com um `DeployBackend` erro ou `Build @cdk_bootstrap`, pode ser porque você executou o aplicativo web moderno de três camadas anteriormente e ele deixou recursos antigos para trás e esse conflito com o blueprint atual. Você precisará excluir esses recursos antigos e, em seguida, executar novamente o fluxo de trabalho. Para ter mais informações, consulte [Limpeza de recursos](#).

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



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 verticalmente, com as ações na parte superior sendo executadas antes das 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 o commit que iniciou automaticamente a execução do fluxo de trabalho. CodeCatalyst gerou esse commit quando você criou seu projeto.
 - `Construir` — Esse bloco representa um agrupamento de duas ações que devem ser concluídas com êxito para que a próxima ação seja iniciada.
 - `DeployBackend`— Esse bloco representa uma ação que implanta os componentes de back-end do aplicativo na AWS nuvem.
 - `Testes` — Esse bloco representa um agrupamento de duas ações de teste que devem ser concluídas com êxito para que a próxima ação seja iniciada.
 - `DeployFrontend`— Esse bloco representa uma ação que implanta os componentes de front-end do aplicativo na AWS nuvem.
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 `Triggers` seção, na parte superior. Esta seção indica que o fluxo de trabalho deve começar sempre que o código for enviado para a `main` ramificação 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 `main` ramificação.
 - Uma `Actions` seção, abaixo `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 registros de compilação e os relatórios de teste

1. Escolha a guia Estado mais recente.
2. No diagrama do fluxo de trabalho, escolha a `DeployFrontend` ação.
3. Aguarde até que a ação termine. Observe que o ícone “em andamento” () muda para um ícone de “sucesso” (.
4. Escolha a ação `build_backend`.
5. Escolha a guia Registros e expanda algumas seções para ver as mensagens de registro dessas etapas. Você pode ver mensagens relacionadas à configuração do back-end.
6. Escolha a guia Relatórios e, em seguida, escolha o `backend-coverage.xml` relatório. 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 obter mais informações sobre relatórios de teste, consulte [Testando com fluxos de trabalho](#).

i Tip

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

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

Para confirmar que o aplicativo moderno foi implantado com sucesso

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

i Note

Se você não vê o link Exibir aplicativo dentro da DeployFrontendação, certifique-se de escolher o link de ID de execução.

3. Procure o pégaso Mysfit chamado Twilight Glitter. Anote o valor da idade. Isso é6. Você fará uma alteração no código para atualizar a idade.


Etapa 7: peça que outras pessoas revisem suas alterações

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

Para criar uma pull request

1. Navegue até o projeto.
2. Execute um destes procedimentos:
 - No painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha Create pull request.
 - Na página inicial do repositório, escolha Mais e, em seguida, escolha Criar pull request.
 - Na página do projeto, escolha Criar pull request.


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 pull request na página principal do repositório.
4. Na ramificação de destino, escolha a ramificação na qual mesclar o código depois que ele for revisado.
5. Na ramificação de origem, escolha a ramificação que contém o código confirmado.
6. Em Título do pull request, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
7. (Opcional) Na descrição da pull request, forneça informações como um link para problemas ou uma descrição de suas 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 ao pull request.

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).


8. (Opcional) Em Problemas, escolha Vincular problemas e, em seguida, escolha um problema na lista ou insira sua ID. Para desvincular um problema, escolha o ícone de desvinculação.
9. (Opcional) Em Revisores obrigatórios, escolha Adicionar revisores obrigatórios. Escolha na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que o pull request possa ser mesclado na ramificação de destino.

 Note

Você não pode adicionar um revisor como revisor obrigatório e como revisor opcional. Você não pode se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, escolha Adicionar revisores opcionais. Escolha na lista de membros do projeto para adicioná-los. Os revisores opcionais não precisam aprovar as alterações como um requisito para que a pull request possa ser incorporada à ramificação de destino.

11. Analise as diferenças entre as filiais. A diferença exibida em uma pull request 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 pull request 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 você estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Criar.

 Note


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

Quando você cria a pull request, o OnPullRequestfluxo de trabalho começa a usar os arquivos de origem na test ramificação. Enquanto seus 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 pull request do console CodeCatalyst

1. Navegue até seu projeto de aplicativo moderno.
2. Na página do projeto, em Abrir pull requests, escolha a pull request que você deseja mesclar. Se você não vê a pull request, escolha Exibir tudo e, em seguida, escolha-a na lista. Escolha Merge (Mesclar).
3. Escolha entre as estratégias de mesclagem disponíveis para o pull request. Opcionalmente, selecione ou desmarque a opção de excluir a ramificação de origem após mesclar a pull request e, em seguida, escolha Mesclar.

 Note

Se o botão Mesclar não estiver ativo ou você ver o rótulo Não mesclável, um ou mais 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 pull request é indicado por um ícone de relógio em Visão geral na área de detalhes da pull request. Se todos os revisores obrigatórios aprovaram o pull request, 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 CodeCatalyst console e, em seguida, mesclar a pull request, ou pode resolver conflitos e mesclar localmente e, em seguida, enviar o commit que contém a mesclagem para. CodeCatalyst Para obter mais informações, consulte [Mesclando uma pull request \(Git\)](#) e sua documentação do Git.

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

Para ver a confirmação mesclada ser executada no fluxo de trabalho `ApplicationDeploymentPipeline`

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



Etapa 8: fechar o problema

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

Para encerrar um problema de um projeto

1. Navegue até o projeto.
2. No painel de navegação, escolha Problemas.
3. Coloque rag-and-drop o problema na coluna Concluído.

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

Limpeza de recursos

Limpe CodeCatalyst e AWS remova vestígios deste tutorial do seu ambiente.

Você pode optar por continuar usando o projeto usado neste tutorial ou excluir o projeto e seus 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 escolha Configurações do projeto.
2. Na guia General (Geral), escolha Edit (Editar).
3. Abaixo do nome do projeto, escolha Excluir projeto.

Para excluir recursos no AWS CloudFormation Amazon S3

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

3. Exclua a pilha de *strings* mysfits.
4. **Exclua a pilha de strings development-mysfits.**
5. Escolha (mas não exclua) a pilha CDKToolkit. Escolha a guia Recursos. Escolha o StagingBucketlink 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 blueprint de aplicativo web moderno de três camadas.

6. (Opcional) Exclua a pilha CDKToolkit.

Referência

O modelo moderno de aplicativo web de três camadas implanta recursos em seu CodeCatalyst espaço e em sua AWS conta na nuvem. AWS Esses recursos são:

- No seu CodeCatalyst espaço:
 - Um CodeCatalyst projeto que inclui os seguintes recursos:
 - Um [repositório de origem](#) — Esse repositório contém código de amostra para um aplicativo web 'Mysfits'.
 - Um [fluxo de trabalho](#) — Esse fluxo de trabalho cria e implanta o código do aplicativo Mysfits sempre que uma alteração é feita na ramificação padrão
 - [Quadro de problemas](#) e lista de pendências — Esse quadro e a lista de pendências podem ser usados para planejar e acompanhar o trabalho.
 - Um [conjunto de relatórios de teste](#) — Esse pacote inclui relatórios automatizados incluídos no código de amostra.
- No associado Conta da AWS:
 - Uma pilha CDKToolkit — Essa pilha implanta os seguintes recursos:
 - Um bucket de armazenamento do Amazon S3, uma política de bucket e a AWS KMS chave usada para criptografar o bucket.
 - Uma função de implantação do IAM para a ação de implantação.
 - AWS Funções e políticas do IAM em apoio aos recursos na pilha.

Note

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

- Uma pilha de **strings** development-mysfits — Essa BackEnd pilha implanta os seguintes recursos de back-end:
 - Um endpoint do Amazon API Gateway.
 - AWS Funções e políticas do IAM em apoio aos recursos na pilha.
 - Uma AWS Lambda função e uma camada fornecem a plataforma de computação sem servidor para o aplicativo moderno.
 - Uma política e uma função do IAM para a implantação do bucket e a função Lambda.
- Uma pilha de **strings** mysfits — Essa pilha implanta o aplicativo de front-end. AWS Amplify

Consulte também

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

- Amazon S3 — Um serviço para armazenar seus ativos de front-end em um serviço de armazenamento de objetos que oferece escalabilidade, alta disponibilidade de dados, segurança e desempenho líderes do setor. Para obter mais informações, consulte o Guia [do usuário do Amazon S3](#).
- Amazon API Gateway — Um serviço para criar, publicar, manter, monitorar e proteger REST, HTTP e WebSocket APIs em qualquer escala. Para obter mais informações, consulte o Guia do [desenvolvedor do API Gateway](#).
- Amplify — Um serviço para hospedar seu aplicativo de front-end. Para obter mais informações, consulte o [Guia do usuário de AWS Amplify hospedagem](#).
- AWS Cloud Development Kit (AWS CDK)— Uma estrutura para definir a infraestrutura de nuvem em código e provisioná-la por meio dela. AWS CloudFormation AWS CDK Isso inclui o AWS CDK Toolkit, que é uma ferramenta de linha de comando para interagir com AWS CDK aplicativos e pilhas. Para obter 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 obter mais informações, consulte o [Guia do desenvolvedor do Amazon DynamoDB](#).
- AWS Lambda— Um serviço para invocar seu código em uma infraestrutura computacional 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 com segurança o acesso AWS e seus recursos. Para obter mais informações, consulte o [Manual do usuário do IAM](#).

Tutorial: começar com um projeto vazio e adicionar recursos manualmente

Você pode criar um projeto vazio sem nenhum recurso predefinido dentro dele escolhendo o blueprint do 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 blueprint ficam vazios na criação, essa opção exige mais conhecimento sobre criação e configuração de CodeCatalyst recursos para começar.

Tópicos

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

Pré-requisitos

Para criar um projeto vazio, você deve ter a função de administrador do espaço ou usuário avançado atribuída a você. Se esta é a primeira vez que você faz login CodeCatalyst, consulte [Configure e faça login em CodeCatalyst](#).

Crie 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 seu projeto. Os membros do projeto podem clonar esse repositório em seus computadores locais para trabalhar no código. Como alternativa, você pode optar por vincular um repositório hospedado em um serviço compatível, mas isso não é abordado neste tutorial. Para ter mais informações, consulte [Vinculando 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 obter 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. (Opcional) Adicione um `.gitignore` arquivo para o tipo de código que você planeja enviar.
8. 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 alterar ou excluir a ramificação padrão.

Você pode adicionar código rapidamente ao seu repositório criando um ambiente de desenvolvimento. Para este tutorial, recomendamos que você crie um ambiente de desenvolvimento usando 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** dessa ramificação, mas você pode inserir um nome de ramificação diferente, se preferir.

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. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, escolha Repositórios de origem e escolha o repositório para o qual você deseja criar um ambiente de desenvolvimento.
4. Na página inicial do repositório, escolha Create Dev Environment.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para 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.

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

Em CodeCatalyst, você organiza a criação, o teste e a implantação de seus aplicativos 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 pull request. Para obter mais informações sobre fluxos de trabalho, consulte [Crie, teste e implante com fluxos de trabalho em CodeCatalyst](#).

Siga as instruções em [Introdução aos fluxos de trabalho](#) para criar seu primeiro fluxo de trabalho.

Convide alguém para seu projeto

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

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

Ao terminar de adicionar usuários, escolha Convidar.

Crie problemas para colaborar e monitorar o trabalho

CodeCatalyst ajuda você a monitorar recursos, tarefas, bugs e qualquer outro trabalho envolvido em seu projeto com problemas. Você pode criar problemas para monitorar o trabalho e as ideias necessárias. Por padrão, quando você cria um problema, ele é adicionado à sua lista de pendências. Você pode mover os problemas para um quadro onde você acompanha 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 CodeCatalyst console 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, escolha Amazon e CodeCatalyst, se necessário, escolha Visualizar todos os projetos. Escolha o projeto em que você deseja criar ou trabalhar com problemas.

2. No painel de navegação, escolha Rastrear e, em seguida, escolha Backlog.
3. Escolha Criar problema.
4. Em Título da edição, forneça um nome para a edição. Opcionalmente, 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 a partir da lista de membros do projeto.

Tip

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

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).

5. Selecione Salvar.

Depois de criar problemas, você pode atribuí-los aos membros do projeto, estimá-los e monitorá-los em um quadro Kanban. Para ter 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 suas alterações de código aos colegas de equipe ao criar pull requests 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. 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 o Amazon Q Developer?](#) no Amazon Q Developer User Guide.

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Escrever descrição para mim, Criar resumo de conteúdo e Atribuir problemas ao Amazon Q com o Amazon Q Developer Agent para desenvolvimento de software são criados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a proteção e o uso responsável da inteligência artificial (IA).

Neste tutorial, você aprenderá a usar os recursos generativos de IA CodeCatalyst para ajudá-lo a resumir as mudanças entre as ramificações ao criar pull requests e resumir os comentários deixados

em uma pull request. 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.

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
- Seu projeto está em um espaço que tem recursos generativos de IA ativados. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).
- Você tem a função de colaborador ou administrador do projeto em um projeto nesse espaço.
- O projeto tem pelo menos um repositório de origem configurado para ele. Não há suporte para repositórios vinculados.
- Ao atribuir problemas para ter uma solução inicial criada pela IA generativa, o projeto não pode ser configurado com a extensão do Jira Software. A extensão não é compatível com esse recurso.

Para obter mais informações, consulte [Criando um espaço](#), [Acompanhe e organize o trabalho com problemas no CodeCatalyst](#), [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Concedendo acesso com funções de usuário](#).

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

Crie um resumo das alterações de código entre as ramificações ao criar uma pull request

Uma pull request é a principal forma pela qual você e outros membros do projeto podem revisar, comentar e mesclar alterações de código de uma ramificação para outra. Você pode usar pull requests para revisar as alterações no código de forma colaborativa em busca de pequenas alterações ou correções, adições de recursos importantes ou novas versões do seu software lançado. Resumir as alterações no código e a intenção por trás das mudanças como parte da descrição da pull request 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 suas alterações com detalhes suficientes para que os revisores entendam o que estão revisando ou qual era a intenção por trás das mudanças no código.

Você pode usar o recurso Write description for me ao criar pull requests para que o Amazon Q crie uma descrição das alterações contidas em um pull request. Quando você escolhe essa opção, o Amazon Q analisa as diferenças entre a ramificação de origem que contém as alterações de código e a filial de destino na qual você deseja mesclar essas alterações. Em seguida, ele cria um resumo do que são essas mudanças, bem como sua melhor interpretação da intenção e do efeito dessas mudanças.

Note

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

Você pode experimentar esse recurso com qualquer pull request que você criar, mas neste tutorial, vamos testá-lo fazendo algumas alterações simples no código contido em um projeto criado em um blueprint de aplicativo web moderno de três camadas baseado em Python.

Tip

Se você estiver usando um projeto criado com um blueprint 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 seu projeto em uma ramificação e, em seguida, crie uma pull request 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. Em seguida, você criará uma pull request e usará o recurso Escrever descrição para mim para resumir as alterações feitas.

Para criar uma ramificação (console)

1. No CodeCatalyst console, navegue até o projeto em que seu repositório de origem reside.

2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.
3. Escolha o repositório em que você deseja criar uma ramificação.
4. Na página de visão geral do repositório, escolha Mais e, em seguida, escolha Criar ramificação.
5. Insira um nome para a filial.
6. Escolha uma ramificação a partir da qual criar a ramificação e, em seguida, escolha Criar.

Depois de ter uma ramificação, edite um arquivo nessa ramificação com uma simples alteração. Neste exemplo, você editará o `test_endpoint.py` arquivo para alterar o número de novas tentativas de testes de 1 **3** para 5.

Tip

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

Para editar o `test_endpoint.py` arquivo no console

1. Na página de visão geral do repositório de **mysfits** origem, escolha a ramificação suspensa 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 `test_endpoint.py` arquivo, expanda os testes, expanda o número inteiro e escolhatest_endpoint.py.
3. Selecione a opção Editar.
4. Na linha 7, altere o número de vezes que todos os testes serão repetidos a partir de:

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

para:

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

5. Escolha Confirmar e confirme suas alterações em sua filial.

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

Crie uma pull request com um resumo das alterações

1. Na página de visão geral do repositório, escolha Mais e, em seguida, escolha Criar pull request.
2. Na ramificação de destino, escolha a ramificação na qual mesclar o código depois que ele for revisado.

 Tip

Escolha a ramificação a partir da qual você criou sua ramificação no procedimento anterior para obter 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 para sua pull request.

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

Crie um resumo dos comentários deixados sobre alterações de código em uma pull request

Quando os usuários revisam uma pull request, eles geralmente deixam vários comentários sobre as alterações nessa pull request. Se houver muitos comentários de muitos revisores, pode ser difícil escolher temas comuns no feedback 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 pull request e crie um resumo desses comentários.

Note

Os resumos dos comentários são transitórios. Se você atualizar uma pull request, o resumo desaparecerá. Os resumos de conteúdo não incluem comentários sobre a pull request geral, apenas comentários deixados sobre diferenças no código nas revisões da pull request. Esse recurso não funciona com nenhum comentário deixado sobre alterações de código nos submódulos do Git.

Para criar um resumo dos comentários em uma pull request

1. Navegue até a pull request que você criou no procedimento anterior.

Tip


Se preferir, você pode usar qualquer pull request aberta em seu projeto. Na barra de navegação, escolha Código, escolha Pull requests e escolha qualquer pull request aberta.

2. Adicione alguns comentários à pull request em Changes se a pull request ainda não tiver comentários.
3. Em Visão geral, escolha 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 pull request e compare-o com os comentários na pull request.

Crie um problema e atribua-o 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 generativo de IA chamado Amazon Q, que pode analisar um problema com base no título e na descrição. Se você atribuir o problema à Amazon Q, ela tentará criar um rascunho de solução para você avaliar. Isso pode ajudar você e sua equipe a se concentrarem e otimizarem seu trabalho em questões que exigem sua atenção,

enquanto a 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 simples e problemas simples. Para obter 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 que ela prossiga sem feedback. Se você optar por confirmar cada etapa, poderá responder à Amazon Q com feedback sobre a abordagem que ela cria para que ela possa iterar sua abordagem, se necessário. O Amazon Q também pode revisar o feedback que os usuários deixam sobre qualquer pull request criado se você escolher essa opção. Se você optar por não confirmar cada etapa, o Amazon Q poderá concluir seu trabalho mais rapidamente, mas não analisará nenhum feedback que você fornecer sobre o problema ou em qualquer pull request criado.
- Se você deseja permitir que ele atualize os arquivos do fluxo de trabalho como parte de seu trabalho. Seu projeto pode ter fluxos de trabalho configurados para iniciar execuções em eventos de pull request. Nesse caso, qualquer pull request criada pela Amazon Q que inclua a criação ou atualização do fluxo de trabalho YAML pode iniciar uma execução desses fluxos de trabalho incluídos na pull request. Como prática recomendada, não escolha permitir 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 pull request criada por ele.
- Se você quiser permitir que ele sugira a criação de tarefas para dividir o trabalho na edição em incrementos menores que podem ser atribuídos individualmente aos usuários, incluindo o próprio Amazon Q. Permitir que a 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 a totalidade do trabalho, pois o trabalho necessário para concluir cada tarefa é idealmente mais simples do que o problema ao qual pertence.

- Em qual repositório de origem você deseja que ele funcione. 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 para 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 de seu trabalho. Depois de analisar os dados, a Amazon Q formulará uma possível abordagem para uma solução. O Amazon Q registra suas ações atualizando seu comentário fixo e comentando sobre seu progresso na questão em todas as etapas. Ao contrário dos comentários e respostas fixos, ele não mantém um registro estritamente cronológico de seu trabalho. Em vez disso, ele coloca as informações mais relevantes sobre seu trabalho no nível superior do comentário fixado. Ele tentará criar código com base em sua abordagem e análise do código que já está no repositório. Se gerar com sucesso uma solução potencial, ele criará uma ramificação e enviará o código para essa ramificação. Em seguida, ele cria uma pull request que mesclará essa ramificação com a ramificação padrão. Quando o Amazon Q conclui seu trabalho, ele move o problema para Em análise para que você e sua equipe saibam que há um código pronto para serem avaliados.

Note

Esse recurso só está disponível por meio de edições na região Oeste dos EUA (Oregon). Ela não estará disponível se você tiver configurado seu projeto para usar o Jira com a extensão Jira Software. Além disso, se você personalizou o layout do seu quadro, o problema pode não mudar de estado. Para obter 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 uma edição à Amazon Q, você não pode alterar o título ou a descrição da edição nem atribuí-la a outra pessoa. Se você cancelar a atribuição do Amazon Q ao problema, ele concluirá a etapa atual e, em seguida, deixará de funcionar. Ele não pode retomar o trabalho nem ser reatribuído ao problema depois de não ter sido atribuído.

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 recursos potenciais para o código incluído em projetos criados com o modelo moderno de aplicativo web de três camadas: um para adicionar um para criar uma nova 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.

Para criar um problema e gerar uma solução para você avaliar

1. No painel de navegação, escolha Problemas e verifique se você está na visualização do quadro.
2. Escolha Criar problema.
3. Dê à edição um título que explique o que você quer fazer em linguagem simples. Por exemplo, para esse problema, insira um título de **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 a
```

```
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.
```

- (Opcional) Anexe uma imagem para usar como miniatura e foto do perfil do mysfit à edição. Se você fizer isso, atualize a descrição para incluir detalhes sobre quais imagens você 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 essas imagens anexadas a esta edição 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 deixar comentários para que o Amazon Q atualize seu código para apontar para as imagens que você deseja usar depois de criar uma pull request.

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


- Em Destinatários, escolha Atribuir ao Amazon Q.
- Em Repositório de origem, escolha o repositório de origem que contém o código do projeto.
- Deslize o botão Exigir que o Amazon Q pare após cada etapa e aguarde a revisão de seu seletor de 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 qualquer tarefa criada para ter a opção de fazer com que o Amazon Q altere sua abordagem em 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 seu trabalho, o trabalho poderá prosseguir mais rapidamente porque o Amazon Q não está esperando por seu feedback, mas você não poderá influenciar a direção que o Amazon Q tomará deixando comentários. O Amazon Q também não responderá aos comentários deixados em uma pull request se você escolher essa opção.


- Deixe o seletor Permitir que o Amazon Q modifique arquivos de fluxo de trabalho no estado inativo.

9. Deslize o botão Permitir que o Amazon Q sugira a criação do seletor de tarefas para o estado ativo.
10. Escolha Criar problema. Sua visão muda para o quadro de problemas.
11. Escolha Criar edição para criar outra edição, 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. Escolha Criar edição para criar outra edição, desta vez com o título **Update the `OnPullRequest workflow to include a branch named test in its triggers.`** Opcionalmente, vincule ao fluxo de trabalho na descrição. Atribua esse problema ao Amazon Q, mas desta vez certifique-se de 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 de problemas.

 Tip

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

Depois de criar e atribuir os problemas, eles passarão para Em andamento. O Amazon Q adicionará comentários acompanhando seu progresso no problema em um comentário fixo. Se for capaz de definir uma abordagem para uma solução, atualizará a descrição do problema com uma seção de plano de fundo que contém a análise da base de código e uma seção de abordagem que detalha a abordagem proposta para criar uma solução. Se a Amazon Q conseguir encontrar uma solução para o problema descrito na questão, ela 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 a 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 cria uma pull request para que você possa revisar as alterações de código sugeridas, adiciona um link para essa pull request para o problema e move o problema para Em revisão.

 Important

Você deve sempre revisar todas as alterações de código em uma pull request antes de mesclá-la. A mesclagem de alterações de código feitas pelo Amazon Q, como qualquer outra alteração de código, pode afetar negativamente sua base de código e código de

infraestrutura se o código mesclado não for revisado adequadamente e contiver erros ao mesclar.

Para analisar um problema e um pull request vinculado que contém alterações feitas pela Amazon Q

1. Em Problemas, escolha um problema atribuído ao Amazon Q que esteja em andamento. Analise os comentários para monitorar o progresso do Amazon Q. Se estiver presente, analise o histórico 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 realizar 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 na edição. 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 pull requests 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 em análise. Analise o histórico 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 seu progresso, quaisquer ações que você possa precisar realizar e quaisquer comentários. Em Pull requests, escolha o link para o pull request ao lado do rótulo Abrir para revisar o código.


 Tip

As pull requests geradas para tarefas só aparecem como pull requests vinculadas na visualização da tarefa. Eles não aparecem como pull requests vinculados para o problema.

3. Na pull request, revise as alterações no código. Para ter mais informações, consulte [Analisando uma pull request](#). Deixe comentários sobre o pull request se quiser que o Amazon Q altere qualquer código sugerido. Seja específico ao deixar comentários no Amazon Q para obter melhores resultados.


Por exemplo, ao revisar a pull request 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 à 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 a Amazon Q atualizar o mysfit com ponteiros para a imagem e miniatura para usar no novo mysfit.

 Note

A Amazon Q não responderá a comentários individuais. O Amazon Q só incorporará o feedback deixado nos comentários em pull requests 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, escolha Criar revisão para que o Amazon Q crie uma revisão da pull request que incorpore as alterações solicitadas nos comentários. O progresso no progresso da criação da revisão será relatado pela Amazon Q em Visão geral, não em Alterações. Certifique-se de atualizar seu navegador para ver as atualizações mais recentes da Amazon Q sobre a criação da revisão.

 Note

Somente o usuário que criou o problema pode criar uma revisão da pull request. Você só pode solicitar uma revisão de um pull request. 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 pull request neste projeto de exemplo. Certifique-se de ver um fluxo de trabalho executado com sucesso antes de mesclar a pull request. Você

também pode optar por criar fluxos de trabalho e ambientes adicionais para testar o código antes de mesclá-lo. Para ter mais informações, consulte [Introdução aos fluxos de trabalho](#).

6. Quando estiver satisfeito com a revisão mais recente da pull request, escolha Mesclar.

Limpar 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 quaisquer problemas que não estejam mais sendo resolvidos. Se o Amazon Q tiver concluído seu trabalho em um problema ou não conseguir encontrar uma solução, certifique-se de cancelar a atribuição do Amazon Q para evitar atingir a cota máxima de recursos generativos de IA. Para obter mais informações, consulte [Gerenciamento de recursos e preços generativos de IA](#).
- Mova qualquer problema em que o trabalho esteja concluído para Concluído.
- Se o projeto não for mais necessário, exclua-o.

Tutorial: Criando um aplicativo de pilha completa com esquemas de PDK compostos

CodeCatalyst A Amazon fornece vários modelos diferentes para ajudar você a começar seus projetos rapidamente. Um projeto criado com um blueprint inclui os recursos de que você precisa, incluindo um repositório de origem, exemplo de código-fonte, fluxos de trabalho de CI/CD, relatórios de criação e teste e ferramentas integradas de controle de problemas. No entanto, às vezes você pode querer criar gradualmente um projeto ou adicionar funcionalidade a um projeto existente criado por meio de um blueprint. Você também pode fazer isso com plantas. Este tutorial demonstra como você pode começar com um único blueprint que define uma base e permite armazenar todo o código do seu projeto em um único repositório. A partir daí, você tem a flexibilidade de incorporar recursos e infraestrutura adicionais aplicando outros esquemas sobre o plano inicial, conforme sua conveniência. Por meio desse método básico, você pode atender a requisitos específicos em vários projetos.

Este tutorial mostra como compor vários esquemas do AWS Project Development Kit (AWS PDK) juntos para criar um aplicativo composto por um site do React, a API Smithy e a infraestrutura de CDK de suporte para implantá-lo na AWS. O AWS PDK fornece elementos básicos para padrões

comuns, além de ferramentas de desenvolvimento para gerenciar e criar seus projetos. Para obter mais informações, consulte o [repositório de GitHub origem do AWS PDK](#).

Os seguintes esquemas de PDK foram projetados para serem usados entre si para criar um aplicativo de forma combinável:

- [Monorepo](#) - Cria um projeto de nível raiz que gerencia as interdependências entre projetos dentro do monorepo. O projeto também fornece cache de compilação e visualização de dependências.
- [API Type Safe](#) - cria uma API que pode ser definida no [Smithy](#) ou na [OpenAPI v3](#) e gerencia a geração de código em tempo de construção para permitir que você implemente e interaja com sua API de maneira segura. Fornece uma construção de CDK que gerencia a implantação de sua API no API Gateway e configura a validação automática de entrada.
- Site do [Cloudscape React - Cria um site](#) baseado em React criado usando o [Cloudscape](#) que vem pré-integrado ao Cognito Auth e (opcionalmente) à sua API criada, que fornece a capacidade de chamar sua API com segurança.
- [Infraestrutura](#) - Cria um projeto que configura toda a infraestrutura relacionada ao CDK necessária para implantar seu aplicativo. Ele também vem pré-configurado para gerar um diagrama baseado em seu código CDK toda vez que você cria.
- [DevOps](#)- Cria DevOps fluxos de trabalho compatíveis com construções encontradas no AWS Project Development Kit (AWS PDK).

O tutorial também inclui etapas sobre como visualizar o aplicativo implantado, convidar outros usuários para trabalhar nele e fazer alterações no código com pull requests que são automaticamente criadas e implantadas em recursos na conta conectada da AWS quando a pull request é mesclada.

Quando você cria um projeto composto por blueprints do PDK, seu projeto é criado com os seguintes recursos em um projeto: CodeCatalyst

- Um repositório de origem configurado como um monorepo.
- Um fluxo de trabalho que executa análise estática de código e verificação de licenças, além de criar e implantar o código de amostra sempre que uma alteração é feita na ramificação padrão. Um diagrama de arquitetura é gerado sempre que você faz alterações no código.
- Um quadro de problemas e uma lista de pendências que você pode usar para planejar e monitorar o trabalho.
- Um conjunto de relatórios de teste com relatórios automatizados.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um projeto monorepo](#)
- [Etapa 2: adicionar a API Type Safe ao projeto](#)
- [Etapa 3: Adicionar um site do Cloudscape React para o projeto](#)
- [Etapa 4: gerar a infraestrutura para implantar o aplicativo na nuvem da AWS](#)
- [Etapa 5: configurar um DevOps fluxo de trabalho para implantar seu projeto](#)
- [Etapa 6: confirme o fluxo de trabalho de lançamento e visualize seu site](#)
- [Colabore e repita o projeto PDK](#)

Pré-requisitos

Para criar e atualizar um projeto, você deve ter concluído as tarefas da [Configure e faça login em CodeCatalyst](#) seguinte forma:

- Tenha um AWS Builder ID para fazer login CodeCatalyst.
- Pertença a um espaço e tenha a função de administrador do espaço ou usuário avançado atribuída a você nesse espaço. Para obter mais informações, consulte [Criando um espaço](#), [Concedendo permissões de espaço aos usuários](#) e [Função de administrador de espaço](#).
- Tenha uma conta da AWS associada ao seu espaço e tenha a função do IAM que você criou durante a inscrição. Por exemplo, durante a inscrição, você tem a opção de escolher criar uma função de serviço com uma política de função chamada política de função CodeCatalystWorkflowDevelopmentRole- **SpaceName**. A função terá um nome CodeCatalystWorkflowDevelopmentRole- **spaceName** com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-**spaceName** de serviço](#). Para obter as etapas para criar a função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-**spaceName** função para sua conta e espaço](#).

Etapa 1: criar um projeto monorepo

Comece com o PDK - Monorepo blueprint para criar sua base de código monorepo que atua como base, permitindo que você adicione outros blueprints PDK.

Para criar um projeto usando o blueprint PDK - Monorepo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.
3. No painel do espaço, escolha Criar projeto.
4. Escolha Começar com um blueprint.
5. Escolha o blueprint PDK - Monorepo e, em seguida, escolha Avançar.
6. 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.
7. Em Recursos do projeto, faça o seguinte:
 - a. Em Linguagem de programação primária, escolha uma linguagem na qual você deseja desenvolver o código do seu projeto. Você pode escolher TypeScript entre Java ou Python.
 - b. Escolha a configuração do código
 - c. No campo de entrada de texto do repositório de origem, insira o nome de um repositório de origem, que criará um novo repositório ou selecionará um repositório vinculado existente. O repositório existente deve estar vazio. Para ter mais informações, consulte [Vinculando um repositório de origem](#).
 - d. (Opcional) No menu suspenso Package Manager, escolha um gerenciador de pacotes. Isso só é necessário se você selecionou TypeScript como sua linguagem de programação principal.
8. (Opcional) Para visualizar o código que será gerado com base nas seleções de parâmetros do projeto que você fez, escolha Visualizar código em Gerar visualização do projeto.
9. (Opcional) Escolha Exibir detalhes no cartão do blueprint para ver detalhes específicos sobre o blueprint, como uma visão geral da arquitetura do blueprint, conexões e permissões necessárias e o tipo de recursos que o blueprint cria.
10. Escolha Criar projeto para criar seu projeto monorepo. O projeto de nível raiz criado gerencia as interdependências entre projetos dentro do monorepo, além de fornecer cache de compilação e gerenciamento de dependências.

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

O modelo PDK - Monorepo gera apenas a base do projeto. Para criar um aplicativo viável usando o blueprint, você precisa adicionar outros blueprints de PDK, como Type Safe API, Cloudscape React Website, Infrastructure ou. DevOps Na próxima etapa, você aplicará uma API Type Safe ao projeto.

Etapa 2: adicionar a API Type Safe ao projeto

O blueprint da API PDK - Type Safe permite que você defina uma API usando o Smithy ou o OpenAPI v3. Ele gera pacotes de tempo de execução a partir de sua definição de API, que incluem clientes para interagir com sua API e código do lado do servidor para implementar sua API. O blueprint também gera uma construção de CDK com segurança de tipo para cada operação de API. Você pode aplicar o blueprint a um projeto monorepo PDK existente para adicionar recursos de API ao projeto.

Para aplicar o plano de API PDK - Type Safe

1. No painel de navegação do seu projeto monorepo, escolha Blueprints e, em seguida, escolha Aplicar blueprint.
2. Escolha o blueprint PDK - Type Safe API e, em seguida, escolha Avançar.
3. Em Configurar blueprint, configure os parâmetros do blueprint:
 - Em Linguagem do modelo, escolha o idioma em que o modelo de API está definido.
 - No campo de entrada de texto Namespace, insira um namespace para sua API.
 - No campo de entrada de texto do nome da API, insira um nome para sua API.
 - Em Linguagem CDK, escolha seu idioma preferido para escrever a infraestrutura CDK na qual implantar a API.
 - Escolha o menu suspenso Idiomas do manipulador e, em seguida, escolha os idiomas nos quais você deseja implementar manipuladores para operações de API.
 - Escolha o menu suspenso Formatos de documentação e, em seguida, escolha os formatos que você deseja para gerar a documentação da API.
4. Na guia Alterações no código, revise as alterações propostas. A diferença exibida em uma pull request mostra as alterações em seu projeto no momento em que a pull request foi criada.
5. Quando estiver satisfeito com as alterações propostas que serão feitas quando o blueprint for aplicado, escolha Aplicar blueprint.

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

Note

O blueprint não será aplicado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

6. Na coluna Status, escolha pull request pendente para a linha do blueprint PDK - Type Safe API e, em seguida, escolha o link da pull request aberta.
7. Escolha Mesclar, escolha sua estratégia de mesclagem preferida e, em seguida, escolha Mesclar para incorporar as alterações do blueprint aplicado.

Depois que a pull request é mesclada, uma nova `packages/apis/myjdkapi` pasta é gerada em seu projeto monorepo, que contém todo o código-fonte relacionado à API da sua API Type Safe configurada.

8. No painel de navegação, escolha Blueprints para confirmar se o status da API PDK - Type Safe mostra Atualizado.

Etapa 3: Adicionar um site do Cloudscape React para o projeto


O blueprint do site PDK - Cloudscape React gera um site. Você pode associar um parâmetro opcional (APIs Type Safe) para configurar automaticamente seu site para configurar clientes de tipo seguro autenticados junto com um explorador de API interativo para testar suas várias APIs.

Para aplicar o plano do site PDK - Cloudscape React

1. No painel de navegação do seu projeto monorepo, escolha Blueprints e, em seguida, escolha Aplicar blueprint.
2. Escolha o blueprint do site PDK - Cloudscape React e, em seguida, escolha Avançar.
3. Em Configurar blueprint, configure os parâmetros do blueprint:
 - No campo de entrada de texto Nome do site, insira um nome para seu site.
 - Escolha o menu suspenso Type Safe APIs e, em seguida, escolha os esquemas de API que você deseja integrar ao site. A transmissão de uma API configura clientes autenticados e adiciona as dependências necessárias, o explorador de API e outros recursos.
4. Na guia Alterações no código, revise as alterações propostas. A diferença exibida em uma pull request mostra as alterações em seu projeto no momento em que a pull request foi criada.

5. Quando estiver satisfeito com as alterações propostas que serão feitas quando o blueprint for aplicado, escolha Aplicar blueprint.

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

 Note

O blueprint não será aplicado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

6. Na coluna Status, escolha Pull Request pendente para a linha de blueprint PDK - Cloudscape React Website e, em seguida, escolha o link da pull request aberta.
7. Escolha Mesclar, escolha sua estratégia de mesclagem preferida e, em seguida, escolha Mesclar para incorporar as alterações do blueprint aplicado.

Depois que a pull request é mesclada, uma nova `packages/websites/my-website-name` pasta é gerada em seu projeto monorepo, que contém todo o código-fonte do seu novo site.

8. No painel de navegação, escolha Blueprints para confirmar se o status do site PDK - Cloudscape React aparece atualizado.

Em seguida, você aplicará o PDK — Infrastructure blueprint para gerar a infraestrutura para implantar seu site na nuvem da AWS.


Etapa 4: gerar a infraestrutura para implantar o aplicativo na nuvem da AWS

O PDK - Infrastructure blueprint configura um pacote contendo todo o seu código CDK para implantar seu site e sua API. Ele também fornece geração de diagramas e conformidade com o pacote de mensagens de prototipagem por padrão.

Para aplicar o PDK - Plano de infraestrutura

1. No painel de navegação do seu projeto monorepo, escolha Blueprints e, em seguida, escolha Aplicar blueprint.


2. Escolha o plano PDK - Infraestrutura e, em seguida, escolha Avançar.
3. Em Configurar blueprint, configure os parâmetros do blueprint:
 - Em Linguagem CDK, escolha a linguagem com a qual você deseja desenvolver sua infraestrutura.
 - No campo de entrada de texto Nome da pilha, insira o nome da CloudFormation pilha gerada para seu blueprint.

 Note

Anote o nome dessa pilha na próxima etapa, na qual você definirá um DevOps fluxo de trabalho.

- Escolha o menu suspenso Type Safe APIs e, em seguida, escolha os esquemas de API que você deseja integrar ao site.
 - Escolha o menu suspenso Sites do Cloudscape React TS e, em seguida, escolha os esquemas do site que você deseja implantar em sua infraestrutura (por exemplo, PDK - Site do Cloudscape React).
4. Na guia Alterações no código, revise as alterações propostas. A diferença exibida em uma pull request mostra as alterações em seu projeto no momento em que a pull request foi criada.
 5. Quando estiver satisfeito com as alterações propostas que serão feitas quando o blueprint for aplicado, escolha Aplicar blueprint.

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

 Note

O blueprint não será aplicado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

6. Na coluna Status, escolha Pull Request pendente para a linha PDK - Infrastructure blueprint e, em seguida, escolha o link da pull request aberta.

7. Escolha Mesclar, escolha sua estratégia de mesclagem preferida e, em seguida, escolha Mesclar para incorporar as alterações do blueprint aplicado.

Depois que a pull request é mesclada, uma nova `packages/infra` pasta é gerada em seu projeto monorepo, que contém a infraestrutura que implantará seu projeto na nuvem da AWS.

8. No painel de navegação, escolha Blueprints para confirmar se o Status do PDK - Infraestrutura aparece Atualizado.

Em seguida, você aplicará o PDK - DevOps blueprint para implantar seu aplicativo.

Etapa 5: configurar um DevOps fluxo de trabalho para implantar seu projeto

O PDK - DevOps blueprint gera os DevOps fluxos de trabalho necessários para criar e implantar seu projeto usando a conta e a função da AWS especificadas na configuração.

Para aplicar o PDK - blueprint DevOps

1. No painel de navegação do seu projeto monorepo, escolha Blueprints e, em seguida, escolha Aplicar blueprint.
2. Escolha o PDK - DevOps blueprint e, em seguida, escolha Avançar.
3. Em Configurar blueprint, configure os parâmetros do blueprint:
 - Escolha Bootstrap CDK no ambiente atual.
 - No campo de entrada de texto Nome da pilha, insira o nome da CloudFormation pilha que você deseja implantar. Isso deve corresponder ao nome da pilha configurado [Etapa 4: gerar a infraestrutura para implantar o aplicativo na nuvem da AWS](#) para o PDK - Infrastructure blueprint.
 - Escolha o menu suspenso de conexão da conta da AWS e, em seguida, escolha a conta da AWS que você deseja usar para os recursos. Para ter mais informações, consulte [Adicionando um Conta da AWS a um espaço](#).
 - Escolha o papel a ser usado para implantar seu aplicativo no menu suspenso e, em seguida, escolha o papel do IAM que você deseja usar para implantar seu aplicativo de projeto.

Note

Ao criar uma função do IAM, restrinja a SourceArn à atual ProjectID encontrada nas configurações do projeto. Para ter mais informações, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName*de serviço](#).

- Escolha o menu suspenso Região e, em seguida, escolha a região na qual você deseja implantar seu projeto monorepo. A implantação só funciona em regiões onde existem os serviços necessários da AWS. Para obter mais informações, consulte [Serviços da AWS por região](#).
4. Na guia Alterações no código, revise as alterações propostas. A diferença exibida em uma pull request mostra as alterações em seu projeto no momento em que a pull request foi criada.
 5. Quando estiver satisfeito com as alterações propostas que serão feitas quando o blueprint for aplicado, escolha Aplicar blueprint.

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

Note

O blueprint não será aplicado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

6. Na coluna Status, escolha Pull Request pendente para a linha PDK - Infrastructure blueprint e, em seguida, escolha o link da pull request aberta.
7. Escolha Mesclar, escolha sua estratégia de mesclagem preferida e, em seguida, escolha Mesclar para incorporar as alterações do blueprint aplicado.

Depois que a pull request é mesclada, uma nova `.codecatalyst/workflows` pasta é gerada dentro do seu projeto monorepo.

8. No painel de navegação, escolha Blueprints para confirmar se o Status do PDK - DevOps mostra Atualizado.

Note

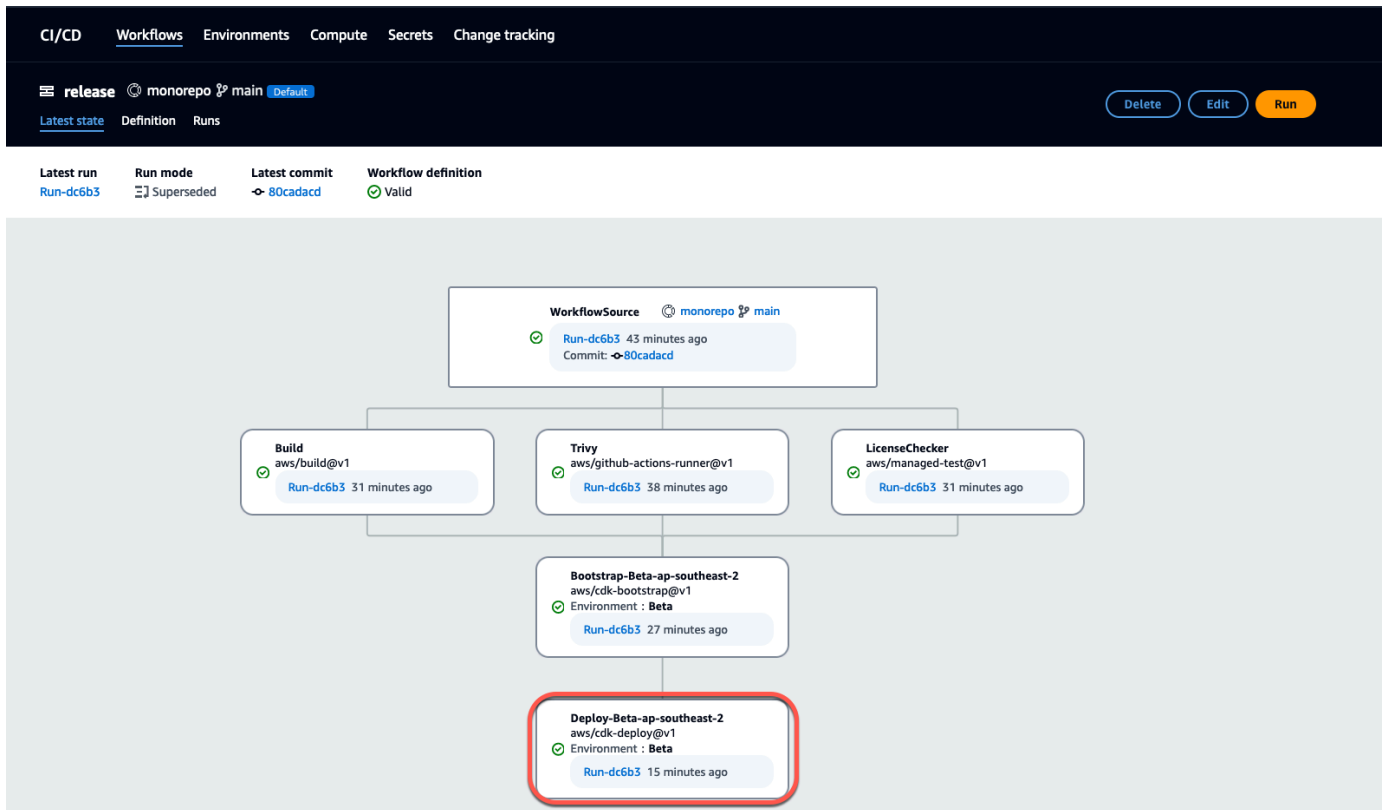
O PDK - DevOps blueprint e todas as alterações subsequentes nos blueprints PDK serão significativamente mais lentos a partir de agora, porque os arquivos de bloqueio nos bastidores são gerados para garantir que as compilações e implantações sejam reproduzíveis no futuro. Ele gerará arquivos de bloqueio para todos os pacotes em qualquer um dos idiomas suportados.

Etapa 6: confirme o fluxo de trabalho de lançamento e visualize seu site

Depois de concluir as etapas anteriores, você pode confirmar o fluxo de trabalho de lançamento para garantir que o projeto esteja sendo construído.

Para confirmar o fluxo de trabalho de lançamento e visualizar seu site

1. No painel de navegação do seu projeto monorepo, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Para o fluxo de trabalho de lançamento, escolha a última execução do fluxo de trabalho para ver os detalhes. Para ter mais informações, consulte [Visualizando o status e os detalhes de uma única execução](#).
3. Depois que a execução do fluxo de trabalho for concluída com êxito, escolha a última ação no fluxo de trabalho (por exemplo, Deploy-B eta-ap-souteast -2) e escolha Variáveis.



4. Visualize o site implantado copiando e colando o link encontrado na tabela Variáveis (por exemplo, **MyPDKAPI** websiteDistributionDomain NameXXXX) em uma nova janela do navegador.

Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds

Logs


Summary

Configuration

Variables

Output variables (7)

🔍 Filter by variable name or value

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityUserPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

Você precisa de uma conta do Amazon Cognito para fazer login no seu site. Por padrão, o grupo de usuários não está configurado para permitir o autorregistro.

- a. Navegue até o console do [AWS Cognito](#).
- b. *Na tabela **Grupos de usuários**, escolha o nome do grupo de usuários que corresponda ao grupo de usuários criado pelo PDK - DevOps blueprint, que pode ser encontrado na tabela **Variáveis** (por exemplo, **infra calcule betaUserIdentityinfracalcule XXXXX**). betaUserIdentity IdentityPoolId* Para obter mais informações, consulte [Introdução aos grupos de usuários](#).

Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds


Logs

Summary

Configuration

Variables

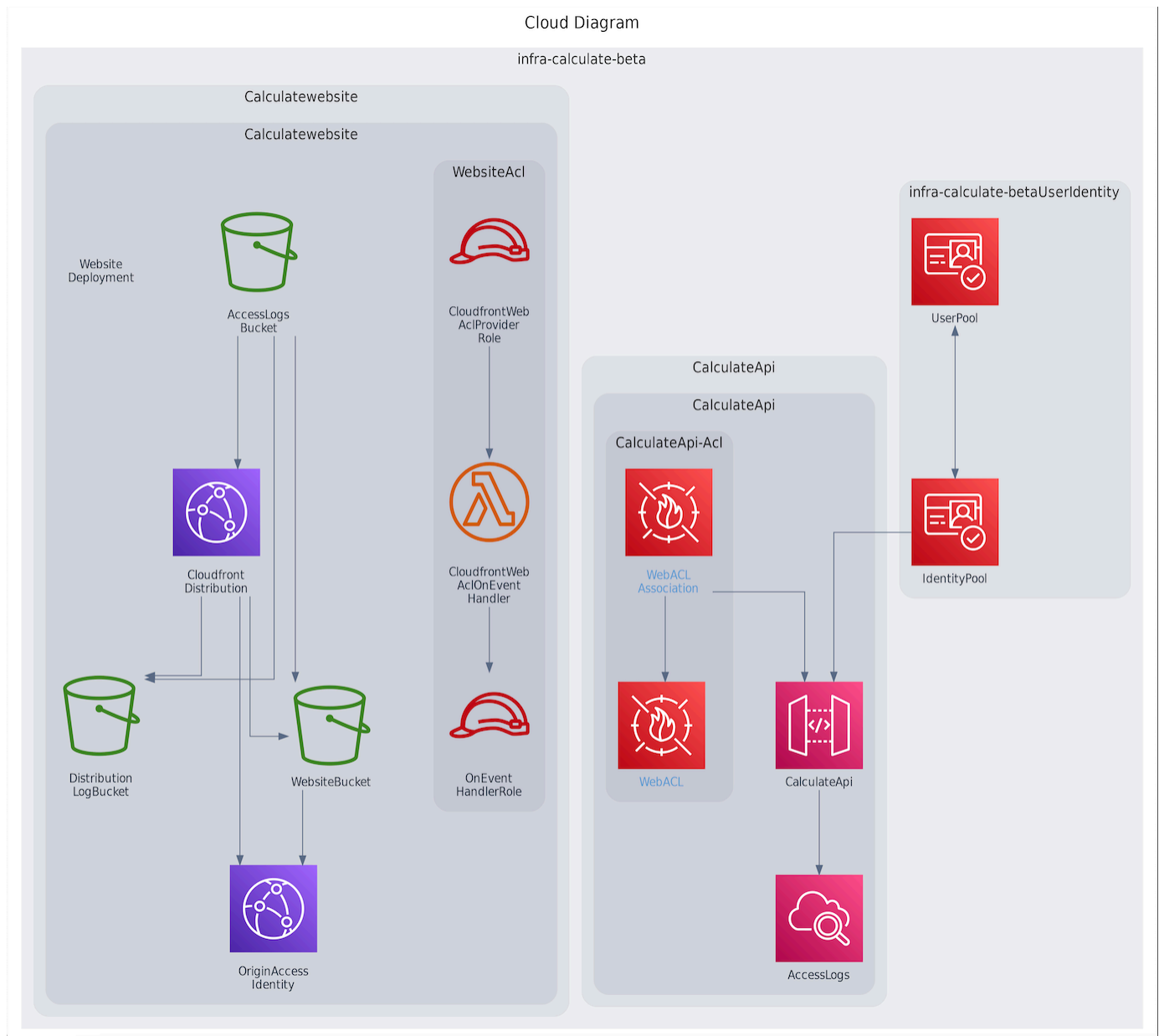
Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

- c. Selecione Criar usuário.
 - d. Configure os parâmetros de informações do usuário:
 - Em Mensagem de convite, escolha Enviar um convite por e-mail.
 - No campo de entrada de texto Nome de usuário, insira um nome de usuário.
 - No campo de entrada de texto do endereço de e-mail, insira um nome de usuário.
 - Em Senha temporária, escolha Gerar uma senha.
 - e. Selecione Criar usuário.
 - f. Navegue até a conta de e-mail que você inseriu para os parâmetros de informações do usuário e abra um e-mail com uma senha temporária. Anote a senha.
 - g. Volte para o site implantado, digite o nome de usuário que você criou e a senha temporária que você recebeu e, em seguida, escolha Entrar.
5. (Opcional) Depois que a execução do fluxo de trabalho for concluída com êxito, você também poderá visualizar o diagrama gerado. Escolha a guia Artefatos em CodeCatalyst, escolha Baixar para a linha Diagrama e abra o (s) arquivo (s) baixado (s).

The screenshot shows the Amazon CodeCatalyst interface for a workflow named 'Run-ef953'. The 'Artifacts' tab is selected, displaying a table of artifacts. The 'Diagram' artifact is highlighted with a red circle. The table has columns for 'Artifact name', 'Files', 'Produced by', and 'Consumed by'. The 'Diagram' artifact has a 'Download' link next to it.

Artifact name	Files	Produced by	Consumed by
Built	Download	Build	Bootstrap-Beta-ap-southeast-2 Deploy-Beta-ap-southeast-2
Diagram	Download	Build	-
bdd254b65baac169f6ac50e8175ce6d930c1fcb086dec59808d3e0170ae2291d_report	Download	Build	-
a093422585a8a4cb763d89a0fa8e76744a80830fe24724c7e7943a50ec479240_report	Download	Trivy	-



Colabore e repita o projeto PDK

Depois que seu projeto estiver configurado, você poderá fazer alterações no código-fonte. Você também pode convidar outros membros do espaço para trabalhar no projeto. Os blueprints do PDK permitem que você crie seu aplicativo de forma iterativa, adicionando somente o que você precisa, quando precisar, mantendo o controle total da configuração de cada blueprint.

Tópicos

- [Etapa 1: convidar membros para seu projeto](#)

- [Etapa 2: criar problemas para colaborar e monitorar o trabalho](#)
- [Etapa 3: Exibir seu repositório de origem](#)
- [Etapa 4: criar um ambiente de desenvolvimento e fazer alterações no código](#)
- [Etapa 5: enviar e mesclar alterações de código](#)

Etapa 1: convidar membros para seu 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 a função de administrador do projeto ou administrador do espaço.

Você não precisa convidar um usuário com a função de administrador do Space 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 a função de administrador do Space), o usuário aparece na tabela de membros do projeto em projetos e na tabela de membros do projeto em espaços.

Para convidar um membro para seu projeto na guia Configurações do projeto

1. Navegue até o projeto.


Tip

Você pode escolher qual projeto 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, escolha Convidar novo membro.
5. Digite o endereço de e-mail do novo membro, escolha a função para esse membro e escolha Convidar. Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).

Para convidar um membro para seu projeto na página de visão geral do projeto

1. Navegue até o projeto.

 Tip

Você pode escolher qual projeto visualizar na barra de navegação superior.


2. Escolha o botão Membros +.
3. Digite o endereço de e-mail do novo membro, escolha a função para esse membro e escolha Convidar. Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).

Etapa 2: criar problemas para colaborar e monitorar o trabalho

CodeCatalyst ajuda você a monitorar recursos, tarefas, bugs e qualquer outro trabalho envolvido em seu projeto com problemas. Você pode criar problemas para monitorar o trabalho e as ideias necessárias. Por padrão, quando você cria um problema, ele é adicionado à sua lista de pendências. Você pode mover os problemas para um quadro onde você acompanha o trabalho em andamento. Você também pode atribuir um problema a um membro específico do projeto. Nesta etapa, crie um problema para fazer alterações em seu projeto PDK.

Para criar um problema

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto monorepo onde você deseja criar um problema.
3. Na página inicial do projeto, escolha Criar problema. Como alternativa, no painel de navegação, escolha Problemas.
4. Escolha Criar problema.

 Note

Você também pode adicionar problemas em linha ao usar uma visualização em grade.

5. Insira um título para a edição.
6. (Opcional) Insira uma descrição. Para esse problema, insira a seguinte descrição: `a change in the src/mysfit_data.json file..` Você pode usar o Markdown para adicionar formatação.
7. (Opcional) Escolha um status, prioridade e estimativa para o problema.

8. (Opcional) Adicione uma etiqueta existente ou crie uma nova etiqueta e adicione-a escolhendo + Adicionar etiqueta.
 - a. Para adicionar uma etiqueta existente, escolha a etiqueta 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 novo rótulo e adicioná-lo, digite o nome do rótulo que você deseja criar no campo de pesquisa e pressione enter.
9. (Opcional) Adicione um destinatário escolhendo + Adicionar um destinatário. Você pode se adicionar rapidamente como destinatário escolhendo + Me adicionar.

 Tip

Você pode escolher atribuir um problema ao Amazon Q para que o Amazon Q tente resolver o problema. Para ter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#). Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).

10. (Opcional) Adicione um campo personalizado existente ou crie um novo 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 novo campo personalizado e adicioná-lo, digite o nome do campo personalizado que você deseja criar no campo de pesquisa e pressione enter. Em seguida, escolha o tipo de campo personalizado que você deseja criar e defina um valor.
11. Escolha Criar problema. Uma notificação aparece no canto inferior direito: se o problema foi criado com sucesso, uma mensagem de confirmação será exibida informando que o problema foi criado com sucesso. Se o problema não tiver sido criado com êxito, uma mensagem de erro com o motivo da falha será exibida. Em seguida, você pode escolher Tentar novamente para editar e tentar criar o problema novamente ou escolher Descartar para descartar o problema. Ambas as opções ignorarão a notificação.

Note

Você não pode vincular uma pull request a um problema ao criá-lo. No entanto, você pode [editá-lo](#) depois de criá-lo para adicionar links às pull requests.

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

Etapa 3: Exibir seu 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 filiais no repositório
- O número de pull requests abertos para o repositório
- O número de fluxos de trabalho relacionados para o repositório
- Os arquivos e pastas na ramificação padrão ou na ramificação que você escolher
- 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 os commits, branches e pull requests 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 ver informações sobre repositórios vinculados, escolha o link na lista de repositórios para abrir esse repositório no serviço que o hospeda.

Para navegar até os repositórios de origem de um projeto

1. Navegue até seu projeto e faça o seguinte:
 - Na página de resumo do seu projeto, escolha o repositório que você deseja na lista e, em seguida, escolha Exibir repositório.
 - No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem. Em Repositórios de origem, escolha 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 pull requests e fluxos de trabalho. Por padrão, o conteúdo da ramificação padrão é exibido. Você pode alterar a exibição escolhendo uma ramificação diferente na lista suspensa.

Tip

Você também pode navegar rapidamente até os repositórios do seu projeto escolhendo Ver código do projeto na página de resumo do projeto.


Etapa 4: criar um ambiente de desenvolvimento e fazer alterações no código

Nesta etapa, crie um ambiente de desenvolvimento e faça alterações no código que serão então mescladas na ramificação principal. Embora este tutorial mostre um projeto simples do AWS PDK, você também pode seguir um exemplo mais complexo fornecido no repositório do [AWS PDK GitHub](#).

Para criar um ambiente de desenvolvimento com uma nova ramificação

1. No painel de navegação do seu projeto monorepo, faça o seguinte:
 - Escolha Visão geral e, em seguida, navegue até a seção Meus ambientes de desenvolvimento.
 - Escolha Código e, em seguida, escolha Dev Environments.
 - Escolha Código, escolha Repositórios de origem e, em seguida, escolha o repositório monorepo para o qual você deseja criar um ambiente de desenvolvimento.
2. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para mais informações.


3. Escolha Clonar um repositório.
4. 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 a partir da página de repositórios de origem ou de um repositório de origem específico, não precisará escolher um repositório. O ambiente de desenvolvimento será criado a partir do repositório de origem que você escolheu na página Repositórios de origem.

5. (Opcional) Em Alias - opcional, insira um alias para o Dev Environment.
6. (Opcional) Escolha o botão de edição da configuração do Dev Environment para editar a configuração de computação, armazenamento ou tempo limite do Dev Environment.
7. (Opcional) Na 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 o AWS Toolkit.

 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 CIDR IPv4 não esteja configurado para o intervalo de 172.16.0.0/12 endereços IP.

8. 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.

Depois que seu ambiente de desenvolvimento estiver em execução, você poderá trabalhar com seu aplicativo de amostra CodeCatalyst gerado fazendo alterações no código com pull requests que são automaticamente criadas e implantadas em recursos na conta conectada da AWS quando a pull

request é mesclada. O monorepo vende um devfile para que todas as dependências e tempos de execução globais necessários estejam presentes automaticamente.

Para alterar o código em seu projeto

1. Em um terminal funcional do seu ambiente de desenvolvimento, navegue até seu projeto monorepo e, em seguida, instale as dependências do projeto executando o seguinte comando:

```
npx projen install
```

2. Navegue até `opackages/apis/myjdkapi/model/src/main/smithy/operations/say-hello.smithy`, que define um exemplo de operação de API. Neste tutorial, você criará uma `Calculate` operação simples que soma dois números. Faça uma alteração no código para definir essa operação, incluindo sua entrada e saída.

Exemplo:

```
$version: "2"
namespace com.aws

@http(method: "POST", uri: "/calculate")
@handler(language: "typescript")
operation Calculate {
  input := {
    @required
    numberA: Integer
    @required
    numberB: Integer
  }
  output := {
    @required
    result: Integer
  }
}
```

A `@handler` característica diz à API Type Safe que você implementará essa operação como um manipulador do AWS Lambda escrito em TypeScript. A API Type Safe gerará um stub para essa operação para você implementar. A `@required` característica foi adicionada, o que significa que ela será aplicada em tempo de execução pelo gateway de API que será implantado. Para obter mais informações, consulte a documentação do [Smithy](#).

3. Renomeie o nome do `/say-hello.smithy` arquivo com um que esteja alinhado com suas alterações de código (por exemplo, `calculate.smithy`)
4. Navegue até `opackages/apis/myjdkapi/model/src/main/smithy/main.smithy`, e faça uma alteração no código para conectar a operação. Você pode expor a `Calculate` operação definida no `/calculate.smithy` listando-a no `operations` campo desse arquivo.

Exemplo:

```
$version: "2"
namespace com.aws

use aws.protocols#restJson1

/// A sample smithy api
@restJson1
service MyPDKApi {
  version: "1.0"
  operations: [Calculate]
  errors: [
    BadRequestError
    NotAuthorizedError
    InternalFailureError
  ]
}
```

5. Crie as alterações executando o seguinte comando:

```
npx projen build
```

Note

Opcionalmente, você pode passar o `--parallel X` sinalizador, que distribuirá a compilação entre os `X` núcleos.

Como a `@handler` característica foi adicionada, os seguintes arquivos são gerados após a conclusão da construção:

- `/packages/apis/myjdkapi/handlers/typescript/src/calculate.ts`
- `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts`

- Navegue até `packages/apis/myjdkapi/handlers/typescript/src/calculate.ts` o e faça alterações no código. Esse arquivo é o manipulador do servidor que é chamado para a API.

```
import {
  calculateHandler,
  CalculateChainedHandlerFunction,
  INTERCEPTORS,
  Response,
  LoggingInterceptor,
} from 'myjdkapi-typescript-runtime';

/**
 * Type-safe handler for the Calculate operation
 */
export const calculate: CalculateChainedHandlerFunction = async (request) => {
  LoggingInterceptor.getLogger(request).info('Start Calculate Operation');

  const { input } = request;

  return Response.success({
    result: input.body.numberA + input.body.numberB,
  });
};

/**
 * Entry point for the AWS Lambda handler for the Calculate operation.
 * The calculateHandler method wraps the type-safe handler and manages marshalling
 * inputs and outputs
 */
export const handler = calculateHandler(...INTERCEPTORS, calculate);
```

- Navegue até o `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts` arquivo e faça alterações no código para atualizar os testes de unidade.

Exemplo:

```
import {
  CalculateChainedRequestInput,
  CalculateResponseContent,
} from 'myjdkapi-typescript-runtime';
```

```
import {
  calculate,
} from '../src/calculate';

// Common request arguments
const requestArguments = {
  chain: undefined as never,
  event: {} as any,
  context: {} as any,
  interceptorContext: {
    logger: {
      info: jest.fn(),
    },
  },
} satisfies Omit<CalculateChainedRequestInput, 'input'>;

describe('Calculate', () => {

  it('should return correct sum', async () => {
    const response = await calculate({
      ...requestArguments,
      input: {
        requestParameters: {},
        body: {
          numberA: 1,
          numberB: 2
        }
      },
    });

    expect(response.statusCode).toBe(200);
    expect((response.body as CalculateResponseContent).result).toEqual(3);
  });
});
```

8. Navegue até o `/packages/infra/main/src/constructs/apis/myjdkapi.ts` arquivo e faça alterações no código para adicionar uma integração para a Calculate operação em sua infraestrutura de CDK. A construção da API tem uma propriedade de integrações, na qual você pode transmitir a implementação adicionada anteriormente. Como você está usando a `@handler` característica em seu modelo Smithy para a Calculate operação, você pode usar a construção `CalculateFunction` CDK gerada, que é pré-configurada, para apontar para a implementação do manipulador.

Exemplo:

```
import { UserIdentity } from "@aws/pdk/identity";
import { Authorizers, Integrations } from "@aws/pdk/type-safe-api";
import { Stack } from "aws-cdk-lib";
import { Cors } from "aws-cdk-lib/aws-apigateway";
import {
  AccountPrincipal,
  AnyPrincipal,
  Effect,
  PolicyDocument,
  PolicyStatement,
} from "aws-cdk-lib/aws-iam";
import { Construct } from "constructs";
import { Api, CalculateFunction } from "calculateapi-typescript-infra";

/**
 * Api construct props.
 */
export interface CalculateApiProps {
  /**
   * Instance of the UserIdentity.
   */
  readonly userIdentity: UserIdentity;
}

/**
 * Infrastructure construct to deploy a Type Safe API.
 */
export class CalculateApi extends Construct {
  /**
   * API instance
   */
  public readonly api: Api;

  constructor(scope: Construct, id: string, props?: CalculateApiProps) {
    super(scope, id);

    this.api = new Api(this, id, {
      defaultAuthorizer: Authorizers.iam(),
      corsOptions: {
        allowOrigins: Cors.ALL_ORIGINS,
        allowMethods: Cors.ALL_METHODS,
      },
    });
  }
}
```

```

    },
    integrations: {
      calculate: {
        integration: Integrations.lambda(new CalculateFunction(this,
"CalculateFunction"))
      }
    },
    policy: new PolicyDocument({
      statements: [
        // Here we grant any AWS credentials from the account that the prototype
is deployed in to call the api.
        // Machine to machine fine-grained access can be defined here using more
specific principals (eg roles or
        // users) and resources (ie which api paths may be invoked by which
principal) if required.
        // If doing so, the cognito identity pool authenticated role must still
be granted access for cognito users to
        // still be granted access to the API.
        new PolicyStatement({
          effect: Effect.ALLOW,
          principals: [new AccountPrincipal(Stack.of(this).account)],
          actions: ["execute-api:Invoke"],
          resources: ["execute-api:/*"],
        }),
        // Open up OPTIONS to allow browsers to make unauthenticated preflight
requests
        new PolicyStatement({
          effect: Effect.ALLOW,
          principals: [new AnyPrincipal()],
          actions: ["execute-api:Invoke"],
          resources: ["execute-api:/*/OPTIONS/*"],
        }),
      ],
    }),
  });

// Grant authenticated users access to invoke the api
props?.userIdentity.identityPool.authenticatedRole.addToPrincipalPolicy(
  new PolicyStatement({
    effect: Effect.ALLOW,
    actions: ["execute-api:Invoke"],
    resources: [this.api.api.arnForExecuteApi("*", "/*", "*")],
  }),
);

```

```
}  
}
```

9. Crie as alterações executando o seguinte comando:

```
npx projen build
```

Depois que a construção do seu projeto for concluída, você poderá visualizar o diagrama gerado atualizado, que pode ser encontrado no `packages/infra/main/cdk.out/cdkgraph/diagram.png`. O diagrama mostra como a função é adicionada e conectada à API criada. À medida que o código CDK é modificado, esse diagrama também é atualizado.

Agora você pode implantar suas alterações enviando-as e mesclando-as na ramificação principal do seu repositório.

Etapa 5: enviar e mesclar alterações de código

Confirme e envie suas alterações de código, que podem então ser mescladas na ramificação principal do seu repositório de origem.

Para enviar alterações à sua ramificação de recursos

- Confirme e envie alterações para sua ramificação de recursos executando os seguintes comandos:

```
git add .
```

```
git commit -m "my commit message"
```

```
git push
```

Fazer alterações aciona a execução de um novo fluxo de trabalho para sua ramificação de recursos, que você pode visualizar no console. CodeCatalyst Em seguida, você pode criar uma pull request para mesclar as alterações na ramificação principal do seu repositório de origem. A fusão da ramificação de recursos com sua ramificação principal aciona o fluxo de trabalho de lançamento. Você também pode vincular a pull request ao seu problema.

Para criar uma pull request e vinculá-la ao seu problema

1. Em seu projeto monorepo, faça o seguinte:
 - No painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha Create pull request.
 - Na página inicial do repositório, escolha Mais e, em seguida, escolha Criar pull request.
 - Na página do projeto, escolha Criar pull request.
2. 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 pull request na página principal do repositório.
3. Na ramificação de destino, escolha a ramificação principal na qual mesclar o código depois que ele for revisado.
4. Na ramificação Source, escolha a ramificação do recurso que contém o código confirmado.
5. Em Título do pull request, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
6. (Opcional) Na descrição da pull request, forneça informações como um link para problemas ou uma descrição de suas 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 ao pull request.

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA na Amazon CodeCatalyst](#).

7. Em Problemas, escolha Vincular problemas e, em seguida, escolha o problema em que você criou [Etapa 2: criar problemas para colaborar e monitorar o trabalho](#). Para desvincular um problema, escolha o ícone de desvinculação.
8. (Opcional) Em Revisores obrigatórios, escolha Adicionar revisores obrigatórios. Escolha na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que o pull request possa ser mesclado na ramificação de destino.

Note

Você não pode adicionar um revisor como revisor obrigatório e como revisor opcional.
Você não pode se adicionar como revisor.

9. (Opcional) Em Revisores opcionais, escolha Adicionar revisores opcionais. Escolha na lista de membros do projeto para adicioná-los. Os revisores opcionais não precisam aprovar as alterações como um requisito para que a pull request possa ser incorporada à ramificação de destino.
10. Seu pull request deve ser revisado e incorporado à ramificação principal pelos revisores ou por você mesmo. Para ter mais informações, consulte [Mesclando uma pull request](#).

Quando suas alterações são mescladas na ramificação principal do seu repositório de origem, um novo fluxo de trabalho é acionado automaticamente.
11. Depois que a mesclagem for concluída, você poderá mover seu problema para Concluído.
 - a. No painel de navegação, escolha Problemas.
 - b. Escolha o problema criado em [Etapa 2: criar problemas para colaborar e monitorar o trabalho](#), escolha a lista suspensa Status e, em seguida, escolha Concluído.

O fluxo de trabalho de lançamento implanta seu aplicativo após uma execução bem-sucedida, para que você possa visualizar as alterações.

Para confirmar o fluxo de trabalho de lançamento e visualizar seu site

1. No painel de navegação do seu projeto monorepo, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Para o fluxo de trabalho de lançamento, escolha a última execução do fluxo de trabalho para ver os detalhes. Para ter mais informações, consulte [Visualizando o status e os detalhes de uma única execução](#).
3. Depois que a execução do fluxo de trabalho for concluída com êxito, escolha a última ação no fluxo de trabalho (Deploy-B eta-ap-southeast -2) e escolha Variáveis.
4. Visualize o site implantado copiando e colando o link da linha **MyPDKAPI** websiteDistributionDomain NameXXXX em uma nova janela do navegador.
5. Insira o nome de usuário e a senha que você criou e escolha Entrar. [Etapa 6: confirme o fluxo de trabalho de lançamento e visualize seu site](#)

6. (Opcional) Teste as alterações em seu aplicativo.
 - a. Escolha o menu suspenso POST.
 - b. Insira dois valores para `numberA` e `number B`, em seguida, escolha Executar.
 - c. Confirme os resultados no corpo da resposta.

Com o tempo, as versões do catálogo das plantas do PDK podem mudar. Você pode alterar os blueprints do seu projeto para as versões do catálogo para se manter atualizado com as alterações mais recentes. Você pode visualizar as alterações no código e os ambientes afetados antes de alterar as versões do blueprint do seu projeto. Para ter mais informações, consulte [Alterando as versões do blueprint em um projeto](#).

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.

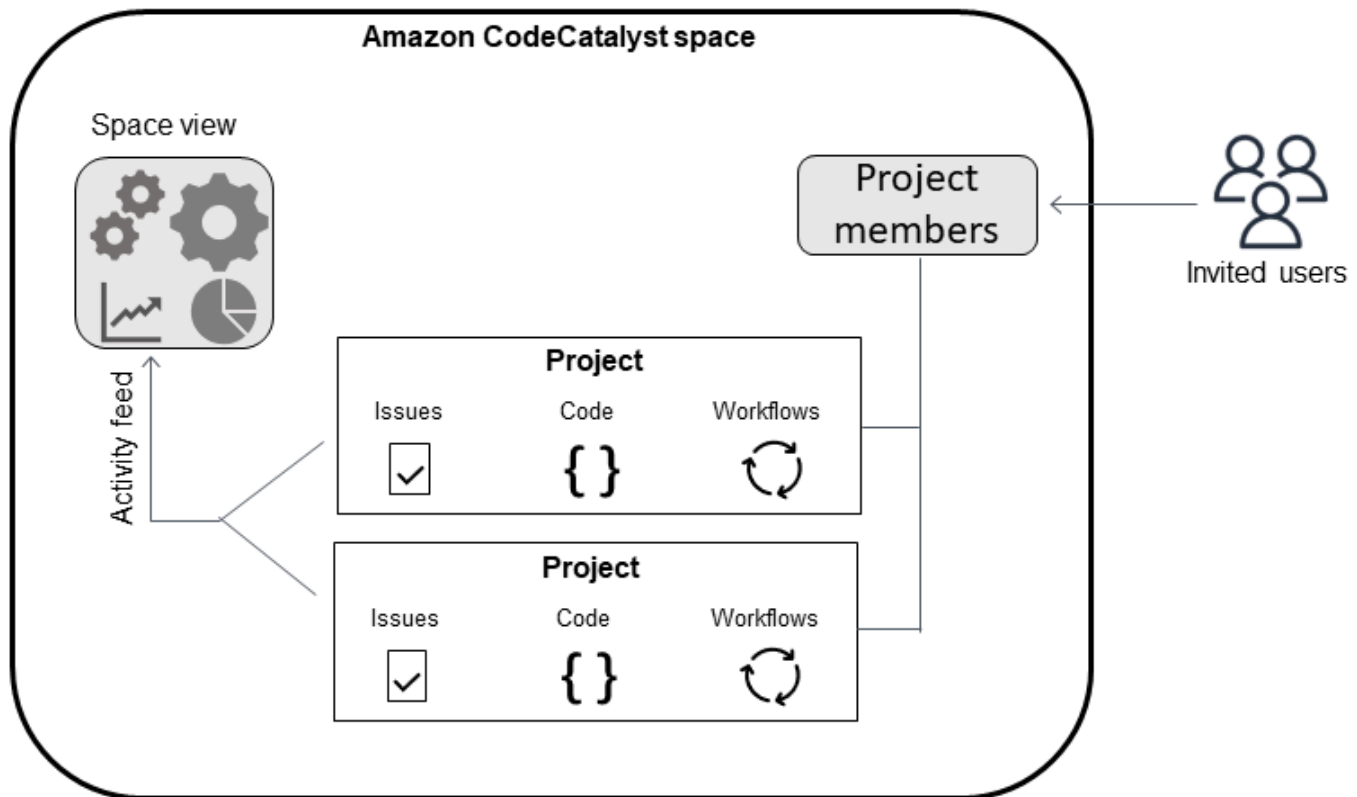
Ao criar um espaço, você recebe automaticamente a função de administrador do espaço. Você pode adicionar essa função a outros usuários no espaço.

Com a função de administrador do espaço, você pode gerenciar o espaço da seguinte maneira:

- Adicione outros administradores de espaço ao espaço
- Alterar funções e permissões dos membros
- Edite ou exclua o espaço
- Crie projetos e convide membros para o projeto
- Veja uma lista de todos os projetos no espaço
- Veja o feed de atividades de todos os projetos no espaço

Ao criar um espaço, você é automaticamente adicionado ao espaço com duas funções: a função de administrador do espaço e a função 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 pela função que o usuário tem em um projeto específico.

Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).



Veja a seguir outras considerações sobre contas adicionadas:

- Há um one-to-one mapeamento da conexão da conta com um espaço. Conta da AWS Um único Conta da AWS pode ser adicionado a vários espaços diferentes. As contas da AWS nas quais você implanta não precisam ser exclusivas e podem ser usadas em mais de um espaço.
- 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. Somente uma Conta da AWS pode ser usada como conta de cobrança para um espaço em CodeCatalyst. Se uma conta já estiver sendo usada para um espaço, você deverá usar uma conta de cobrança diferente para o espaço adicional.
- 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

obter mais informações sobre como os ambientes são usados, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

Tópicos

- [Criando um espaço](#)
- [Editando um espaço](#)
- [Excluindo um espaço](#)
- [Monitoramento de atividades para usuários e recursos em um espaço](#)
- [Permitindo acesso a AWS recursos com conexão Contas da AWS](#)
- [Configurando funções do IAM para contas conectadas](#)
- [Concedendo permissões de espaço aos usuários](#)
- [Permitindo o acesso ao espaço usando equipes](#)
- [Permitindo o acesso ao espaço para os recursos da máquina](#)
- [Administrando ambientes de desenvolvimento para um espaço](#)
- [Cotas para espaços](#)

Criando um espaço


Quando você se inscreve pela primeira vez na Amazon CodeCatalyst com seu ID de AWS construtor, é necessário criar um espaço. Para ter mais informações, consulte [Configure e faça login em CodeCatalyst](#). Você pode optar por criar espaços adicionais para atender às necessidades da sua empresa.

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


Para criar outro espaço

1. No AWS Management Console, 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, escolha um espaço na barra de navegação superior.

4. Escolha Criar espaço.
5. Na página Criar um espaço, em Nome do espaço, insira um nome para o espaço. Você não pode mudar isso 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, escolha a região em que você deseja armazenar seu espaço e dados do projeto. Você não pode mudar isso mais tarde.
7. Em Conta da AWS ID, insira o ID de doze dígitos da conta que você deseja conectar ao seu espaço.

No token de verificação AWS da conta, 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 AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

No AWS Management Console, 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 AWS Management Console 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. Escolha o link a seguir: Para adicionar funções do IAM a esse espaço, veja os detalhes do espaço.

A página de detalhes do CodeCatalyst espaço é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

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 uma função de serviço que contém a política de permissões e a política de confiança para a função 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 AWS CloudFormation. A função terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName` com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

14. Escolha Criar função 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 Ir para a Amazon CodeCatalyst.
17. Na página de criação em CodeCatalyst, escolha Criar espaço.

Editando um espaço

Você pode alterar a descrição de um espaço para ajudar os usuários a entender melhor para que serve.

Você deve ter a função 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, escolha um espaço na barra de navegação superior.

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

Excluindo um espaço

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

Note

Você não pode desfazer uma exclusão de espaço.

Depois de excluir um espaço, todos os membros do espaço não poderão acessar os recursos do espaço. A cobrança por recursos espaciais também será interrompida 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.

Para 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha 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 pertence a um espaço, você é 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 ver 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, escolha um espaço na barra de navegação superior.

3. Escolha Atividade.
4. Veja as informações em Atividade.
5. Para filtrar por atividade, escolha o seletor no canto superior direito.
6. Para ver todas as atividades em seu espaço, escolha 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 funções do IAM às conexões da conta](#)
- [Adicionar a conexão da conta e as funções do IAM ao seu ambiente de implantação](#)
- [Visualizando conexões da conta](#)
- [Removendo uma conta de um espaço \(em CodeCatalyst\)](#)
- [Configurar uma conta de cobrança 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.
- Permita CodeCatalyst que você assuma funções do IAM para acessar AWS recursos e implantá-los Serviços da AWS na conta. Consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

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

Adicionando um Conta da AWS a um espaço

Você usa o CodeCatalyst console e o AWS Management Console 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 a função ou funções do IAM que você deseja associar à conexão da sua conta, incluindo as políticas do IAM com permissões para as funções.
- Adquira a função de administrador do CodeCatalyst Space no espaço em que você deseja criar a conexão.

Tópicos

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

Etapa 1: criar 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 o que Conta da AWS deseja adicionar.

Como criar uma conexão

1. No AWS Management Console, 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 a função ou funções serão aplicadas.
8. Selecione Associar Conta da AWS.
9. A página retorna à página de Conta da AWS detalhes, 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 AWS Management Console com a mesma Conta da AWS para a qual está criando a conexão.

Para aprovar uma solicitação de conexão (console)

1. No AWS Management Console, 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 de Conta da AWS detalhes, escolha Configuração completa no AWS Management Console.
5. A página Verify Amazon CodeCatalyst Space é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

Para acessar diretamente a página, faça login no Amazon CodeCatalyst Spaces AWS Management Console 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, escolha Autorizar níveis pagos (Standard, Enterprise) para ativar os níveis pagos em sua conta de cobrança.

Note

Isso não eleva o nível de cobrança 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 Você pode ativar os níveis pagos a qualquer momento. Sem fazer 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 aprovar uma conexão, você pode ver a conexão no console, junto com as funções do IAM que você adicionou a ela.

Para 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 de Conta da AWS detalhes é exibida.

Etapa 4: adicionar funções do IAM à sua 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 ter mais informações, consulte [Adicionar funções do IAM às conexões da conta](#).

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

Depois de criar uma conexão, você pode criar funções adicionais do IAM para adicionar a ela. As funções do IAM que você adiciona dependem dos seus fluxos de trabalho. Por exemplo, uma ação de CodeCatalyst construção requer a função de CodeCatalyst construção.

Para conectar sua conta, você precisará do Amazon Resource Name (ARN) para as funções que você criou. Copie o ARN para sua função ou funções, conforme detalhado aqui. Para obter mais

informações sobre como trabalhar com ARNs para funções do IAM, consulte [Amazon Resource Name \(ARN\)](#).

Para acessar o ARN da sua função 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 da função que você deseja adicionar.
4. Escolha a função na lista.

A página de resumo da função é exibida.

5. Na parte superior, copie o valor do ARN da função.

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

1. No AWS Management Console, 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 AWS Management Console.

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. A função terá um nome CodeCatalystWorkflowDevelopmentRole-*spaceName* com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

Escolha Criar função de desenvolvimento.

- Para adicionar uma função que você já criou no IAM, escolha Adicionar uma função existente do IAM. Em Selecionar função existente do IAM, escolha a função na lista suspensa.

Escolha Add role (adicionar função).

A página é aberta no AWS Management Console. Talvez seja necessário fazer login para acessar a página.


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

Para acessar diretamente a página, faça login no Amazon CodeCatalyst Spaces AWS Management Console 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 da função que descreva como você deseja usar a função na conexão da sua conta. Especifique RUNNER as funções que você usa para executar ações em fluxos de trabalho. Especifique SERVICE as funções que você usa para acessar outro serviço.

Você pode especificar mais de uma finalidade.

 Note

É necessário escolher uma finalidade para o ARN da função.

10. Escolha Associar uma função do IAM. Repita essas etapas para funções adicionais do IAM.

Adicionar a conexão da conta e as funções 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. Em seguida, você adiciona a função 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 as funções do IAM que deseja usar com ambientes de implantação em seus projetos. Adicionar as funções à conexão da conta não adiciona as funções e a conexão aos ambientes de implantação do projeto. Para adicionar sua conexão de conta e funções do IAM ao seu ambiente de implantação, certifique-se de que a conexão e as funções da conta sejam criadas conforme detalhado em [Etapa 4: adicionar funções do IAM à sua 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 sua conta, conexão e funções 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 as funções do IAM.
3. Expanda CI/CD e escolha Ambientes.
4. Escolha seu ambiente e, em seguida, as guias adicionais serão exibidas.
5. Escolha a guia de Conta da AWS conexões. Em Nome da conexão, as contas que foram adicionadas ao ambiente, se houver, são listadas.

6. Selecione **Associar Conta da AWS**. A <environment_name>página **Associar Conta da AWS** com é exibida.
7. Em **Conexão**, escolha o nome da conexão da conta com as funções do IAM que você deseja adicionar. Selecione **Associar**.

Visualizando conexões da conta

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

Você deve ter a função de administrador do espaço ou usuário avançado para gerenciar as conexões do 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 da conta que você deseja visualizar.
3. Escolha a guia **AWS Contas**.
4. Em **AWS contas**, veja a lista de conexões de conta para o 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 funções do IAM associadas à conexão junto com outros detalhes.

Removendo uma conta de um espaço (em CodeCatalyst)

Você pode excluir uma conexão de conta da qual não precisa mais. Para este 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 cobrança do espaço.

⚠ Important

Depois que uma conexão de conta for excluída, você não poderá reconectá-la. Você deve criar uma nova conexão de conta e, em seguida, associar funções 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 cobrança 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 as conexões de conta do seu espaço, você deve ter a função de administrador do espaço ou usuário avançado.

Para 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 Remove Conta da AWS. Confirme a exclusão inserindo o nome no campo e escolha Remove.

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

Configurar uma conta de cobrança 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. 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, você faz login na página Amazon CodeCatalyst Spaces no AWS Management Console para remover uma Conta da AWS 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 nova conexão entre a conta e o espaço. Você precisará associar novamente todas as funções do IAM à conta adicionada.

Configurando funções 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 funções.

No seu Conta da AWS, você deve ter permissões para criar funções para o que Conta da AWS deseja adicionar ao seu espaço. Para obter mais informações sobre funções e políticas do IAM, incluindo referências e exemplos de políticas do IAM, 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.

Você pode adicionar funções às conexões da sua conta usando um dos métodos a seguir.

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

- Para obter etapas detalhadas para criar funções para ações de fluxo de trabalho, consulte o tutorial do fluxo de trabalho para essa ação da seguinte forma:
 - [Tutorial: Faça upload de artefatos para o Amazon S3](#)
 - [Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation](#)
 - [Tutorial: Implantar um aplicativo no Amazon ECS](#)
 - [Tutorial: código Lint usando uma GitHub ação em um fluxo de trabalho](#)

Tópicos

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

Função da CodeCatalystWorkflowDevelopmentRole-*spaceName*

Você cria a função de desenvolvedor como uma função de 1 clique no IAM. Você deve ter a função 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 AWS Management Console 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 AWS Management Console, 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 de Conta da AWS detalhes é exibida.
5. Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

- Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento. O papel terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

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.

- Escolha Criar função 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.

Função da `AWSRoleForCodeCatalystSupport`

Você cria a função de suporte como uma função de 1 clique no IAM. Você deve ter a função 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 AWS Management Console 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 AWS Management Console, 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 de Conta da AWS detalhes é exibida.
- Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

5. Em detalhes do CodeCatalyst espaço, escolha Adicionar função de CodeCatalyst Support. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento de pré-visualização. A função terá um nome `AWSRoleForCodeCatalystSupport` com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função AWSRoleForCodeCatalystSupport de serviço](#).
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.

Como alternativa, você pode criar uma função de serviço que contenha a política de permissões e a política de confiança da `CodeCatalystWorkflowDevelopmentRole-spaceName` função. Para ter mais informações, consulte [Adicionar funções do IAM às conexões da conta](#).

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Selecione Funções e, em seguida, Criar função.
3. Escolha a 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. Selecione Next (Próximo).
6. Em Adicionar permissões, pesquise e selecione uma política personalizada que você já criou no IAM.
7. Selecione Next (Próximo).
8. Em Nome da função, insira um nome para a função, por exemplo: `codecatalyst-project-role`
9. Selecione Criar função.
10. Copie a função Amazon Resource Name (ARN). Você precisará fornecer essas informações ao adicionar a função à conexão ou ao ambiente da sua conta.

Concedendo permissões de espaço aos usuários

Você pode gerenciar membros de um espaço visualizando, adicionando, removendo ou alterando as funções 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.

Visualizando membros em um espaço

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

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

Você não pode alterar essa função posteriormente sem primeiro remover o usuário. Para ter mais informações, consulte [Função de administrador de 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 ser capazes de criar projetos em um espaço e ajudar a gerenciar os usuários e recursos do espaço. Para ter mais informações, consulte [Função de usuário avançado](#).
- **Acesso limitado** — Essa função é atribuída por padrão aos usuários que ingressam no espaço aceitando convites para projetos no espaço. Os membros do projeto recebem uma função em um projeto. Para obter informações sobre o gerenciamento de membros do projeto, consulte [Conceder permissões de projeto aos usuários](#).

A tabela Administradores do Space mostra os usuários com a função de administrador do Space. Esses usuários não são mostrados nos membros do Espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm uma função em um projeto.

A tabela de membros do Space mostra todos os membros no espaço que têm uma função em um projeto, mas não têm a função de administrador do Space.

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 a função de administrador do Space que posteriormente aceitar um convite e uma função do projeto não aparecerá na tabela de membros do Space em espaços ou na tabela de membros do projeto em Projetos. Eles continuarão sendo mostrados na tabela de administradores do Space em ambos os lugares. Em cada projeto, todos os usuários com a função de administrador do Space são mostrados na tabela de administradores do Space do projeto.

- Um usuário que aceita um convite de projeto para participar com uma função de projeto é adicionado ao espaço com a função de acesso limitado. Se a função do usuário mudar posteriormente para a função 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 funções 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Membros.

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

 Tip

Se você tiver a função de administrador do Space, poderá ver para quais projetos você foi convidado diretamente. Navegue até as configurações do projeto e escolha Meus projetos.

Na coluna Status, os valores a seguir são válidos:

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

Convidar 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-lhe a função de administrador do espaço ou usuário avançado. Atribuir uma dessas funções a outros usuários pode

ajudá-lo a distribuir as responsabilidades de gerenciar o espaço entre mais pessoas sem precisar convidar esses usuários para nenhum projeto.

Note

Você deve ter a função de administrador do espaço ou usuário avançado para convidar membros.

A tabela Administradores do Space mostra os usuários com a função de administrador do Space. Esses usuários não são mostrados na tabela de membros do Space porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm uma função 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 uma função em um projeto.

Para obter mais informações sobre como aceitar um convite e entrar pela primeira vez, consulte [Configure e faça login em CodeCatalyst](#).

Para convidar um usuário para seu espaço

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

Cancelando um convite para um espaço

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

Para gerenciar convites de espaço, você deve ter a função 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Membros.
4. Verifique se o membro tem o status Convidado.

Note

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

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

Alterando a função de um membro do espaço

Você pode alterar a função atribuída a um membro do seu espaço. Você deve ter a função de administrador do espaço para alterar a função de um usuário no espaço.

A tabela Administradores do Space mostra os usuários com a função de administrador do Space. Esses usuários não são mostrados na tabela de membros do espaço porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço.

Para alterar a função 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Membros.
4. Na tabela de membros do Space, escolha o usuário cuja função você deseja alterar. Escolha Alterar função.

Removendo 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 a função de administrador do espaço para remover um membro de um espaço.

A tabela Administradores do Space mostra os usuários com a função de administrador do Space. Esses usuários não são mostrados na tabela de membros do Space porque são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm uma função 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Membros.
4. Escolha o usuário na tabela de membros do projeto. Escolha Remove.

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.

Removendo ou alterando a função de um usuário com a função de administrador do Space

Você pode remover ou alterar a função de um usuário com a função de administrador do espaço no seu espaço.

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

Note

Como usuário com a função de administrador do Space, você não pode se remover. Entre em contato com outro usuário com a função de administrador do Space.

Para remover um usuário com a função de administrador do Space da tabela de membros do Space

Note

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

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o espaço em que você deseja remover ou alterar a função de um usuário com a função de administrador do espaço.
3. Escolha Configurações e, em seguida, escolha 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 a função de administrador do Space, você deve verificar se nenhum convite pendente foi iniciado.

- Escolha a guia Membros. Na tabela Administradores de espaço, escolha o usuário e, em seguida, escolha Remover.

Na caixa de diálogo Remover membro, faça o seguinte.

- Escolha a opção para remover somente a função de administrador do Space do usuário. Escolha Remover.

⚠ Important

Se o usuário não tiver nenhuma outra função atribuída, a alteração da função de administrador do espaço removerá o usuário do espaço.

- Escolha a opção para remover um usuário com a função de administrador do espaço e de todos os seus 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 funções no projeto. Se a função de administrador do espaço for a única função do usuário, o usuário será totalmente removido do espaço.

Permitindo 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 a função de administrador do Space para gerenciar equipes.

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

Tópicos

- [Criando uma equipe](#)

- [Visualizando uma equipe](#)
- [Conceder funções especiais para uma equipe](#)
- [Conceder funções de projeto para uma equipe no nível do espaço](#)
- [Adicionar um usuário diretamente a uma equipe](#)
- [Removendo um usuário de uma equipe diretamente](#)
- [Adicionar um grupo de SSO a uma equipe](#)
- [Excluindo uma equipe](#)

Criando uma equipe

Uma equipe pode ter permissões de função, 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 funções 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 várias funções. Os usuários com várias funções mostrarão um indicador quando tiverem várias funções e serão exibidos primeiro com a função com mais permissões.

Note

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

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

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

Você deve ter a função de administrador do Space para gerenciar equipes.

Para criar uma equipe

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

 Note

O nome da equipe deve ser exclusivo em seu 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. A função será herdada por todos os membros da equipe.
 - Administrador de espaço - Para obter detalhes, consulte [Função de administrador de espaço](#).
 - Acesso limitado - Para obter detalhes, consulte [Função de acesso limitado](#).
 - Usuário avançado - Para obter detalhes, consulte [Função de usuário avançado](#).
6. Em Associação à equipe, escolha uma das opções a seguir para escolher o método para adicionar membros à equipe.
 - Escolha 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.
 - Escolha Usar grupos de SSO para escolher grupos de SSO que você já configurou no IAM Identity Center.

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

Note

Você não pode mudar isso mais tarde. 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. Quando você escolhe gerenciar membros da equipe usando grupos de SSO, a opção de adicionar usuários diretamente não estará mais disponível.

7. Selecione Create (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.

Visualizando uma equipe

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

Para ver uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações e, em seguida, escolha Equipes.
3. Em Função espacial, visualize a função atribuída 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 funções espaciais 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 as funções e permissões dos usuários, dando a uma equipe a capacidade de gerenciar um espaço para os usuários.

Uma equipe pode ter permissões de função, como usuário avançado, em um espaço. Você pode alterar a função de espaço de uma equipe, mas observe que todos os membros da equipe herdarão essas permissões.

Você deve ter a função de administrador do Space para gerenciar equipes.

Alterando a função espacial de uma equipe

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

Conceder funções 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 função será aplicada à equipe e todos os membros da equipe herdarão essas permissões. Você pode escolher uma função para cada projeto que será automaticamente concedida à equipe.

Você deve ter a função de administrador do Space para gerenciar equipes.


Para adicionar ou alterar uma função do projeto

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

3. Escolha a guia Funções do projeto.
4. Para alterar uma função, escolha o seletor ao lado do projeto nessa lista e escolha Alterar função. Para adicionar uma função, escolha Adicionar função do projeto. Em Projeto, escolha o projeto que você deseja adicionar e, em Função, escolha a função. Escolha uma das funções de projeto disponíveis:
 - Administrador do projeto - Para obter detalhes, consulte [Função de administrador do projeto](#).
 - Colaborador - Para obter detalhes, consulte [Função de colaborador](#).
 - Revisor - Para obter detalhes, consulte [Função de revisor](#).
 - Somente leitura - Para obter detalhes, consulte [Função somente para leitura](#).
5. Selecione Save (Salvar).

Para remover uma função do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações e, em seguida, escolha Equipes.
3. Escolha a guia Funções do projeto.
4. Escolha a função que você deseja remover.

 Important

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

5. Escolha Salvar.

Adicionar 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 todas as funções 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 seu 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 [Adicionar um grupo de SSO a uma equipe](#).

Você deve ter a função de administrador do Space para gerenciar equipes.

Para adicionar um usuário diretamente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações e, em seguida, escolha 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 escolha Salvar. Escolha usuários do AWS Builder ID ou usuários de SSO que já estejam configurados no IAM Identity Center.

Removendo 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 a função de administrador do Space para gerenciar equipes.

Para remover um membro da equipe

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

Adicionar um grupo de SSO a uma equipe

Se seu espaço estiver configurado como um espaço com usuários e grupos de SSO gerenciados no IAM Identity Center, você pode 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 [Adicionar um usuário diretamente a uma equipe](#).

Você deve ter a função de administrador do Space 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, escolha Teams. Escolha a guia Grupos de SSO.
3. Escolha os grupos de SSO que você deseja adicionar. Você pode adicionar até cinco grupos de SSO.

Excluindo 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 a função de administrador do Space para gerenciar equipes.

Excluir uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações e, em seguida, escolha Equipes.
3. Em Ações, escolha Excluir equipe. Isso muda o papel de toda a equipe.
4. Escolha Excluir.

Permitindo o acesso ao espaço para os 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 instância do Amazon EC2, mas se refere a um modelo 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 plantas e fluxos de trabalho. Você pode visualizar os recursos da máquina no seu espaço e escolher ativar ou desativar os recursos da máquina para o seu espaço. Por exemplo, talvez você queira desabilitar um recurso da máquina para gerenciar o acesso e reativá-lo posteriormente.

Essas operações estão disponíveis para recursos da máquina nos casos em que um recurso da máquina precisa ser revogado ou desativado. Por exemplo, se você suspeitar que as credenciais

podem ter sido comprometidas, você pode desativar o recurso da máquina. Geralmente, essas operações não precisarão ser usadas.

Você deve ter a função de administrador do espaço para visualizar esta página e gerenciar os recursos da 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 [Permitindo o acesso ao espaço para os recursos da máquina](#).

Tópicos

- [Visualizando o acesso ao espaço para recursos da máquina](#)
- [Desativando o acesso ao espaço para recursos da máquina](#)
- [Habilitando o acesso ao espaço para os recursos da máquina](#)

Visualizando o 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 a função de administrador do Space 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 escolha Configurações. Escolha os recursos da máquina.
3. No menu suspenso, escolha Ação de fluxo de trabalho para visualizar somente os recursos da máquina para fluxos de trabalho. Escolha Blueprint para visualizar somente os recursos da máquina para blueprints.

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

Desativando o 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 blueprints ou fluxos de trabalho associados no espaço.

Você deve ter a função de administrador do Space para gerenciar os recursos da máquina.

Para desativar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu espaço e escolha Configurações. Escolha os recursos da máquina.
3. Escolha uma das seguintes opções.

⚠ Important

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

- Para desativar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja desativar. Escolha Desativar e, em seguida, escolha Este recurso.
- Para desativar todos os recursos, escolha Desativar e, em seguida, escolha Todos os recursos.
- Para desativar todas as ações do fluxo de trabalho, escolha Desativar e, em seguida, escolha Todas as ações do fluxo de trabalho.
- Para desativar todos os blueprints, escolha Desabilitar e, em seguida, escolha Todos os blueprints.

Habilitando o acesso ao espaço para os 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 a função de administrador do Space para gerenciar os recursos da máquina.

Para habilitar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu espaço e escolha Configurações. Escolha os recursos da máquina.
3. Escolha uma das seguintes opções.
 - Para habilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja ativar. Escolha Habilitar e, em seguida, escolha Este recurso.
 - Para habilitar todos os recursos, escolha Habilitar e, em seguida, escolha Todos os recursos.
 - Para habilitar todas as ações do fluxo de trabalho, escolha Habilitar e, em seguida, escolha Todas as ações do fluxo de trabalho.
 - Para ativar todos os blueprints, escolha Habilitar e, em seguida, escolha Todos os blueprints.

Administrando 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 Space 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 de espaço mantêm os ambientes de desenvolvimento no nível do espaço.

Considerações para administrar ambientes de desenvolvimento

- Você deve ter a função de administrador do Space para visualizar a página Dev Environments em Configurações e para gerenciar os Dev Environments 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](#).

Para outras considerações sobre ambientes de desenvolvimento, incluindo interromper 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

- [Visualizando ambientes de desenvolvimento para seu espaço](#)
- [Editando um ambiente de desenvolvimento para seu espaço](#)
- [Interrompendo um ambiente de desenvolvimento para seu espaço](#)
- [Excluindo um ambiente de desenvolvimento para seu espaço](#)

Visualizando 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 obter mais informações sobre como criar e executar ambientes de desenvolvimento, consulte [Criar um ambiente de desenvolvimento](#).

Você deve ter a função de administrador do espaço para visualizar esta página e gerenciar ambientes de desenvolvimento no nível do espaço.

Para ver os 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Dev Environments.

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.

Editando um ambiente de desenvolvimento para seu 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 obter mais informações sobre a edição de um ambiente de desenvolvimento, consulte [Editando um ambiente de desenvolvimento](#).

Você deve ter a função 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Dev Environments.
4. Escolha o seletor ao lado do ambiente de desenvolvimento que você deseja gerenciar. Selecione a opção Editar.
5. Faça as alterações desejadas no tempo limite de computação ou inatividade do ambiente de desenvolvimento.
6. Escolha Salvar.

Interrompendo um ambiente de desenvolvimento para seu espaço

Você pode interromper a execução de um ambiente de desenvolvimento antes que ele fique ocioso se o ambiente de desenvolvimento estiver configurado para ter um tempo limite. Caso contrário, um ambiente de desenvolvimento com um tempo limite expirado já será interrompido. Para obter mais informações sobre como interromper um ambiente de desenvolvimento, consulte [Interromper um ambiente de desenvolvimento](#).

Você deve ter a função de administrador do espaço para visualizar esta página e gerenciar ambientes de desenvolvimento no nível do espaço.

Para interromper um ambiente 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Dev Environments.
4. Escolha o seletor ao lado do ambiente de desenvolvimento que você deseja gerenciar. Escolha Parar.

Excluindo um ambiente de desenvolvimento para seu espaço

Você pode excluir um ambiente de desenvolvimento que não seja mais necessário ou que não tenha mais um proprietário. Para obter mais informações sobre as considerações sobre a exclusão de um ambiente de desenvolvimento, consulte [Excluir um ambiente de desenvolvimento](#)

Você deve ter a função de administrador do espaço para visualizar esta página e gerenciar ambientes de desenvolvimento no nível do espaço.

Para excluir 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, escolha um espaço na barra de navegação superior.

3. Escolha Configurações e, em seguida, escolha Dev Environments.
4. Escolha o seletor ao lado do ambiente de desenvolvimento que você deseja gerenciar. Escolha Excluir. Para confirmar a deleção, digite 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 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	<p>As descrições do espaço são opcionais. Se especificados, eles devem ter entre 0 e 200 caracteres. Eles podem conter qualquer combinação de letras, números, espaços, pontos, sublinhados, vírgulas, traços e os seguintes caracteres especiais:</p> <p>? & \$ % + = / \ ; : \n \t \r</p>
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 devem ter entre 3 e 63 caracteres. Eles também devem começar com um caractere alfanumérico. Os nomes de espaço podem conter qualquer combinação de letras, números, pontos, sublinhados e traços. Eles não podem conter nenhum dos seguintes 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 compartilhados de integração contínua/entrega contínua (CI/CD). Ao criar um projeto, você pode adicionar, atualizar ou remover recursos. Você também pode monitorar o progresso 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 geradas funções padrão para seu projeto, que você atribui aos usuários que você convida para seu projeto.

- Qualquer pessoa atribuída ao projeto com uma função de projeto, como a função de Colaborador, pode acessar os recursos do projeto, como um repositório de origem.
- Qualquer pessoa com a função de administrador do Space, administrador do projeto ou administrador do projeto, pode enviar convites para participar de um projeto.
- Os usuários com a função 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 fluxos de trabalho de CI/CD.

Os fluxos de trabalho são usados para criar, testar e liberar ou atualizar aplicativos como um pipeline de CI/CD. Você pode montar fluxos de trabalho adicionando ações que transferem e funcionam nos artefatos de origem. Quando você executa ações, os recursos de nuvem do seu projeto são usados para fornecer capacidade de computação sob demanda para suas ações de fluxo de trabalho. Você pode configurar mais fluxos de trabalho de CI/CD com base na atividade e na saída que deseja configurar. Por exemplo, você pode criar um fluxo de trabalho somente para ações de compilação e teste, onde você pode 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 seu aplicativo em um ambiente de teste.

Ao criar um projeto, você pode usar um blueprint para criar um projeto que contém código de amostra e cria recursos, ou você 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. Você pode adicionar ou remover recursos manualmente depois de criar um projeto.

Cada projeto rastreia a atividade do projeto 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 obter mais informações sobre como trabalhar com dados de atividades, consulte [Visualizando 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 seu projeto depois de criá-lo. Você deve ter a função de administrador do Space para criar projetos.

Como criar um projeto

Com CodeCatalyst projetos, você pode realizar tarefas de desenvolvimento com fluxos de trabalho e repositórios compartilhados de integração contínua/entrega contínua (CI/CD), gerenciar recursos, rastrear problemas e adicionar usuários.

Antes de criar um projeto, você deve ter a função de administrador do espaço ou usuário avançado.

Tópicos

- [Criando um projeto com um blueprint](#)
- [Criando um projeto vazio na Amazon CodeCatalyst](#)
- [Criação de um projeto com um repositório de terceiros vinculado](#)
- [Adicionar recursos e tarefas aos projetos criados](#)

Criando um projeto com um blueprint

Você pode provisionar todos os recursos do projeto e o código de amostra com um blueprint do projeto. Para obter informações sobre plantas, consulte o [Criando um projeto abrangente com CodeCatalyst plantas](#)

Para criar um projeto com um blueprint


1. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.

2. No painel do espaço, escolha Criar projeto.
3. Escolha Começar com um blueprint.
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. Como alternativa, você pode optar por armazenar o código-fonte do blueprint em um repositório de terceiros. Para ter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Faça o seguinte, dependendo do provedor de repositório terceirizado 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, escolha Bitbucket como provedor de repositório e, em seguida, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Recursos do projeto, configure os parâmetros do blueprint. Dependendo do blueprint, você pode ter a opção de nomear o nome do repositório de origem.
8. (Opcional) Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, escolha Exibir código ou Exibir fluxo de trabalho em Gerar visualização do projeto.

9. (Opcional) Escolha Exibir detalhes no cartão do blueprint para ver detalhes específicos sobre o blueprint, como uma visão geral da arquitetura do blueprint, conexões e permissões necessárias e o tipo de recursos que o blueprint cria.
10. Escolha Criar projeto.

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 a função 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.

Criação de 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 repositório GitHub ou ao Bitbucket. 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 [Criando um espaço](#) e [Convidar um usuário diretamente para um espaço](#).

Para criar um projeto vinculado CodeCatalyst a um repositório de origem em sua GitHub conta ou espaço de trabalho do Bitbucket, você precisará concluir as três tarefas a seguir:

1. Instale os GitHub repositórios ou a extensão Bitbucket. Você é solicitado em um site externo a se conectar e CodeCatalyst fornecer acesso ao seu repositório de terceiros, o que é feito como parte da próxima etapa.


 Important

Para instalar os GitHub repositórios ou a extensão de repositórios do Bitbucket 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 a. CodeCatalyst

 Important

Para conectar sua GitHub conta ou espaço de trabalho do Bitbucket 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.

 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](#).

3. Crie um CodeCatalyst projeto vinculado ao seu GitHub repositório ou repositório Bitbucket.

⚠ 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

ℹ Note

- A só GitHub pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar GitHub repositórios vazios ou arquivados com CodeCatalyst projetos.
- Você não pode vincular um GitHub repositório 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.

Para ter 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. Escolha Traga seu próprio código.
4. Em Vincular repositório existente, escolha GitHub repositórios ou repositórios do Bitbucket, dependendo do provedor de repositórios terceirizado que você deseja usar. Você será solicitado a conectar sua GitHub conta ou espaço de trabalho do Bitbucket. Se a extensão de terceiros ainda não estiver instalada, um prompt de instalação será exibido.
5. Se solicitado, escolha 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 ou espaço de trabalho do Bitbucket ao seu CodeCatalyst espaço.

Para conectar seu provedor de repositório GitHub ou do Bitbucket ao CodeCatalyst

Siga um destes procedimentos, dependendo da extensão de terceiros que você escolheu 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 o aplicativo para o acesso ao repositório que você deseja permitir e escolha 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-se a um espaço de trabalho do Bitbucket.
 1. Escolha o espaço de trabalho Connect Bitbucket para acessar o site externo do Bitbucket.
 2. Faça login no seu espaço de trabalho do Bitbucket usando suas credenciais do Bitbucket.

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 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 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 o aplicativo para o acesso ao espaço de trabalho que você deseja permitir e escolha 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.

Depois de conectar seu provedor de repositórios terceirizado ao 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 ou o espaço de trabalho do Bitbucket que você conectou.
2. Dependendo do provedor de repositório terceirizado que você conectou, escolha o menu suspenso GitHub repositórios ou repositórios do Bitbucket 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 Nomeie seu projeto, insira o nome que você deseja atribuir ao seu projeto. O nome deve ser exclusivo em seu espaço.
4. Escolha Criar projeto.

Depois de instalar os GitHub repositórios ou a extensão de repositórios do Bitbucket, conectar seu provedor de recursos e vincular seu repositório de terceiros ao seu CodeCatalyst projeto, você pode

usá-lo em CodeCatalyst fluxos de trabalho e ambientes de desenvolvimento. Você também pode criar repositórios de terceiros na GitHub conta conectada ou no espaço de trabalho do Bitbucket com código gerado a partir de um blueprint. Para obter mais informações, consulte [Iniciando automaticamente a execução de um fluxo de trabalho após eventos de repositórios de terceiros](#) e [Criar um ambiente de desenvolvimento](#).

Adicionar recursos e tarefas aos projetos criados

Depois que seu projeto estiver pronto, você poderá adicionar recursos e tarefas.

- Para saber mais sobre os fluxos de trabalho de CI/CD criados com seu projeto, consulte. [Introdução aos fluxos de trabalho](#)
- Para trabalhar com ações de construção semelhantes às do seu novo projeto que implantam artefatos de construção em um bucket do Amazon S3, consulte e. [Construindo com fluxos de trabalho Tutorial: Faça upload de artefatos para o Amazon S3](#)
- Para começar com um projeto vazio e trabalhar com a implantação de um aplicativo sem servidor semelhante com uma implantação em AWS CloudFormation pilha, consulte. [Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation](#)
- 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 [Obter uma lista de projetos](#).
- Para ver o código-fonte ou criar uma pull request, 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 [Gerenciando notificações na Amazon CodeCatalyst](#).
- Para convidar membros para seu projeto, consulte [Conceder permissões de projeto aos usuários](#).
- Para configurar ambientes de desenvolvimento, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

Obter 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 a função de administrador do Space para o 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 pull requests abertas, os ambientes de desenvolvimento do projeto e os problemas.
- Nas configurações do projeto, você pode visualizar e gerenciar os detalhes do projeto, excluir o projeto, convidar novos membros para o projeto, gerenciar os membros do projeto e configurar as notificações.

Visualizando 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 Overview (Visão geral).
4. Visualize as tarefas do projeto atribuídas a você e criadas por você.
 - Veja a lista Membros + Exibir tudo para ver uma lista dos membros do projeto.
 - Visualize o cartão Repositórios para ver os repositórios de origem associados ao projeto.
 - Visualize o cartão Execuções do fluxo de trabalho para visualizar os fluxos de trabalho associados ao projeto.
 - Veja o cartão Open pull requests para ver um resumo do status do repositório de código, além das pull requests atribuídas a você e criadas por você.

- Veja o cartão Meus ambientes de desenvolvimento para ver um resumo dos ambientes de desenvolvimento associados ao projeto.
- Visualize o cartão Problemas para ver um resumo das tarefas atribuídas ou das tarefas que você criou.

Visualizando todos os projetos em um espaço

Na lista de projetos do seu espaço, você pode ver todos os projetos nos quais você 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.

Visualizando as configurações do projeto

Nas configurações do projeto, você pode visualizar os membros do projeto, os repositórios de origem, as execuções do fluxo de trabalho, os pull requests abertos, 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 a lista suspensa 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 a função de administrador do espaço ou administrador do projeto para excluir um projeto. Depois de 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 seu 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. Digite **delete** para confirmar a exclusão.
6. Escolha Excluir projeto.

Conceder 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 as funções 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 várias funções. Os usuários com várias funções mostrarão um indicador quando tiverem várias funções e serão exibidos primeiro com a função com mais permissões.

Obter uma lista dos membros e suas funções no projeto

Ao adicionar um usuário ao seu projeto, você atribui um papel que concede permissões ao projeto da seguinte forma:

- A função de administrador do projeto tem todas as permissões em um projeto. Atribua essa função 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 ter mais informações, consulte [Função de administrador do projeto](#).
- A função de Colaborador tem as permissões necessárias para trabalhar em um projeto. Atribua essa função aos usuários que precisam trabalhar com código, fluxos de trabalho, problemas e ações em um projeto. Para ter mais informações, consulte [Função de colaborador](#).
- A função de revisor tem permissões de revisão. Para obter detalhes, consulte [Função de revisor](#).
- A função Somente leitura tem permissões de leitura. Para obter detalhes, consulte [Função somente para leitura](#).

Você não precisa convidar um usuário com a função de administrador do Space 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 a função de administrador do Space), o usuário aparece na tabela de membros do projeto em projetos e na tabela de membros do projeto em espaços.

Para visualizar usuários e funções 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 de membros do projeto mostra todos os membros que têm uma função em um projeto.

 Tip

Se você tiver a função de administrador do Space, poderá ver para quais projetos você foi convidado diretamente. Navegue até as configurações do projeto e escolha Meus projetos.

A tabela Administradores do Space mostra os usuários com a função de administrador do Space. Esses usuários são atribuídos automaticamente (implicitamente) a todos os projetos no espaço e não têm uma função em um projeto.

Na coluna Status, os valores a seguir são válidos:

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

Tópicos

- [Convidar um usuário para um projeto](#)
- [Cancelando um convite](#)
- [Removendo um usuário do seu projeto](#)
- [Aceitar ou recusar um convite para um projeto](#)

Convidar 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 a função de administrador do projeto ou administrador do espaço.

Você não precisa convidar um usuário com a função de administrador do Space 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 a função de administrador do Space), o usuário aparece na tabela de membros do projeto em projetos e na tabela de membros do projeto em espaços.

Para convidar um membro para seu projeto na guia Configurações do projeto

1. Navegue até o projeto.

 Tip

Você pode escolher qual projeto 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, escolha Convidar novo membro.
5. Digite o endereço de e-mail do novo membro, escolha a função para esse membro e escolha Convidar. Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).

Para convidar um membro para seu projeto na página de visão geral do projeto

1. Navegue até o projeto.

 Tip

Você pode escolher qual projeto visualizar na barra de navegação superior.

2. Escolha o botão Membros +.
3. Digite o endereço de e-mail do novo membro, escolha a função para esse membro e escolha Convidar. Para obter mais informações sobre funções, consulte [Concedendo acesso com funções de usuário](#).


Cancelando 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 a função de administrador do projeto ou administrador do espaço.

Para cancelar o convite de um 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. Escolha a opção ao lado da linha com o membro convidado e escolha Cancelar convite.
5. Uma janela de confirmação é exibida. Escolha Cancelar convite para confirmar.

Removendo um usuário do seu projeto

Você pode usar o console para remover um usuário do seu projeto.

Para remover um usuário do seu projeto, você deve estar conectado com a função 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, em seguida, escolha Remover.

6. Confirme que você deseja remover o usuário e escolha Remover.

Aceitar ou recusar 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 de convite.
2. Escolha o link do projeto no e-mail.
3. Escolha Aceitar ou Recusar.

Se você escolher Recusar, um e-mail será enviado para a conta de gerenciamento do projeto notificando que você recusou o convite.

Permitindo o acesso ao projeto usando equipes

Depois de criar um projeto, 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 CodeCatalyst como membros do projeto e do espaço.

Você deve ter a função 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 [Permitindo o acesso ao espaço usando equipes](#).

Tópicos

- [Adicionar uma equipe a um projeto](#)
- [Concessão de funções de projeto para uma equipe](#)
- [Removendo uma função de projeto para uma equipe](#)

Adicionar 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 várias funções. Os usuários com várias funções mostrarão um indicador quando tiverem várias funções e serão exibidos primeiro com a função com mais permissões.

Para adicionar uma equipe

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto. Escolha Configurações do projeto e, em seguida, escolha Teams.
3. Escolha 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 obter detalhes, consulte [Função de administrador do projeto](#).
 - Colaborador — Para obter detalhes, consulte [Função de colaborador](#).
 - Revisor — Para obter detalhes, consulte [Função de revisor](#).
 - Somente leitura — Para obter detalhes, consulte [Função somente para leitura](#).
6. Escolha Adicionar equipe.

Concessão de funções de projeto para uma equipe

Uma equipe pode ter permissões de função, como usuário avançado, em um espaço. Você pode alterar a função de espaço de uma equipe, mas observe que todos os membros da equipe herdarão essas permissões.

Para adicionar ou alterar a função de um projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações do projeto e, em seguida, escolha Teams.
3. Para alterar uma função, escolha o seletor ao lado da equipe nessa lista e escolha Alterar função. Para adicionar uma função, escolha Adicionar função do projeto. Em Projeto, escolha o projeto que você deseja adicionar e, em Função, escolha a função. Escolha uma das funções de projeto disponíveis:
 - Administrador do projeto - Para obter detalhes, consulte [Função de administrador do projeto](#).
 - Colaborador - Para obter detalhes, consulte [Função de colaborador](#).

- Revisor - Para obter detalhes, consulte [Função de revisor](#).
- Somente leitura - Para obter detalhes, consulte [Função somente para leitura](#).

4. Escolha Salvar.

Removendo uma função de projeto para uma equipe

Em CodeCatalyst, você pode ver as funções do projeto para sua equipe. Você também pode ver os membros de uma equipe. Você pode remover a função do projeto de uma equipe.

Para remover uma função do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu espaço. Escolha Configurações do projeto e, em seguida, escolha Teams.
3. Escolha a guia Funções do projeto.
4. Escolha a função que você deseja remover.

Important

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

5. Escolha Salvar.

Permitindo o 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 instância do EC2, 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 blueprint 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 seu projeto, como esquemas e fluxos de trabalho. Você pode visualizar os recursos da máquina em seu projeto e escolher ativar ou desativar os recursos da máquina para seu projeto. Por exemplo, talvez você queira desativar um recurso da máquina para gerenciar o acesso e reativá-lo posteriormente.

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

Você deve ter a função de administrador do espaço ou a função de 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 [Permitindo o acesso ao espaço para os recursos da máquina](#)

Tópicos

- [Visualizando o acesso ao projeto para recursos da máquina](#)
- [Desabilitando o acesso ao projeto para recursos da máquina](#)
- [Habilitando o acesso ao projeto para recursos da máquina](#)

Visualizando 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 a função de administrador do espaço ou a função de administrador do projeto.

Para visualizar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e escolha Configurações do projeto. Escolha Recursos da máquina.

3. No menu suspenso, escolha Ação de fluxo de trabalho para visualizar somente os recursos da máquina para fluxos de trabalho. Escolha Blueprint para visualizar somente os recursos da máquina para blueprints.

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

Desabilitando o 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 blueprints ou fluxos de trabalho associados no espaço.

Você deve ter a função de administrador do espaço ou a função de administrador do projeto.

Para desativar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e escolha Configurações do projeto. Escolha os recursos da máquina.
3. Escolha uma das seguintes opções.

Important

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

- Para desativar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja desativar. Escolha Desativar e, em seguida, escolha Este recurso.
- Para desativar todos os recursos, escolha Desativar e, em seguida, escolha Todos os recursos.
- Para desativar todas as ações do fluxo de trabalho, escolha Desativar e, em seguida, escolha Todas as ações do fluxo de trabalho.

- Para desativar todos os blueprints, escolha Desabilitar e, em seguida, escolha Todos os blueprints.

Habilitando o 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 a função de administrador do espaço ou a função de administrador do projeto.

Para habilitar os recursos da máquina

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto e escolha Configurações do projeto. Escolha os recursos da máquina.
3. Escolha uma das seguintes opções.
 - Para habilitar individualmente, escolha o seletor ao lado de um ou mais recursos da máquina que você deseja ativar. Escolha Habilitar e, em seguida, escolha Este recurso.
 - Para habilitar todos os recursos, escolha Habilitar e, em seguida, escolha Todos os recursos.
 - Para habilitar todas as ações do fluxo de trabalho, escolha Habilitar e, em seguida, escolha Todas as ações do fluxo de trabalho.
 - Para ativar todos os blueprints, escolha Habilitar e, em seguida, escolha Todos os blueprints.

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 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	<p>Os nomes dos projetos devem ser exclusivos em um espaço. Os nomes devem ter entre 3 e 63 caracteres. Os nomes diferenciam letras maiúsculas de minúsculas. Os nomes dos projetos devem começar com um caractere alfanumérico. Caracteres válidos: A-Z, a-z, 0-9, espaços e., _ (sublinhado) - (hífen)</p> <p>Os nomes dos projetos não podem conter nenhum dos seguintes caracteres: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :</p>
Descrições de projetos	<p>As descrições do projeto podem ter até 200 caracteres. Caracteres válidos: A-Z, a-z, 0-9, espaços e., _ (sublinhado) - (hífen). As descrições de projetos são opcionais.</p>

Trabalhando com notificações em 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 seus próprios IDs de membro do Slack para que 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 o mesmo conjunto de eventos sobre os quais os usuários podem escolher serem notificados por e-mail.

Tópicos

- [Como as notificações funcionam?](#)
- [Começando a usar as notificações do Slack](#)
- [Gerenciando notificações na Amazon CodeCatalyst](#)

Como as notificações funcionam?

Você pode configurar seu projeto para fornecer notificações ao aplicativo de mensagens da sua 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 em CodeCatalyst. No entanto, somente usuários com a função de administrador do Space 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 CodeCatalyst.

Sobre quais CodeCatalyst eventos posso configurar notificações?

Você pode configurar CodeCatalyst para enviar notificações a um ou mais canais do Slack sobre eventos de fluxo de trabalho. Depois que as notificações são configuradas entre um CodeCatalyst projeto e o Slack, os usuários do projeto podem escolher adicionar seu próprio ID de membro do Slack para receber mensagens diretas nos canais do Slack sobre eventos. CodeCatalyst Os usuários que adicionarem seus IDs de membro do Slack receberão menções diretas aos seus IDs nos canais do Slack configurados para seus 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 sua ID do AWS Builder.

Como as notificações são exibidas?

Você pode configurar CodeCatalyst para enviar notificações para um ou mais canais do Slack. Você precisa se autorizar a conceder permissões CodeCatalyst para acessar seu espaço de trabalho do Slack. Depois que a autorização for fornecida, CodeCatalyst poderá enviar notificações aos canais

do Slack que você configura. Se um membro do projeto optar por adicionar seu ID de membro do Slack, ele poderá receber menções sobre CodeCatalyst eventos nos canais do Slack configurados para esse projeto.

Como faço para configurar as 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 seu projeto, você deve concluir as seguintes tarefas de alto nível.

Para configurar notificações (tarefas de alto nível)

1. Em CodeCatalyst, você configura uma conexão entre 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 função de administrador do Space podem adicionar ou excluir um workspace do Slack.

2. Em seu projeto em CodeCatalyst, adicione o canal em que você deseja que sua equipe receba notificações.
3. Em 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 obter detalhes das etapas, consulte, [Começando a usar as notificações do Slack](#).

Depois que as notificações forem configuradas entre um CodeCatalyst espaço e o Slack, os usuários podem optar por adicionar seus próprios IDs de membro do Slack para receber mensagens diretas sobre CodeCatalyst eventos nos canais do Slack configurados para seus projetos,

Começando a usar as notificações do Slack

Depois de criar um projeto, você pode configurar as notificações do Slack que ajudam sua 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 [Gerenciando notificações na Amazon CodeCatalyst](#).

Note

O conjunto de eventos do projeto que podem ser enviados aos canais de notificação não é o mesmo conjunto de eventos sobre os quais os usuários podem escolher serem notificados por e-mail. Para obter mais informações, consulte [Gerenciando notificações na Amazon 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 [Configure e faça login em 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 [Concedendo acesso com funções 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 a função de administrador do Space podem adicionar ou excluir espaços de trabalho do Slack. Adicionar ou excluir 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 seguro de autenticação OAuth 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. Em seguida, 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 seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Notificações.
5. Escolha Configurar notificações.
6. Escolha Connect to Slack workspace.
7. Leia o conteúdo da caixa de diálogo e escolha Conectar ao espaço de trabalho do Slack.
8. Na mensagem do AWSChatbot:
 - 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

Para obter o ID do seu canal do Slack

1. Faça login no Slack. Para obter 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 obter mais informações, consulte [Alternar entre espaços de trabalho do Slack ou Entrar 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 você deseja que as notificações sejam enviadas e escolha Abrir detalhes do canal.

O ID do canal é exibido na parte inferior da caixa de diálogo.

4. Copie o valor da 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 seu canal do Slack for privado, adicione o aplicativo AWS Chatbot ao canal da seguinte forma:
 - a. Na caixa de mensagem do seu canal do Slack, insira **@aws** e escolha o aplicativo aws na caixa de diálogo.
 - b. Pressione Enter.

Uma mensagem do Slackbot é exibida indicando que o Chatbot AWS não está no canal privado.
 - c. Escolha Convidar para convidar o AWS Chatbot 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. Recomendamos usar o nome do canal Slack.
5. Escolha Próximo.
6. Em Selecionar eventos de notificação, escolha o tipo de evento para o qual você deseja receber notificações.
7. Escolha Finish.

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 suas notificações no Slack.

Para ver suas 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. Suas notificações mostram o status mais recente de cada execução do fluxo de trabalho e se ele falhou ou foi bem-sucedido.

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 a novos projetos depois de criá-los. Você também pode informar aos usuários do projeto que eles podem configurar notificações pessoais do Slack para seus IDs de membro do Slack e configurar os eventos para os quais receberão e-mails. Para obter mais informações, consulte [Gerenciando notificações na Amazon CodeCatalyst](#).

Gerenciando notificações na Amazon CodeCatalyst

Você pode configurar CodeCatalyst para enviar notificações sobre eventos em seu projeto. Você pode enviar notificações para clientes de mensagens, como canais do Slack. Os usuários do projeto podem escolher sobre quais eventos do projeto serão notificados por e-mails enviados para o endereço de e-mail configurado para seu perfil.

Note

O conjunto de eventos do projeto que podem ser enviados aos canais de notificação não é o mesmo conjunto de eventos sobre os quais os usuários podem escolher serem notificados por e-mail.

Tópicos

- [Gerenciando notificações enviadas diretamente para você](#)
- [Gerenciando notificações enviadas aos canais](#)

Gerenciando notificações enviadas diretamente para você

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 em seu ID do AWS Builder. Por padrão, você receberá e-mails sobre todos os eventos do projeto para os quais os e-mails podem ser enviados.

Se um projeto foi configurado para enviar notificações para um canal do Slack, você pode adicionar seu ID de membro do Slack para receber menções diretas sobre CodeCatalyst eventos nesse canal do Slack. Isso pode ajudar a aumentar sua conscientização sobre os eventos que acontecem nos projetos em que você tem uma função.

Para configurar notificações por e-mail para eventos do projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e, em seguida, 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 Notificações por e-mail, encontre o projeto na lista em que você deseja configurar as notificações por e-mail e escolha Editar.
4. Selecione os eventos para os quais você deseja receber e-mails e escolha Salvar.

Para configurar notificações pessoais do Slack

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Na barra de menu superior, escolha seu selo de perfil e, em seguida, 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. Nas notificações pessoais do Slack, escolha Connect Slack ID e, em seguida, escolha Connect to Slack workspace. Uma janela separada será aberta.

 Tip

Essa opção não é configurável, a menos que um usuário com a função de administrador do Space tenha adicionado um espaço de trabalho do Slack ao seu espaço.

CodeCatalyst Para ter mais informações, consulte [Começando a usar as notificações do Slack](#) e [Gerenciando notificações enviadas aos canais](#).

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. CodeCatalyst Escolha Permitir para permitir o AWS Chatbot acesso ao 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, verifique se ocorreu um erro ao conectar o espaço de trabalho do Slack. Talvez seja necessário rolar para cima para ver o erro.

5. Para parar de receber notificações pessoais do Slack, escolha o espaço de trabalho conectado do Slack e, em seguida, escolha Desconectar ID do Slack.

Gerenciando notificações enviadas aos canais

Você pode optar por adicionar e gerenciar notificações sobre eventos do projeto CodeCatalyst enviadas aos recursos da equipe, como um canal da equipe no Slack. Ao fazer isso, você pode ajudar a garantir que toda a sua 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 a função de administrador do Space podem adicionar ou excluir espaços de trabalho do Slack.

Adicionar um canal de notificação para um projeto

Você pode adicionar um canal no qual deseja receber notificações, como o canal Slack da sua equipe.

Para adicionar um canal do Slack para notificações

1. Se você estiver adicionando seu primeiro canal do Slack, veja em vez disso [Começando a usar as notificações do Slack](#).

Depois de configurar seu primeiro canal, retorne a este procedimento para configurar canais adicionais.

2. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
3. Navegue até seu projeto.
4. No painel de navegação, escolha Configurações do projeto.
5. Escolha a guia Notificações.
6. Escolha Add channel (Adicionar canal).
7. Escolha Escolher espaço de trabalho e, em seguida, 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 em [Começando a usar as 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 mensagens do seu canal do Slack, insira **@aws** e escolha o aplicativo aws no pop-up.
 - b. Pressione Enter.

Uma mensagem do Slackbot é exibida indicando que o Chatbot AWS não está no canal privado.
 - c. Escolha Convidar para convidar o AWS Chatbot para o canal.
9. No campo ID CodeCatalyst do canal, 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 escolha 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. Recomendamos usar o nome do canal Slack.
11. Em Selecionar eventos de notificação, escolha o tipo de evento para o qual você deseja receber notificações.
12. Escolha Add (Adicionar).

Editando notificações para um canal de notificação

Você pode alterar os canais para os quais as notificações vão e desativar completamente as notificações específicas.

Para editar notificações

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto.
3. No painel de navegação, escolha Configurações do projeto.
4. Escolha a guia Notificações.
5. Escolha Editar notificações.
6. Faça um dos seguintes 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, escolha o X no canal.
7. Escolha Salvar.

Removendo um canal

Para remover um canal

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto. No painel de navegação, escolha Configurações do projeto.
3. Na página de configurações do projeto, escolha a guia Notificações.
4. Escolha o indicador ao lado do canal que você deseja remover e escolha Remover canal. Quando solicitado, escolha OK na janela de confirmação.

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 todo o projeto completo com código de amostra. Os blueprints usam um conjunto arbitrário de opções para gerar um conjunto arbitrário de código de saída que é encaminhado para um projeto. À medida que o blueprint é atualizado com as melhores práticas mais recentes ou novas opções, ele pode regenerar as partes relevantes da sua base de código em projetos que contêm esse blueprint.

Você pode usar um CodeCatalyst plano da Amazon para criar um projeto completo com um repositório de origem, amostra de código-fonte, fluxos de trabalho de CI/CD, 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 aplicá-lo a um projeto existente CodeCatalyst . Você pode aplicar vários blueprints em seu projeto, e cada um pode ser aplicado como um componente independente. Por exemplo, você pode ter um projeto que foi criado com um blueprint de aplicativo web e depois aplicar um blueprint de segurança posteriormente. Quando um dos esquemas é atualizado, você pode incorporar as alterações ou correções em seu projeto por meio do gerenciamento do ciclo de vida. Para obter mais informações, consulte [Criando um projeto abrangente com CodeCatalyst plantas](#) e [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#).

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. As plantas personalizadas podem ser desenvolvidas para atender às necessidades específicas dos projetos do seu espaço. Depois de adicionar uma planta personalizada ao catálogo de plantas do seu espaço, você pode gerenciar a planta e continuar fazendo atualizações para que os projetos do seu espaço permaneçam atualizados com as melhores práticas mais recentes. Para ter mais informações, consulte [Padronizando projetos personalizados de projetos em CodeCatalyst](#). [Para ver o SDK de blueprints e exemplos de blueprints, consulte o repositório de código aberto. GitHub](#)

Tópicos

- [Criando um projeto com um blueprint](#)
- [Aplicando um blueprint em um projeto para adicionar recursos](#)

- [Desassociar um blueprint de um projeto para interromper as atualizações](#)
- [Alterando as versões do blueprint em um projeto](#)
- [Editando uma descrição para um blueprint em um projeto](#)
- [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#)
- [Criando um projeto abrangente com CodeCatalyst plantas](#)
- [Padronizando projetos personalizados de projetos em CodeCatalyst](#)
- [Cotas para plantas em CodeCatalyst](#)

Criando um projeto com um blueprint

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. Para obter mais informações, consulte [Criando um projeto com um blueprint](#) e [Criando um projeto abrangente com CodeCatalyst plantas](#).

Depois de criar um projeto, você pode aplicar plantas adicionais ao seu CodeCatalyst projeto a partir do CodeCatalyst catálogo ou do catálogo do seu espaço com plantas personalizadas. Os blueprints representam componentes arquitetônicos, portanto, vários blueprints podem ser usados juntos em seu projeto para incorporar as melhores práticas da sua equipe. Isso também permite garantir que seu projeto esteja atualizado com as alterações mais recentes nos componentes em evolução.

Para saber mais sobre como trabalhar com blueprints em seu projeto, consulte [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#).

Aplicando um blueprint em um projeto para adicionar recursos

Você pode aplicar vários esquemas em um projeto para incorporar componentes funcionais, recursos e governança. Seus projetos podem suportar vários elementos que são gerenciados de forma independente em esquemas separados. A aplicação de esquemas em um projeto reduz a necessidade de criar recursos manualmente e tornar os componentes de software funcionais. Seus projetos também podem se manter atualizados à medida que os requisitos evoluem. Para saber mais sobre a aplicação de esquemas em seu projeto, consulte [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#).

Ao configurar os detalhes de um blueprint, você também pode optar por armazenar o código-fonte do blueprint em um repositório terceirizado preferido, onde você ainda pode gerenciar o blueprint e

utilizar os recursos de gerenciamento do ciclo de vida para manter seu projeto atualizado. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#).


Para aplicar um blueprint ao 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 aplicar um blueprint.
3. No painel de navegação, escolha Blueprints e, em seguida, escolha Aplicar blueprint.
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 blueprint, escolha uma versão do blueprint no menu suspenso Versão do Target. 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. Como alternativa, você pode optar por armazenar o código-fonte do blueprint em um repositório de terceiros. Para ter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Siga um destes procedimentos, dependendo do provedor de repositório terceirizado 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](#).

- Bitbucket: Conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, escolha Bitbucket como provedor de repositório e, em seguida, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Configurar blueprint, configure os parâmetros do blueprint. Dependendo do blueprint, você pode ter a opção de nomear o repositório de origem.
8. Analise as diferenças entre a versão atual do blueprint e sua versão atualizada. A diferença exibida em uma pull request mostra as alterações entre a versão atual e a versão mais recente, que é a versão desejada no momento em que a pull request foi criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão tanto para a versão atual quanto para a versão desejada.
9. Quando você estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Aplicar blueprint. Depois de criar a pull request, você pode adicionar comentários. Os comentários podem ser adicionados à pull request ou a linhas individuais em arquivos, bem como à pull request geral. Você pode adicionar links a recursos, como arquivos, usando o @ sinal seguido pelo nome do arquivo.

Note

O blueprint não será aplicado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

Os autores do blueprint também podem aplicar um blueprint personalizado a projetos em espaços específicos que não tenham o blueprint disponível para criar novos projetos ou aplicar a projetos existentes. Para ter mais informações, consulte [Publicar e aplicar um blueprint personalizado em espaços e projetos especificados](#).

Se você não quiser mais receber atualizações para um blueprint, você pode desassociar o blueprint do seu projeto. Para ter mais informações, consulte [Desassociar um blueprint de um projeto para interromper as atualizações](#).

Desassociar um blueprint de um projeto para interromper as atualizações

Se você não quiser novas atualizações de um blueprint, você pode desassociar o blueprint do seu projeto. Os recursos e componentes funcionais de software adicionados ao seu projeto a partir do blueprint permanecerão em seu projeto.

Para desassociar um blueprint 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 blueprint com os recursos que você deseja desassociar, escolha o menu suspenso Ações e, em seguida, escolha Dissociar blueprint.
5. Entre `confirm` para confirmar a dissociação.
6. Selecione a opção Confirmar.


Alterando as versões do blueprint em um projeto

Se você criou um projeto com um blueprint ou aplicou um blueprint a um projeto existente, você será notificado sobre novas versões do blueprint. Antes que a versão do blueprint seja atualizada por meio de uma pull request aprovada, você pode ver as alterações no código e os ambientes afetados. O gerenciamento do ciclo de vida permite que você altere as versões de um ou mais blueprints aplicados em seu projeto, para que cada versão do blueprint possa ser alterada sem afetar outras áreas do seu projeto. Você também pode substituir as atualizações do blueprint. Para ter mais informações, consulte [Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint](#).


Para atualizar um blueprint 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 blueprint que você 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 escolha Atualizar versão. A versão mais recente é selecionada automaticamente.
7. (Opcional) Em Configurar blueprint, configure os parâmetros do blueprint.
8. (Opcional) Na guia Alterações de código, analise as diferenças entre a versão atual do blueprint e a versão atualizada. A diferença exibida em uma pull request são as alterações entre a versão atual e a versão mais recente, que é a versão desejada no momento em que a pull request é criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão para a versão atual e a versão desejada.
9. Quando estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Aplicar atualização. Depois de criar a pull request, você pode adicionar comentários. Os comentários podem ser adicionados à pull request ou a linhas individuais em arquivos e à pull request geral. Você pode adicionar links para recursos, como arquivos, usando o @ sinal seguido pelo nome do arquivo.

 Note

O blueprint não será atualizado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

 Note

Se você tiver pull requests existentes abertas para atualizar uma versão do blueprint, feche as pull requests anteriores antes de criar uma nova. Ao escolher Atualizar versão, você será direcionado para a lista de pull requests pendentes para o blueprint. Você também pode visualizar pull requests pendentes na guia Blueprints nas Configurações do projeto e na página de resumo do projeto. Para ter mais informações, consulte [Visualizando pull requests](#).

Para alterar a versão de um blueprint

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 blueprint que você deseja atualizar.
4. No painel de navegação, escolha Blueprints e, em seguida, escolha o botão de rádio do blueprint que você deseja atualizar.
5. Escolha o menu suspenso Ações e, em seguida, escolha Configurar blueprint.
6. No menu suspenso Versão do Target, escolha a versão que você deseja usar. A versão mais recente é selecionada automaticamente.
7. (Opcional) Em Configurar blueprint, configure os parâmetros do blueprint.
8. (Opcional) Na guia Alterações de código, analise as diferenças entre a versão atual do blueprint e a versão atualizada. A diferença exibida em uma pull request são as alterações entre a versão atual e a versão mais recente, que é a versão desejada no momento em que a pull request é criada. Se nenhuma alteração for exibida, as versões podem ser idênticas ou você pode ter escolhido a mesma versão para a versão atual e a versão desejada.
9. Quando estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Aplicar atualização. Depois de criar a pull request, você pode adicionar comentários. Os comentários podem ser adicionados à pull request ou a linhas individuais em arquivos e à pull request geral. Você pode adicionar links para recursos, como arquivos, usando o @ sinal seguido pelo nome do arquivo.

Note

O blueprint não será atualizado até que a pull request seja aprovada e mesclada. Para obter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

Note

Se você tiver pull requests existentes abertas para atualizar uma versão do blueprint, feche as pull requests anteriores antes de criar uma nova. Ao escolher Atualizar versão, você será direcionado para a lista de pull requests pendentes para o blueprint. Você também pode visualizar pull requests pendentes na guia Blueprints nas Configurações do projeto e na página de resumo do projeto. Para ter mais informações, consulte [Visualizando pull requests](#).

Editando uma descrição para um blueprint em um projeto

Você pode editar a descrição de um blueprint usado para criar um projeto ou aplicativo após a criação de um projeto. Um blueprint pode ser usado mais de uma vez em um projeto. Para diferenciar a finalidade dos blueprints em seu projeto, você pode usar descrições para esses blueprints. As descrições também podem ser usadas para identificar os componentes que você está aplicando a partir de um blueprint específico.

Para editar a descrição de um blueprint 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. Escolha o blueprint com a descrição que você deseja atualizar, escolha o menu suspenso Ações e, em seguida, escolha Editar descrição.
5. No campo de entrada de texto da descrição do blueprint, insira uma descrição para identificar o blueprint em seu projeto.
6. Escolha Salvar.

Trabalhando com o gerenciamento do ciclo de vida como usuário do blueprint

O gerenciamento do ciclo de vida é a capacidade de regenerar uma base de código a partir de opções ou versões atualizadas de um blueprint. Isso permite que um autor do blueprint gerencie centralmente o ciclo de vida de desenvolvimento de software de cada projeto que contém um blueprint específico. Por exemplo, enviar uma correção de segurança para um blueprint de aplicativo web permitirá que cada projeto contendo ou criado a partir do blueprint de aplicativo web receba essa correção automaticamente. Essa mesma estrutura de gerenciamento também permite que você, como usuário do blueprint, altere as opções do blueprint após elas terem sido selecionadas.

Tópicos

- [Usando o gerenciamento do ciclo de vida em projetos existentes](#)
- [Usando o gerenciamento do ciclo de vida em vários esquemas em um projeto](#)
- [Trabalhando com conflitos em pull requests do ciclo de vida](#)

- [Optando por não participar das mudanças no gerenciamento do ciclo de vida](#)
- [Substituindo o gerenciamento do ciclo de vida de um blueprint em um projeto](#)

Usando o gerenciamento do ciclo de vida em projetos existentes

Você pode usar o gerenciamento do ciclo de vida para projetos criados a partir de blueprints ou em projetos existentes não associados a nenhum blueprint. 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 blueprint gera um fluxo de trabalho de verificação de segurança e outros códigos relacionados. Essa parte da base de código no aplicativo Java agora será atualizada automaticamente com as melhores práticas da sua equipe sempre que forem feitas alterações no blueprint.

Usando o gerenciamento do ciclo de vida em vários esquemas em um projeto

Como os blueprints representam componentes arquitetônicos, vários blueprints geralmente podem ser usados juntos no mesmo projeto. Por exemplo, um projeto pode ser composto por um plano central de API da web criado por um engenheiro de plataforma da empresa, junto com um plano de verificação de lançamento criado pela equipe de segurança do aplicativo. 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 projetos fazem sentido juntos ou funcionarão logicamente juntos, mesmo que ainda tentem se fundir.

Trabalhando com conflitos em pull requests do ciclo de vida

Ocasionalmente, pull requests do ciclo de vida podem gerar conflitos de mesclagem. Eles podem ser resolvidos manualmente. As resoluções são lembradas nas atualizações subsequentes do blueprint.

Optando por não participar das mudanças no gerenciamento do ciclo de vida

Os usuários podem remover um blueprint de um projeto para dissociar todas as referências ao blueprint 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 blueprint. Para ter mais informações, consulte [Desassociar um blueprint de um projeto para interromper as atualizações](#).

Substituindo o gerenciamento do ciclo de vida de um blueprint 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 blueprint 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/**
```

Criando 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, fluxos de trabalho de CI/CD, relatórios de criação e teste e ferramentas integradas de controle de problemas. Um plano de projeto usa código para provisionar infraestrutura de nuvem, recursos e exemplos de artefatos de origem para diferentes tipos de aplicativos e estruturas.

Para ter mais informações, consulte [Como criar um projeto](#). Você deve ser o administrador do Space para criar um projeto.

Tópicos

- [Plantas disponíveis](#)
- [Encontrando informações sobre o plano do projeto](#)

Plantas disponíveis

Nome do blueprint	Descrição do projeto
API web do ASP.NET Core	Esse esquema cria um aplicativo de API web ASP.NET Core do.NET 6. 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 (CSVs) em Apache Parquet.

Nome do blueprint	Descrição do projeto
DevOps pipeline de implantação	Esse esquema cria um pipeline de implantação usando a Arquitetura de Referência do AWS Pipeline de Implantação, que implanta um aplicativo de referência AWS em vários estágios.
API Java com AWS Fargate	Esse blueprint cria um projeto de serviço web em contêineres. O projeto usa o AWS Copilot CLI para criar e implantar um serviço web Spring Boot Java 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 O aplicativo armazena dados em uma tabela do DynamoDB. Depois que seu fluxo de trabalho for executado com êxito, o serviço web de amostra estará disponível publicamente por meio do Application Load Balancer.
Aplicativo web moderno de três camadas	Esse esquema gera código em Python para a camada do aplicativo e a estrutura front-end do Vue para criar e implantar um aplicativo web moderno de 3 camadas bem arquitetado.
Aplicativo sem servidor do.NET	Esse esquema 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 blueprint	Descrição do projeto
API Node.js com AWS Fargate	Esse blueprint cria um projeto de serviço web em contêineres. O projeto usa o AWS Copilot CLI 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 seu fluxo de trabalho for executado com êxito, o serviço web de amostra estará disponível publicamente por meio do Application Load Balancer.
Modelo de aplicativo sem servidor (SAM)	Esse blueprint cria um projeto que usa um modelo de aplicativo sem servidor (SAM) para criar e implantar uma API. Você pode escolher SDK for Java TypeScript ou SDK for Python como linguagem de programação.
Manipulador de imagens sem servidor	Esse plano cria um aplicativo para processamento de imagem em alta velocidade sem reduzir a qualidade da imagem.
Microserviço RESTful sem servidor	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.
Aplicativo de página única	Esse esquema cria um aplicativo 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 blueprint	Descrição do projeto
Site estático	Esse plano 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 +.
Aplicativo web To Do	Esse esquema cria um aplicativo web sem servidor To Do com componentes de front-end e back-end. Você pode escolher SDK for Java TypeScript ou SDK for Python como linguagem de programação.
Serviço video-on-demand web V	Esse plano cria um video-on-demand serviço que fornece a capacidade de receber, transcodificar e entregar conteúdo. O blueprint usa AWS Lambda, Amazon S3 Amazon CloudWatch, e AWS Elemental MediaConvert

Nome do blueprint	Descrição do projeto
Inscrever-se em um plano externo	<p>Esse blueprint 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 modelo. O pacote de destino deve estar no NPM e o pacote deve ser um esquema. O espaço deve ser inscrito em um nível que ofereça suporte a esquemas personalizados.</p>
Chatbot Bedrock GenAI	<p>Esse plano cria um chatbot de IA generativa com o Amazon Bedrock e o Claude da Anthropic. Com esse plano, você pode criar e implantar seu próprio playground LLM seguro e protegido por login, que pode ser personalizado de acordo com seus dados. Para obter mais informações, consulte a documentação do Bedrock GenAI Chatbot.</p>
Planos do AWS Project Development Kit (AWS PDK)	<p>Esses planos de PDK podem ser compostos juntos para criar um aplicativo composto por um site do React, a API Smithy e a infraestrutura de CDK de suporte para implantá-lo na AWS. O AWS PDK fornece elementos básicos para padrões comuns, além de ferramentas de desenvolvimento para gerenciar e criar seus projetos. Para obter mais informações, consulte o repositório de GitHub origem do AWS PDK e Tutorial: Criando um aplicativo de pilha completa com esquemas de PDK compostos</p>

Encontrando informações sobre o plano do projeto

Vários planos 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 blueprint, enquanto o arquivo README explica o blueprint em detalhes e fornece instruções sobre como usá-lo.

Padronizando projetos personalizados de projetos 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 blueprint personalizado é usado para criar um novo projeto ou aplicado a projetos existentes, todas as alterações no blueprint ficam disponíveis para esses projetos como atualizações de pull request. Como autor do blueprint, você pode ver detalhes sobre quais projetos estão usando seus blueprints em todo o seu espaço, para que você possa ver como os padrões estão sendo aplicados em todos os projetos. O gerenciamento do ciclo de vida de um blueprint permite que você gerencie centralmente o ciclo de vida de desenvolvimento de software de cada projeto, permitindo que você garanta que os projetos em seu espaço continuem seguindo as melhores práticas com as alterações ou correções mais recentes. Para ter mais informações, consulte [Trabalhando com o gerenciamento do ciclo de vida como autor do plano](#).

Os blueprints personalizados oferecem a capacidade de atualizar as versões do blueprint em relação ao projeto anterior por meio de ressíntese. A ressíntese é o processo de reexecutar a síntese do blueprint com versões atualizadas ou a capacidade de incorporar correções e alterações em projetos existentes. Para ter mais informações, consulte [Conceitos de plantas personalizadas](#).

[Para ver o SDK de blueprints e exemplos de blueprints, consulte o repositório de código aberto GitHub](#)

Tópicos

- [Conceitos de plantas personalizadas](#)
- [Introdução aos esquemas personalizados](#)
- [Tutorial: Criando e atualizando um aplicativo React](#)
- [Trabalhando com o gerenciamento do ciclo de vida como autor do plano](#)

- [Desenvolvendo um plano personalizado para atender aos requisitos do projeto](#)
- [Publicar um blueprint personalizado em um espaço](#)
- [Visualizando detalhes, versões e projetos de um blueprint personalizado](#)
- [Adicionar um blueprint personalizado a um catálogo espacial](#)
- [Removendo um blueprint personalizado de um catálogo espacial](#)
- [Definindo permissões de publicação para um blueprint personalizado](#)
- [Alterando as versões do catálogo para um blueprint personalizado](#)
- [Excluindo um blueprint ou uma versão personalizada publicada](#)
- [Lidando com dependências, incompatibilidades e ferramentas](#)
- [Contribua](#)

Conceitos de plantas personalizadas

Aqui estão alguns conceitos e termos que você deve conhecer ao trabalhar com esquemas personalizados em CodeCatalyst.

Tópicos

- [Projeto Blueprint](#)
- [Plantas espaciais](#)
- [Catálogo de plantas espaciais](#)
- [Síntese](#)
- [Ressíntese](#)
- [Opções parciais](#)
- [Projen](#)

Projeto Blueprint

Um projeto de planta oferece a capacidade de desenvolver e publicar plantas 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 blueprint, se você optar por gerar uma versão do fluxo de trabalho, um fluxo de trabalho de publicação será criado no seu blueprint com o blueprint Builder. O fluxo de trabalho publica automaticamente sua versão mais recente.

Plantas espaciais

Você pode visualizar e gerenciar todos os blueprints da tabela Space blueprints ao navegar até a seção Blueprints do seu espaço. Depois que as plantas são publicadas em seu espaço, elas são disponibilizadas como uma planta espacial para serem adicionadas e removidas do catálogo de plantas do seu espaço. Você também pode gerenciar as permissões de publicação e excluir blueprints na seção Blueprints do seu espaço. Para ter mais informações, consulte [Visualizando detalhes, versões e projetos de um blueprint personalizado](#).

Catálogo de plantas espaciais

Você pode visualizar todas as plantas personalizadas adicionadas no catálogo de plantas de um espaço. É aqui que um membro do espaço pode escolher seu plano personalizado para criar um novo projeto. Esse catálogo é diferente do CodeCatalyst catálogo, que já tem plantas disponíveis para todos os membros do espaço. Para ter mais informações, consulte [Criando um projeto abrangente com CodeCatalyst plantas](#).

Síntese

A síntese é o processo de gerar um pacote de CodeCatalyst projetos que representa o código-fonte, a configuração e os recursos de um projeto. O pacote é então usado pelas operações da API CodeCatalyst de implantação para implantar em um projeto. O processo pode ser executado localmente durante o desenvolvimento de seu plano 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 blueprint começa chamando a `blueprint.ts` classe principal com essa opção mesclada. `defaults.json` Um novo pacote de projeto é gerado na `synth/synth.[options-name]/proposed-bundle/` pasta. A saída inclui o pacote de projetos que um blueprint personalizado gera, dadas as opções definidas, incluindo as [opções parciais](#) que você pode ter configurado.

Ressíntese

A ressíntese é o processo de regeneração de um blueprint com diferentes opções de blueprint ou versões de blueprint de projetos existentes. Como autor do blueprint, você pode definir estratégias de mesclagem personalizadas no código do blueprint personalizado. Você também pode definir limites

de propriedade em um `.ownership-file` para especificar em quais partes da base de código um blueprint pode ser atualizado. Embora o blueprint personalizado possa propor atualizações para o `.ownership-file`, os desenvolvedores de projetos que usam o blueprint personalizado podem determinar os limites de propriedade de seus projetos. Você pode executar a ressíntese localmente, testar e atualizar antes de publicar seu blueprint personalizado. Use os comandos a seguir para realizar a ressíntese:

```
yarn blueprint:resynth          # fast mode
yarn blueprint:resynth --cache  # wizard emulation mode
```

O blueprint começa chamando a `blueprint.ts` classe principal com essa opção mesclada. `defaults.json` Um novo pacote de projeto é gerado na `synth/resynth.[options-name]` pasta. A saída inclui o pacote de projetos que um blueprint 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 blueprint de destino.
- `existing-bundle` - Uma simulação do seu projeto existente. Se não houver nada nessa pasta, ela será gerada com a mesma saída que `proposed-bundle`.
- `ancestral-bundle` - Uma simulação do que seu blueprint geraria quando executado com uma versão anterior, opções anteriores ou uma combinação. Se não houver nada nessa pasta, ela será gerada com a mesma saída de `proposed-bundle`.
- `pacote resolvido` - O pacote é sempre regenerado e o padrão é uma mesclagem de três vias entre o, e o. `proposed-bundle existing-bundle ancestor-bundle` Esse pacote fornece uma emulação do que uma ressíntese produziria localmente.

Para saber mais sobre os pacotes de saída do blueprint, consulte. [Gerando arquivos com ressíntese](#)

Opções parciais

Você pode adicionar variações de opções `src/wizard-configuration/` que não precisem enumerar toda a `Options` interface, e as opções são mescladas na parte superior do arquivo. `defaults.json` Isso permite que você personalize os casos de teste em opções específicas.

Exemplo:

Optionsinterface:

```
{
  language: "Python" | "Java" | "Typescript",
  repositoryName: string
  ...
}
```

Arquivo defaults.json:

```
{
  language: "Python",
  repositoryName: "Myrepo"
  ...
}
```

Testes adicionais de configuração:

- #wizard-config-typescript-test.json

```
{
  language: "Typescript",
}
```

- #wizard-config-java-test.json

```
{
  language: "Java",
}
```

Projen

O Projen é uma ferramenta de código aberto que os projetos personalizados usam para se manterem atualizados e consistentes. Os blueprints vêm como pacotes do Projen porque essa estrutura fornece a capacidade de criar, agrupar e publicar projetos, e você pode usar a interface para gerenciar as configurações e 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 plano que gera um pacote de projetos. O Projen possui a configuração de um projeto e isso não deve impactar você como autor do blueprint. Você pode executar `yarn projen` para regenerar a configuração do seu projeto depois

de adicionar dependências ou pode alterar as opções no `projenc.ts` arquivo. O Projenc também é a ferramenta de geração subjacente para esquemas personalizados para sintetizar um projeto. Para obter mais informações, consulte a [página do projeto GitHub](#). Para saber mais sobre como trabalhar com o Projenc, consulte a [documentação do Projenc](#) e [Como simplificar a configuração do projeto com o Projenc](#).

Introdução aos esquemas personalizados

Durante o processo de criação de um blueprint, você pode configurar o blueprint e gerar uma prévia dos recursos do projeto. Cada blueprint personalizado é gerenciado por um CodeCatalyst projeto, que contém um fluxo de trabalho por padrão para publicação no catálogo de blueprints do espaço.

Ao configurar os detalhes do seu blueprint personalizado, você também pode optar por armazenar o código-fonte do seu blueprint em um repositório de terceiros, onde você ainda pode gerenciar o blueprint personalizado e utilizar os recursos de gerenciamento do ciclo de vida para manter os projetos do seu espaço sincronizados quando o blueprint personalizado é modificado. Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#) e [Trabalhando com o gerenciamento do ciclo de vida como autor do plano](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um blueprint personalizado no CodeCatalyst](#)
- [Etapa 2: Desenvolver um plano personalizado com componentes](#)
- [Etapa 3: visualizar um blueprint personalizado](#)
- [\(Opcional\) Etapa 4: publicar uma versão de pré-visualização personalizada do blueprint](#)

Pré-requisitos

Antes de criar um blueprint personalizado, considere os seguintes requisitos:

- Seu CodeCatalyst espaço deve ser do nível Enterprise. Para obter mais informações, consulte [Gerenciando o faturamento](#) no Guia do CodeCatalyst Administrador da Amazon.
- Você precisa ter a função de administrador do Space ou de usuário avançado para criar blueprints personalizados. Para ter mais informações, consulte [Concedendo acesso com funções de usuário](#).

Etapa 1: criar um blueprint personalizado no CodeCatalyst

Quando você cria um blueprint personalizado a partir das configurações do seu espaço, um repositório é criado para você. O repositório inclui todos os recursos necessários para desenvolver seu blueprint antes de publicá-lo no catálogo de blueprints do espaço.

Para criar um blueprint personalizado

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja criar um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Escolha Criar blueprint.
5. Em Nomeie seu blueprint, 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. Como alternativa, você pode optar por armazenar o código-fonte do blueprint em um repositório de terceiros. Para ter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Faça o seguinte, dependendo do provedor de repositório terceirizado 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](#).

- Bitbucket: Conecte um espaço de trabalho do Bitbucket.

Escolha o menu suspenso Avançado, escolha Bitbucket como provedor de repositório e, em seguida, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo blueprint.

7. Em Detalhes do Blueprint, faça o seguinte:
 - a. No campo de entrada de texto do nome de exibição do Blueprint, insira um nome que aparecerá no catálogo do blueprint do seu espaço.
 - b. No campo de entrada de texto Descrição, insira uma descrição para seu blueprint personalizado.
 - c. No campo de entrada de texto Nome do autor, insira um nome de autor para seu blueprint personalizado.
 - d. (Opcional) Escolha as configurações avançadas.
 - i. Escolha + Adicionar para adicionar tags que são adicionadas ao `package.json` arquivo.
 - ii. Escolha o menu suspenso Licença e, em seguida, escolha uma licença para seu blueprint personalizado.
 - iii. No campo de entrada de texto do nome do pacote do Blueprint, insira um nome para identificar seu pacote do blueprint.
 - iv. Por padrão, um fluxo de trabalho de lançamento é gerado usando um blueprint de publicação em seu projeto chamado Blueprint Builder. O fluxo de trabalho publica a versão mais recente do blueprint 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 Liberar fluxo de trabalho.
8. (Opcional) Um projeto de blueprint vem com código predefinido para dar suporte à publicação do blueprint no catálogo espacial. Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, escolha Exibir código ou Exibir fluxo de trabalho em Gerar visualização do blueprint.
9. Escolha Criar blueprint.

Se você não desativou a geração do fluxo de trabalho para seu blueprint personalizado, o fluxo de trabalho começa a ser executado automaticamente quando seu blueprint é criado. Quando a execução do fluxo de trabalho estiver concluída, seu blueprint personalizado estará disponível

para ser adicionado ao catálogo de blueprints do seu espaço por padrão. Você pode desativar as permissões de publicação se não quiser que a versão mais recente do blueprint seja publicada automaticamente no seu espaço. Para obter mais informações, consulte [Definindo permissões de publicação para um blueprint personalizado](#) e [Executar um fluxo de trabalho](#).

Como o fluxo de trabalho de publicação chamado `blueprint-release` é criado usando um blueprint, o blueprint pode ser encontrado como um blueprint aplicado em seu projeto. Para obter mais informações, consulte [Aplicando um blueprint em um projeto para adicionar recursos](#) e [Desassociar um blueprint de um projeto para interromper as atualizações](#).

Etapa 2: Desenvolver um plano personalizado com componentes

Um assistente de blueprint é gerado quando você cria um blueprint personalizado e pode ser modificado com componentes ao desenvolver o blueprint personalizado. Você pode atualizar os `src/defaults.json` arquivos `src/blueprints.js` e para modificar o assistente.

Important

Se você quiser usar pacotes de blueprint de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos blueprints personalizados que adiciona ao seu espaço e pelo código que eles geram.

Crie um ambiente de desenvolvimento em seu CodeCatalyst projeto com um ambiente de desenvolvimento integrado (IDE) compatível antes de configurar seu código de blueprint. É necessário um ambiente de desenvolvimento para trabalhar com as ferramentas e pacotes necessários.

Para criar um ambiente de desenvolvimento

1. No painel de navegação, faça o seguinte:
 - a. Escolha Visão geral e, em seguida, navegue até a seção Meus ambientes de desenvolvimento.
 - b. Escolha Código e, em seguida, escolha Dev Environments.
 - c. Escolha Código, escolha Repositórios de origem e escolha o repositório que você criou ao criar seu blueprint.
2. Escolha Criar ambiente de desenvolvimento.

3. Escolha um IDE compatível no menu suspenso. Consulte Ambientes de [desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) para obter mais informações.
4. Escolha Trabalhar na ramificação existente e, no menu suspenso Ramificação existente, escolha a ramificação do recurso que você criou.
5. (Opcional) No campo Alias - entrada de texto opcional, insira um alias para identificar o ambiente de desenvolvimento.
6. Selecione Create (Criar). Enquanto seu ambiente de desenvolvimento está sendo criado, a coluna de status do ambiente de desenvolvimento exibe Iniciando e a coluna de status exibe Executando quando o ambiente de desenvolvimento foi criado.

Para ter mais informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#).

Para desenvolver seu plano personalizado

1. Em um terminal em funcionamento, use o seguinte yarn comando para instalar dependências:

```
yarn
```

As ferramentas e os pacotes necessários são disponibilizados por meio do CodeCatalyst Dev Environment, incluindo o Yarn. Se você estiver trabalhando em um blueprint personalizado sem um ambiente de desenvolvimento, instale primeiro o Yarn em seu sistema. Para obter mais informações, consulte a [documentação de instalação do Yarn](#).

2. Desenvolva seu plano personalizado para que ele seja configurado de acordo com suas preferências. Você pode modificar o assistente do seu blueprint adicionando componentes. Para obter mais informações, consulte [Desenvolvendo um plano personalizado para atender aos requisitos do projeto](#), [Modificando recursos do blueprint com um assistente de front-end](#) e [Publicar um blueprint personalizado em um espaço](#).

Etapa 3: visualizar um blueprint personalizado

Depois de configurar e desenvolver seu blueprint personalizado, você pode visualizar e publicar a versão prévia do seu blueprint em seu espaço. Uma versão prévia permite verificar se o blueprint é o que você deseja antes de ser usado para criar novos projetos ou aplicado a projetos existentes.

Para visualizar um blueprint personalizado

1. Em um terminal em funcionamento, use o seguinte yarn comando:

```
yarn blueprint:preview
```

2. Navegue até o `See this blueprint at:` link fornecido para visualizar seu plano personalizado.
3. Verifique se a interface do usuário, incluindo o texto, aparece conforme o esperado com base na sua configuração. Se quiser alterar seu blueprint personalizado, você pode editar o `blueprint.ts` arquivo, ressintetizar o blueprint e, em seguida, publicar uma versão prévia novamente. Para ter mais informações, consulte [Ressíntese](#).

(Opcional) Etapa 4: publicar uma versão de pré-visualização personalizada do blueprint

Você pode publicar uma versão prévia do seu blueprint personalizado no seu espaço se quiser adicioná-la ao catálogo de blueprints do seu espaço. Isso permite que você visualize o blueprint 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 `0.0.1` versão, 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 blueprint personalizado

Navegue até o `Enable version [version number] at:` link fornecido para ativar seu plano personalizado. Esse link é fornecido ao executar o yarn comando em [Etapa 3: visualizar um blueprint personalizado](#).

Depois de criar, desenvolver, visualizar e publicar seu blueprint personalizado, você pode publicar e adicionar a versão final do blueprint ao catálogo de blueprints do seu espaço. Para ter mais informações, consulte [Publicar um blueprint personalizado em um espaço](#) e [Adicionar um blueprint personalizado a um catálogo espacial](#).

Tutorial: Criando e atualizando um aplicativo React

Como autor de plantas, você pode desenvolver e adicionar plantas personalizadas ao catálogo de plantas do seu espaço. Esses esquemas podem então ser usados pelos membros do espaço para criar novos projetos ou aplicá-los a projetos existentes. Você pode continuar fazendo alterações em seus blueprints, que são então disponibilizadas como atualizações por meio de pull requests.

Este tutorial fornece um passo a passo a partir da perspectiva do autor do projeto e da perspectiva do usuário do projeto. O tutorial mostra como criar um blueprint de aplicativo web de página única do React. O blueprint é então usado para criar um novo projeto. Quando o blueprint é atualizado com alterações, o projeto criado a partir do blueprint incorpora essas alterações por meio de uma pull request.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar um blueprint personalizado](#)
- [Etapa 2: Visualizar o fluxo de trabalho de lançamento](#)
- [Etapa 3: Adicionar um blueprint ao catálogo](#)
- [Etapa 4: criar um projeto com o blueprint](#)
- [Etapa 5: Atualizar o blueprint](#)
- [Etapa 6: atualizar a versão publicada do catálogo do blueprint para a nova versão](#)
- [Etapa 7: atualizar o projeto com a nova versão do blueprint](#)
- [Etapa 8: Visualize as alterações no projeto](#)

Pré-requisitos

Para criar e atualizar um blueprint personalizado, você deve ter concluído as tarefas da [Configure e faça login em CodeCatalyst](#) seguinte forma:

- Tenha um ID de AWS construtor para fazer login CodeCatalyst.
- Pertença a um espaço e tenha a função de administrador do espaço ou usuário avançado atribuída a você nesse espaço. Para obter mais informações, consulte [Criando um espaço](#), [Concedendo permissões de espaço aos usuários](#) e [Função de administrador de espaço](#).

Etapa 1: criar um blueprint personalizado

Quando você cria um blueprint personalizado, é criado um CodeCatalyst projeto que contém o código-fonte e as ferramentas e recursos de desenvolvimento do seu blueprint. Seu projeto é onde você desenvolverá, testará e publicará o plano.

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. Escolha Configurações para navegar até as configurações de espaço.
4. Na guia Configurações de espaço, escolha Blueprints e escolha Criar blueprint.
5. Atualize os campos no assistente de criação de blueprint com os seguintes valores:
 - Em Nome do blueprint, insira `react-app-blueprint`.
 - Em Nome de exibição do Blueprint, insira `react-app-blueprint`.
6. Opcionalmente, escolha Exibir código para visualizar o código-fonte do blueprint para seu blueprint. Da mesma forma, escolha Exibir fluxo de trabalho para visualizar o fluxo de trabalho que será criado no projeto que cria e publica o blueprint.
7. Escolha Criar blueprint.
8. Depois que seu blueprint é criado, você é levado ao projeto do blueprint. Esse projeto contém o código-fonte do blueprint, junto com as ferramentas e os recursos necessários para desenvolver, testar e publicar o blueprint. Um fluxo de trabalho de lançamento foi gerado e publicou automaticamente seu blueprint no espaço.
9. Agora que seu blueprint e seu projeto de blueprint foram criados, a próxima etapa é configurá-los atualizando o código-fonte. Você pode usar Dev Environments para abrir e editar seu repositório de origem diretamente no seu navegador.

No painel de navegação, escolha Código e, em seguida, escolha Dev Environments.

10. Escolha Create Dev Environment e, em seguida, escolha AWS Cloud9 (no navegador).
11. Mantenha as configurações padrão e escolha Criar.
12. No AWS Cloud9 terminal, navegue até o diretório do projeto blueprint executando o seguinte comando:

```
cd react-app-blueprint
```

13. Uma `static-assets` pasta é criada e preenchida automaticamente quando um blueprint é criado. Neste tutorial, você excluirá a pasta padrão e gerará uma nova para um blueprint do aplicativo react.

Exclua a pasta `static-assets` executando o seguinte comando:

```
rm -r static-assets
```

AWS Cloud9 é construído em uma plataforma baseada em Linux. Se você estiver usando um sistema operacional Windows, poderá usar o seguinte comando em vez disso:

```
rmdir /s /q static-assets
```

14. Agora que a pasta padrão foi excluída, crie uma `static-assets` pasta para um blueprint do `react-app` executando o seguinte comando:

```
npx create-react-app static-assets
```

Se solicitado, insira `y` para continuar.

Um novo aplicativo `react` foi criado na `static-assets` pasta com os pacotes necessários. As alterações precisam ser enviadas para seu repositório CodeCatalyst de origem remoto.

15. Verifique se você tem as alterações mais recentes e, em seguida, confirme e envie as alterações para o repositório de CodeCatalyst origem do blueprint executando os seguintes comandos:

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Quando uma alteração é enviada para o repositório de origem do blueprint, o fluxo de trabalho de lançamento é iniciado automaticamente. Esse fluxo de trabalho incrementa a versão do blueprint, cria o blueprint e o publica em seu espaço. Na próxima etapa, você navegará até a execução do fluxo de trabalho de lançamento para ver como está o desempenho.

Etapa 2: Visualizar o fluxo de trabalho de lançamento

1. No CodeCatalyst console, no painel de navegação, escolha `CI/CD` e, em seguida, escolha `Fluxos de trabalho`.
2. Escolha o fluxo de trabalho de lançamento do `blueprint-release`.
3. Você pode ver que o fluxo de trabalho tem ações para criar e publicar o blueprint.

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, sua nova versão do blueprint será publicada. As versões publicadas do blueprint podem ser vistas nas Configurações do seu espaço, mas não podem ser usadas em projetos até serem adicionadas ao catálogo de blueprints do espaço. Na próxima etapa, você adicionará o blueprint ao catálogo.

Etapa 3: Adicionar um blueprint ao catálogo

Adicionar uma planta ao catálogo de plantas do espaço torna a planta disponível para uso em todos os projetos em um espaço. Os membros do espaço podem usar o blueprint para criar novos projetos ou aplicá-los a projetos existentes.

1. No CodeCatalyst console, navegue de volta para o espaço.
2. Escolha Configurações e, em seguida, escolha Blueprints.
3. Escolha e react-app-blueprint, em seguida, escolha Adicionar ao catálogo.
4. Selecione Salvar.

Etapa 4: criar um projeto com o blueprint

Agora que o blueprint foi adicionado ao catálogo, ele pode ser usado em projetos. Nesta etapa, você criará um projeto com o blueprint que acabou de criar. Em uma etapa posterior, você atualizará esse projeto atualizando e publicando uma nova versão do blueprint.

1. Escolha a guia Projetos e, em seguida, escolha Criar projeto.
2. Escolha Space blueprints e, em seguida, escolha react-app-blueprint.

Note

Depois que o blueprint for escolhido, você poderá ver o conteúdo do arquivo do blueprint. README.md

3. Escolha Próximo.

4.

Note

O conteúdo desse assistente de criação de projeto pode ser configurado no blueprint.

Insira o nome do projeto como usuário do blueprint. Para este tutorial, insira `react-app-project`. Para ter mais informações, consulte [Desenvolvendo um plano personalizado para atender aos requisitos do projeto](#).

Em seguida, você atualizará o blueprint e adicionará a nova versão ao catálogo, que será usada para atualizar esse projeto.

Etapa 5: Atualizar o blueprint

Depois que um blueprint é usado para criar um novo projeto ou aplicado a projetos existentes, você pode continuar fazendo atualizações como autor do blueprint. Nesta etapa, você fará alterações no blueprint e publicará automaticamente uma nova versão no espaço. A nova versão pode então ser adicionada como a versão do catálogo.

1. Navegue até o `react-app-blueprint` projeto criado em [Tutorial: Criando e atualizando um aplicativo React](#).
2. Abra o ambiente de desenvolvimento criado em [Tutorial: Criando e atualizando um aplicativo React](#).
 - a. No painel de navegação, escolha Código e, em seguida, escolha Dev Environments.
 - b. Na tabela, localize o Dev Environment e escolha Abrir em AWS Cloud9 (no navegador).
3. Quando o fluxo de trabalho de lançamento do blueprint foi executado, ele incrementou a versão do blueprint atualizando o arquivo `package.json`. Faça essa alteração executando o seguinte comando no AWS Cloud9 terminal:

```
git pull
```

4. Navegue até a `static-assets` pasta executando o seguinte comando:

```
cd /projects/react-app-blueprint/static-assets
```

5. Crie um `hello-world.txt` arquivo na `static-assets` pasta executando o seguinte comando:

```
touch hello-world.txt
```

AWS Cloud9 é construído em uma plataforma baseada em Linux. Se você estiver usando um sistema operacional Windows, poderá usar o seguinte comando em vez disso:

```
echo > hello-world.txt
```

6. Na navegação à esquerda, clique duas vezes no `hello-world.txt` arquivo 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, em seguida, confirme e envie as alterações para o repositório de CodeCatalyst origem do blueprint 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 blueprint no espaço.

Etapa 6: atualizar a versão publicada do catálogo do blueprint para a nova versão

Depois que um blueprint é usado para criar um novo projeto ou aplicado a projetos existentes, você ainda pode atualizar o blueprint como autor do blueprint. Nesta etapa, você fará alterações no blueprint e alterará a versão do catálogo do blueprint.

1. No CodeCatalyst console, navegue de volta para o espaço.
2. Escolha Configurações e, em seguida, escolha Blueprints.
3. Escolha `react-app-blueprint`, em seguida, escolha Gerenciar versão do catálogo.
4. Escolha a nova versão e, em seguida, escolha Salvar.

Etapa 7: atualizar o projeto com a nova versão do blueprint

Uma nova versão já está disponível no catálogo de plantas do espaço. Como usuário do blueprint, você pode atualizar a versão do projeto criado em [Etapa 4: criar um projeto com o blueprint](#). Isso garante que você tenha as alterações e correções mais recentes necessárias para atender às melhores práticas.

1. No CodeCatalyst console, navegue até o react-app-projectprojeto criado em [Etapa 4: criar um projeto com o blueprint](#).
2. No painel de navegação, escolha Esquemas.
3. Escolha Atualizar blueprint na caixa de informações.
4. No painel Alterações de código do lado direito, você pode ver as package.json atualizações hello-world.txt e.
5. Escolha Aplicar atualização.

Escolher Aplicar atualização cria uma pull request no projeto com as alterações da versão atualizada do blueprint. Para fazer as atualizações no projeto, você deve mesclar a pull request. Para ter mais informações, consulte [Analisando uma pull request](#) e [Mesclando uma pull request](#).

1. Na tabela Blueprints, encontre o blueprint. Na coluna Status, escolha pull request pendente e, em seguida, escolha o link para o pull request aberto.
2. Revise a pull request e escolha Mesclar.
3. Escolha Mesclagem rápida para manter os valores padrão e, em seguida, escolha Mesclar.

Etapa 8: Visualize as alterações no projeto

As alterações no blueprint agora estão disponíveis em seu projeto posteriormente [Etapa 7: atualizar o projeto com a nova versão do blueprint](#). Como usuário do blueprint, você pode visualizar as alterações no repositório de origem.

1. No painel de navegação, escolha Repositórios de origem e, em seguida, escolha o nome do repositório de origem criado quando o projeto foi criado.
2. Em Arquivos, você pode visualizar o hello-world.txt arquivo que foi criado em [Etapa 5: Atualizar o blueprint](#).
3. Escolha a hello-world.txt para ver o conteúdo do arquivo.

O gerenciamento do ciclo de vida fornece aos autores do blueprint a capacidade de gerenciar centralmente o ciclo de vida de desenvolvimento de software de cada projeto que contém um plano específico. Conforme visto neste tutorial, você pode enviar atualizações para o blueprint que podem ser incorporadas por projetos que usaram o blueprint para criar um novo projeto ou aplicá-lo a um projeto existente. Para ter mais informações, consulte [Trabalhando com o gerenciamento do ciclo de vida como autor do plano](#).

Trabalhando com o gerenciamento do ciclo de vida como autor do plano

O gerenciamento do ciclo de vida permite que você mantenha um grande número de projetos sincronizados a partir de uma única fonte comum de melhores práticas. Isso dimensiona 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 tempo de execução, mudanças nas melhores práticas e outras práticas de manutenção porque esses padrões são definidos em um só lugar e automaticamente atualizados centralmente quando novos padrões são publicados.

Quando uma nova versão do seu blueprint é publicada, todos os projetos que contêm esse blueprint são solicitados a atualizar para a versão mais recente. Como autor do blueprint, você também pode ver a versão de um determinado blueprint que cada projeto contém para fins de conformidade. Quando há conflitos em um repositório de origem existente, o gerenciamento do ciclo de vida cria pull requests. Para todos os outros recursos, como o Dev Environment, todas as atualizações de gerenciamento do ciclo de vida criam estritamente novos recursos. Os usuários podem mesclar ou não esses pull requests. Quando as pull requests pendentes são mescladas, a versão do blueprint, incluindo as opções, usada em seu projeto é então atualizada. Para saber mais sobre como trabalhar com o gerenciamento do ciclo de vida como usuário do blueprint, consulte e. [Usando o gerenciamento do ciclo de vida em projetos existentes](#) [Usando o gerenciamento do ciclo de vida em vários esquemas em um projeto](#)

Tópicos

- [Testando o gerenciamento do ciclo de vida para saídas de pacotes e conflitos de mesclagem](#)
- [Usando estratégias de mesclagem para gerar pacotes e especificar arquivos](#)
- [Acessando objetos de contexto para obter detalhes do projeto](#)

Testando o 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 blueprint e mesclar a resolução de conflitos. Uma série de pacotes no `synth/` diretório que representam as várias fases de uma atualização do ciclo de vida é gerada. Para testar o gerenciamento do ciclo de vida, você pode executar o seguinte comando yarn em seu blueprint: `yarn blueprint: resynth` Para saber mais sobre ressíntese e pacotes, consulte e. [Ressíntese Gerando arquivos com ressíntese](#)

Usando estratégias de mesclagem para gerar pacotes e especificar arquivos

Tópicos

- [Gerando arquivos com ressíntese](#)
- [Usando estratégias de mesclagem](#)
- [Especificação de arquivos para atualizações de gerenciamento do ciclo de vida](#)
- [Escrevendo estratégias de mesclagem](#)

Gerando arquivos com ressíntese

A ressíntese pode mesclar o código-fonte produzido por um blueprint com o código-fonte que foi gerado anteriormente pelo mesmo blueprint, permitindo que as alterações em um blueprint sejam propagadas para projetos existentes. As mesclagens são executadas a partir da `resynth()` função nos pacotes de saída do blueprint. A ressíntese primeiro gera três pacotes representando diferentes aspectos do plano e do estado do projeto. Ele pode ser executado manualmente localmente com o `yarn blueprint:resynth` comando, 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 blueprints só executam a ressíntese nos repositórios abaixo, `src/*` já que somente essa parte do pacote normalmente está sob controle de origem.

- `existing-bundle`- Esse pacote é uma representação do estado existente do projeto. Isso é construído artificialmente pela computação de síntese para fornecer ao plano 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 para o conteúdo do `ancestor-bundle`.
- `ancestor-bundle`- Esse é o pacote que representa a saída do blueprint se tiver sido sintetizado com algumas opções e/ou versões anteriores. Se for a primeira vez que esse blueprint está sendo adicionado a um projeto, o ancestral não existe, então ele está definido com o mesmo conteúdo

do. `existing-bundle` Localmente, se esse pacote já existir nesse local, ele será respeitado como uma simulação.

- `proposed-bundle`- Este é o pacote que simula o blueprint se ele foi sintetizado com algumas novas opções e/ou versões. Esse é o mesmo pacote que seria produzido pela `synth()` função. Localmente, esse pacote é sempre substituído.

Cada pacote é criado durante uma fase de ressíntese que pode ser acessada a partir da classe de blueprint abaixo. `this.context.resynthesisPhase`

- `resolved-bundle`- Esse é o pacote final, que é uma representação do que é empacotado e implantado em um projeto. CodeCatalyst Você pode ver quais arquivos e diffs são enviados para os mecanismos de implantação. Essa é a saída da `resynth()` função que resolve mesclagens entre os outros três pacotes.

A mesclagem de três vias é aplicada tomando a diferença entre o `ancestor-bundle` `proposed-bundle` e aplicando-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 blueprint `resynth()` e produz o pacote resolvido a partir do resultado.

Usando estratégias de mesclagem

Você pode usar uma estratégia de mesclagem oferecida pela biblioteca de blueprints. Essas estratégias fornecem maneiras de resolver saídas de arquivos e conflitos para arquivos mencionados na [Gerando arquivos com ressíntese](#) seção.

- `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 resolve o arquivo proposto quando um arquivo existente ainda não existe. Caso contrário, resolve para 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 GitHub](#).

Especificação de arquivos para atualizações de gerenciamento do ciclo de vida

Durante a ressíntese, os blueprints controlam como as alterações são mescladas em um repositório de origem existente. No entanto, talvez você não queira enviar atualizações para todos os arquivos do seu blueprint. 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 blueprints podem especificar quais arquivos eles possuem e quais não possuem, especificando estratégias de mesclagem na própria construção do repositório. Os blueprints podem atualizar suas 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 são padronizados para um three-way-merge similar ao Git. Existem várias

estratégias de mesclagem fornecidas por meio da MergeStrategies construção, mas você pode escrever a sua própria. As estratégias fornecidas seguem o driver de [estratégia do git merge](#).

Escrevendo estratégias de mesclagem

Além de usar uma das estratégias de fusão de compilações fornecidas, você também pode escrever suas próprias estratégias. As estratégias devem seguir uma interface estratégica padrão. Você deve escrever uma função de estratégia que pegue versões de um arquivo do `existing-bundle`, `proposed-bundle`, `ancestor-bundle`, e as mescle 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:

```
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/**',
    ],
},
],
```

Acessando objetos de contexto para obter detalhes do projeto

Como autor do blueprint, você pode acessar o contexto do projeto do blueprint durante a síntese para obter informações como espaço e nomes de projetos, ou arquivos existentes no repositório de origem de um projeto. Você também pode obter detalhes como a fase de ressíntese que o projeto está gerando. Por exemplo, você pode acessar o contexto para saber se está ressíntetizando para gerar um pacote ancestral ou um pacote proposto. O contexto de código existente pode então ser usado para transformar seu 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 `blueprint.ts` arquivo 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 modelo 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));
}
}
```

Desenvolvendo um plano personalizado para atender aos requisitos do projeto

Antes de publicar um blueprint personalizado, você pode desenvolver o blueprint para atender a requisitos específicos. Você pode desenvolver seu blueprint personalizado e testar o blueprint criando um projeto durante a pré-visualização. Você pode desenvolver o blueprint 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 blueprint de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos blueprints personalizados que adiciona ao seu espaço e pelo código que eles geram.

Para desenvolver ou atualizar um blueprint personalizado

1. Retome seu ambiente de desenvolvimento. Para ter mais informações, consulte [Retomar um ambiente de desenvolvimento](#).

Se você não tiver um ambiente de desenvolvimento, primeiro deverá criar um. Para ter mais informações, consulte [Criar um ambiente de desenvolvimento](#).

2. Abra um terminal funcional em seu ambiente de desenvolvimento.
3. Se você optou por um fluxo de trabalho de lançamento ao criar seu blueprint, a versão mais recente do blueprint será publicada automaticamente. Faça as alterações para garantir que o `package.json` arquivo tenha a versão incrementada. Use o seguinte comando:

```
git pull
```

4. No `src/blueprint.ts` arquivo, edite as opções do seu blueprint personalizado. A `Options` interface é interpretada dinamicamente pelo CodeCatalyst assistente para gerar uma interface de usuário (UI) de seleção. Você pode desenvolver seu blueprint personalizado adicionando componentes e tags compatíveis. Para obter mais informações, consulte [Modificando recursos do blueprint com um assistente de front-end](#), [Adicionando componentes do ambiente a um blueprint](#), [Adicionando componentes da região a um blueprint](#), [Adicionar componentes de repositório e código-fonte a um blueprint](#), [Adicionar componentes do fluxo de trabalho a um blueprint](#) e [Adicionando componentes do Dev Environments a um blueprint](#).

Você também pode visualizar o SDK de blueprints e exemplos de blueprints para obter suporte adicional ao desenvolver seu blueprint personalizado. Para obter mais informações, consulte o [GitHub repositório de código aberto](#).

Planos 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 CodeCatalyst implantação para implantar em um projeto. Se você quiser continuar desenvolvendo seu blueprint personalizado, execute novamente o processo de síntese do blueprint. Para ter mais informações, consulte [Conceitos de plantas personalizadas](#).

Modificando recursos do blueprint com um assistente de front-end

Um assistente de seleção de blueprint ativado CodeCatalyst é gerado automaticamente pela `Options` interface no `blueprint.ts` arquivo. O assistente de front-end oferece suporte a modificações e recursos de um blueprint `Options` usando comentários e tags no estilo [JSDOC](#). Você pode usar comentários e tags no estilo JSDOC para realizar tarefas. Por exemplo, você pode selecionar o texto exibido acima de uma opção, ativar recursos como validação de entrada ou tornar uma opção dobrável. O assistente funciona interpretando uma árvore de sintaxe abstrata (AST) gerada a partir do TypeScript tipo da `Options` interface. O assistente se configura 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 blueprints:

Options

```
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 Options interface aparece camelCase por padrão. O texto simples no comentário no estilo JSDOC é exibido como texto acima da opção no assistente.

Tópicos

- [Etiquetas suportadas](#)
- [TypeScript Tipos suportados](#)
- [Comunicação com o usuário durante a síntese](#)

Etiquetas suportadas

As seguintes tags JSDOC são suportadas por um blueprint personalizado Options no assistente de front-end.

@inlinePolicy. /caminho/para/policy/file.json

- Requer - Opção de ser um tipoRole.
- Uso - Permite que você comunique as políticas em linha de que uma função precisa. Espere-se que o `policy.json` caminho esteja no código-fonte. Use essa tag quando precisar de uma política personalizada para uma função.
- Dependências - `blueprint-cli 0.1.12` e superiores
- Exemplo - `@inlinePolicy ./deployment-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @inlinePolicy ./path/to/deployment-policy.json
     */
    cdkRole: Role[];
  };
};
```

@trustPolicy. /caminho/para/policy/file.json

- Requer - Opção de ser um tipoRole.
- Uso - Permite que você comunique as políticas de confiança de que uma função precisa. Espere-se que o `policy.json` caminho esteja no código-fonte. Use essa tag quando precisar de uma política personalizada para uma função.
- Dependências - `blueprint-cli 0.1.12` e superiores
- Exemplo - `@trustPolicy ./trust-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @trustPolicy ./path/to/trust-policy.json
     */
    cdkRole: Role[];
  };
};
```

Expressão @validationRegex Regex

- 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. Por padrão, a mensagem de validação está vazia.

string @validationMessage

- Requer - @validationRegex ou outros erros para revisar o uso.
- Uso - Exibe mensagem de validação em @validation* caso de falha.
- Exemplo -@validationMessage Must contain only upper and lowercase letters, numbers, and underscores.
- Recomendação - Use com@validationMessage. Por padrão, a mensagem de validação está vazia.

@collapsed boolean (opcional)

- Requer - N/A
- Uso - Booleano que permite que uma subopção seja dobrável. Se a anotação reduzida estiver presente, seu valor padrão será verdadeiro. Definir o valor como @collapsed false cria uma seção dobrável que é inicialmente aberta.
- Exemplo - @collapsed true

string @displayName

- Requer - N/A
- Uso - Altera o nome de exibição da opção. Permite formatos diferentes do 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 do [CamelCase](#) para o nome de exibição.
- Exemplo - @displayName Blueprint Name

número @defaultEntropy

- Requer - Opção de ser uma string.
- Uso - Anexa uma sequência alfanumérica aleatória 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

@hidden boolean (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

@button boolean (opcional)

- Requer - N/A
- Uso - A anotação deve estar em uma propriedade booleana. Adiciona um botão que será sintetizado como verdadeiro quando escolhido. Não é uma alavanca.
- Exemplo - `buttonExample: boolean;`

```
/**
 * @button
 */
buttonExample: boolean;
```

@showName boolean (opcional)

- Requer - N/A
- Uso - Só pode ser usado em um tipo de conexão de conta. Mostra a entrada de nome oculto. Padronizado como `default_environment`.
- Exemplo - `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
    ...
}>;
```

@ showEnvironmentType boolean (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<{
  ...
}>;
```

@forceDefault boolean (opcional)

- Requer - N/A
- Uso - usa o valor padrão fornecido pelo autor do blueprint em vez do valor usado anteriormente pelo usuário.
- Exemplo - forceDefaultExample: any;

```
/**
 * @forceDefault
 */
forceDefaultExample: any;
```

@requires Nome do blueprint

- Requer - Anota a interface Options
- Uso - avisa o usuário para aplicar o especificado blueprintName no projeto como um requisito para o blueprint atual.
- Exemplo - @requires '@amazon-codecatalyst/blueprints.blueprint-builder'

```
/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
  ...
}
```

TypeScript Tipos suportados

Os TypeScript tipos a seguir são compatíveis com um esquema personalizado Options no assistente de front-end.

Número

- Requer - Opção de ser um tiponumber.

- Uso - Gere um campo de entrada numérica.
- Exemplo - age: number

```
{  
  age: number  
  ...  
}
```

String

- Requer - Opção de ser um tipostring.
- Uso - Gere um campo de entrada de string.
- Exemplo - name: string

```
{  
  age: string  
  ...  
}
```

Lista de strings

- Requer - Opção de ser um tipoboolean.
- Uso - Gere uma caixa de seleção.
- Exemplo - isProduction: boolean

```
{  
  isProduction: boolean  
  ...  
}
```

Rádio

- Requer - Opção de ser uma união de três ou menos cordas.
- Uso - Gere um rádio 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'
  ...
}
```

Suspensa

- Requer - Opção de ser uma união de quatro ou mais cordas.
- 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 matriz de tipos `Tuple`.
- Uso - Gere uma entrada de lista de tuplas.
- Exemplo - `tupleList: Tuple[string, string]>[]`

```
{
  tupleList: Tuple[string, string]>[];
  ...
}
```

Selector

- 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 blueprint, você pode se comunicar com os usuários além das mensagens de validação. Por exemplo, um membro do espaço pode ver uma combinação de opções que produz um esquema que não está claro. Os esquemas personalizados oferecem suporte à capacidade de comunicar mensagens de erro aos usuários invocando a síntese. O blueprint básico implementa uma `throwSynthesisError(...)` função 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.'
})
```

Adicionando componentes do ambiente a um blueprint

O assistente de blueprint personalizado é gerado dinamicamente a partir da `Options` interface exposta por meio do assistente. Os blueprints oferecem suporte à geração de componentes de interface de usuário (UI) a partir de tipos expostos.

Para importar componentes do ambiente Amazon CodeCatalyst Blueprints

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

Tópicos

- [Criação de ambientes de desenvolvimento](#)
- [Exemplos de interface simulada](#)

Criação de ambientes de desenvolvimento

O exemplo a seguir mostra como implantar seu aplicativo na nuvem:

```
export interface Options extends ParentOptions {
  ...
  myNewEnvironment: EnvironmentDefinition{
    thisIsMyFirstAccountConnection: AccountConnection{
      thisIsARole: Role['lambda', 's3', 'dynamo'];
    };
  };
}
```

A interface gera um componente de interface do usuário que solicita um novo ambiente (`myNewEnvironment`) com uma única conexão de conta (`thisIsMyFirstAccountConnection`). Uma função na conexão da conta (`thisIsARole`) também é gerada com `['lambda', 's3', 'dynamo']` os recursos de função 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 uma função. As funções também podem ser anotadas com `@inlinePolicies`. Para ter mais informações, consulte [@inlinePolicy. /caminho/para/policy/file.json](#).

O componente ambiental requer um nome `environmentType` e. O código a seguir é a forma padrão mínima exigida:

```
{
  ...
  "myNewEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION"
  },
}
```

Em seguida, o componente de interface do usuário solicita vários campos. Conforme você preenche os campos, o blueprint ganha uma forma totalmente expandida. Pode ser útil incluir a simulação completa no `defaults.json` arquivo para fins de teste e desenvolvimento.

Exemplos de interface simulada

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"
        ]
    }
}
},
}
```

Adicionando componentes secretos a um blueprint

Os segredos podem ser usados CodeCatalyst para armazenar dados confidenciais que podem ser referenciados em fluxos de trabalho. Você pode adicionar um segredo ao seu plano personalizado e

referenciá-lo em seu fluxo de trabalho. Para ter mais informações, consulte [Configurando e usando segredos em um fluxo de trabalho](#).

Para importar Amazon CodeCatalyst blueprints, digite a região

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-  
component.secrets'
```

Tópicos

- [Criando um segredo](#)
- [Fazendo referência a um segredo em um fluxo de trabalho](#)

Criando um segredo

O exemplo a seguir cria um componente de interface do usuário que solicita que o usuário insira um valor secreto 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"  
  },  
}
```

Fazendo 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 secreto. Para ter mais informações, consulte [Fazendo 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.addAction({
            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 em CodeCatalyst, consulte [Configurando e usando segredos em um fluxo de trabalho](#).

Adicionando componentes da região a um blueprint

O tipo de região pode ser adicionado à `Options` interface do seu blueprint personalizado para gerar um componente no assistente de blueprint. Você pode inserir um ou mais gions da AWS. O tipo de gion pode ser importado do seu blueprint base em seu `blueprint.ts` arquivo. Para obter mais informações, consulte [as regiões da AWS](#).

Para importar Amazon CodeCatalyst blueprints, digite a região

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
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 compatíveis da AWS.

Tópicos

- [Anotações](#)
- [Exemplos de componentes da região](#)

Anotações

As tags JSDoc podem ser adicionadas a cada campo na `Options` interface para personalizar como um campo aparece e se comporta no assistente. Para o tipo de região, as seguintes tags são suportadas:

- A `@displayName` anotação pode ser usada para alterar o rótulo do campo no assistente.

Exemplo: `@displayName AWS Region`

- A `@placeholder` anotação pode ser usada para alterar o espaço reservado do componente de seleção/seleção múltipla.

Exemplo: @placeholder Choose AWS Region

Exemplos de componentes da região

Escolhendo 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']>;
}
```

Escolher 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']>[];
}
```

Escolhendo uma região da AWS

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Region
   */
  region: Region<['*']>;
}
```

Escolher 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 blueprint

Um repositório é usado pela 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 ter 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 o repositório e os componentes do código-fonte do Amazon CodeCatalyst Blueprints

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

Tópicos

- [Adicionando um arquivo](#)
- [Adicionando um arquivo genérico](#)
- [Copiando arquivos](#)
- [Visando vários arquivos](#)
- [Criando um novo repositório e adicionando arquivos](#)

Adicionando um arquivo

Você pode gravar um arquivo de texto em um repositório com a `SourceFile` construção. 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, e ele é especialmente útil para estender o código que os blueprints personalizados podem ter gerado.

Adicionando um arquivo genérico

Você pode gravar bits arbitrários no seu repositório. Você pode ler de um buffer e usar a `File` construção.

```
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());
```

Copiando arquivos

Você pode começar com o código gerado copiando e colando o código inicial e, em seguida, gerando mais código sobre essa base. Coloque o código dentro do `static-assets` diretório e, em seguida, direcione esse código com a `StaticAsset` construção. Nesse caso, o caminho sempre começa na raiz do `static-assets` diretório.

```
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 bigode no arquivo. Pode ser útil para realizar a geração de `copy-and-replace` estilos.

A substituição estática de ativos usa um mecanismo de modelagem de bigode para renderizar os arquivos estáticos que são implantados no repositório de origem gerado. As regras de modelagem do 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 bigode triplo. `{{{name}}}` Para obter mais informações, consulte as regras de [modelagem do bigode](#).

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'
})
```

Visando vários arquivos

Os ativos estáticos oferecem suporte à segmentação global por meio de uma função estática on `StaticAsset` e suas subclasses chamadas `findAll(...)`, que retorna uma lista de ativos estáticos pré-carregados com seus caminhos, conteúdos e muito mais. Você pode encadear a lista com `File` construções para copiar e colar conteúdo no `static-assets` diretório.

```
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());
```

Criando um novo repositório e adicionando arquivos

Você pode usar um componente de repositório para criar um novo repositório em um projeto gerado. Em seguida, 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 nomeado myRepo com um arquivo de origem README.md e um CodeCatalyst fluxo de trabalho na raiz.

Adicionar componentes do fluxo de trabalho a um blueprint

Um fluxo de trabalho é usado pelos CodeCatalyst projetos da Amazon para executar ações com base em gatilhos. Você pode usar componentes do fluxo de trabalho para criar e reunir arquivos YAML do fluxo de trabalho. Para ter mais informações, consulte [Definição de fluxo de trabalho YAML](#).

Para importar componentes de fluxos de trabalho do Amazon CodeCatalyst Blueprints

No seu blueprint.ts arquivo, adicione o seguinte:

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

Tópicos

- [Exemplos de componentes do fluxo de trabalho](#)
- [Conectando-se a um ambiente](#)

Exemplos de componentes do fluxo de trabalho

WorkflowBuilder componente

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 de projeto de fluxo de trabalho

O exemplo a seguir mostra como um componente Projen pode ser usado para gravar um fluxo de trabalho YAML 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 ...});
```

Conectando-se 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 função.

```
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),
});
```

Adicionando componentes do Dev Environments a um blueprint

Ambientes de desenvolvimento gerenciado (MDE) são usados para criar e instalar espaços de trabalho do MDE em CodeCatalyst. O componente gera um `devfile.yaml` arquivo. Para obter mais informações, consulte [Introdução ao Devfile](#) e [Editando um arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento](#)

```
new Workspace(this, repository, SampleWorkspaces.default);
```

Para importar componentes de espaços de trabalho do Amazon CodeCatalyst Blueprints

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

Adicionar componentes de problemas a um blueprint

Em CodeCatalyst, você pode monitorar recursos, tarefas, bugs e qualquer outro trabalho envolvido em seu projeto. Cada peça do trabalho é mantida em um registro distinto, chamado de edição. Cada edição pode ter uma descrição, responsável, status e outras propriedades, que você pode pesquisar, agrupar e filtrar. Você pode visualizar seus problemas usando as visualizações padrão ou criar suas próprias visualizações com filtragem, classificação ou agrupamento personalizados. Para obter mais informações sobre conceitos relacionados a problemas, consulte [Conceitos de questões](#) [Cotas para edições em CodeCatalyst](#) e.

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.

Para importar CodeCatalyst blueprints da Amazon, emite componentes

No seu `blueprint.ts` arquivo, adicione o seguinte:

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

Tópicos

- [Exemplos de componentes de problemas](#)

Exemplos de componentes de problemas

Criando um problema

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
  title: 'myFirstIssue',
  content: 'This is an example issue.',
});
```

Criando 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 ...});
```

Criação de 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'],
});
```

Trabalhando com ferramentas de blueprint e CLI

A [CLI do blueprint](#) fornece ferramentas para gerenciar e trabalhar com seus blueprints personalizados.

Tópicos

- [Trabalhando com ferramentas de projeto](#)
- [Ferramenta de upload de imagens](#)

Trabalhando com ferramentas de projeto

Para trabalhar com as ferramentas do blueprint

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Retome seu ambiente de desenvolvimento. Para ter mais informações, consulte [Retomar um ambiente de desenvolvimento](#).

Se você não tiver um ambiente de desenvolvimento, primeiro deverá criar um. Para ter mais informações, consulte [Criar um ambiente de desenvolvimento](#).

3. Em um terminal em funcionamento, execute o seguinte comando para instalar a CLI do blueprint:

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. No `blueprint.ts` arquivo, 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

Você pode acessar o [CodeCatalyst blueprints GitHub repository](#) para encontrar o nome da ferramenta que deseja usar.

Se você quiser usar a ferramenta de upload de imagens, adicione o seguinte ao seu script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

Exemplos

- Se você quiser usar a função de publicação, adicione o seguinte ao seu script:

```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/publish';
```

- Se você quiser usar a ferramenta de upload de imagens, adicione o seguinte ao seu script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

5. Chame a função.

Exemplos:

- Se você quiser usar a função de publicação, adicione o seguinte ao seu script:

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Se você quiser usar a ferramenta de upload de imagens, adicione o seguinte ao seu script:

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/  
image');
```

Ferramenta de upload de imagens

A ferramenta de upload de imagens oferece a capacidade de carregar sua própria imagem em um bucket do S3 em sua conta da AWS e depois distribuí-la publicamente. CloudFront A ferramenta usa um caminho de imagem no armazenamento local (e o nome opcional do bucket) como entrada e retorna o URL para a imagem que está disponível publicamente. Para obter mais informações, consulte [O que é a Amazon CloudFront?](#) e [o que é o Amazon S3?](#)

Para trabalhar com a ferramenta de upload de imagens

1. Clone o [GitHub repositório de blueprints de código aberto](#) que fornece acesso ao SDK de blueprints e aos exemplos de blueprints. Em um terminal em funcionamento, execute o seguinte comando:

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Execute o comando a seguir para navegar até o GitHub repositório de blueprints:

```
cd codecatalyst-blueprints
```

3. Execute o comando a seguir para instalar dependências:

```
yarn && yarn build
```

4. Execute o comando a seguir para garantir que a versão mais recente da CLI do blueprint esteja instalada:

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Faça login na conta da AWS com o bucket do S3 para o qual você deseja carregar sua imagem. Para obter mais informações, consulte [Configurar a AWS CLI](#) e fazer [login por meio da interface da linha de comando da AWS](#).
6. Execute o seguinte comando na raiz do seu CodeCatalyst repositório para navegar até o diretório com a CLI do blueprint:

```
cd packages/utils/blueprint-cli
```

7. Execute o comando a seguir para carregar sua imagem em um bucket do S3:

```
yarn blueprint upload-image-public <./path/to/your/image>  
  <optional:optional-bucket-name>
```

Um URL para sua imagem é gerado. O URL não estará disponível imediatamente, pois requer algum tempo para que a CloudFront distribuição seja implantada. Verifique o status da distribuição para obter o status de implantação mais recente. Para obter mais informações, consulte [Trabalhando com distribuições](#).

Avaliação de mudanças na interface com testes de instantâneos

Testes de instantâneos gerados em várias configurações do seu blueprint são suportados.

Os blueprints oferecem suporte ao [teste instantâneo](#) em configurações fornecidas por você como autor do blueprint. As configurações são substituições parciais que são mescladas sobre o arquivo `defaults.json` na raiz de um blueprint. Quando o teste de instantâneo é ativado e configurado, o processo de construção e teste sintetiza as configurações fornecidas e verifica se as saídas sintetizadas não foram alteradas em relação ao instantâneo de referência. Para ver o código de teste de snapshot, consulte o repositório de [CodeCatalyst blueprints GitHub](#).

Para habilitar o teste de snapshot

1. No `.projenrc.ts` arquivo, atualize o objeto de entrada `ProjenBlueprint` com os arquivos que você deseja capturar. Por exemplo: .

```
{  
  ....  
  blueprintSnapshotConfiguration: {  
    snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],  
  },  
}
```

2. Sintetize novamente o blueprint para criar TypeScript arquivos em seu projeto de blueprint. Não edite os arquivos de origem, pois eles são mantidos e regenerados pelo Projen. Use o seguinte comando:

```
yarn projen
```

3. Navegue até o `src/snapshot-configurations` diretório para ver o `default-config.json` arquivo 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 `defaults.json` arquivo do projeto, sintetizada e comparada aos instantâneos durante o teste. Use o comando a seguir para testar:

```
yarn test
```

A primeira vez que você usa um comando de teste, a seguinte mensagem é exibida: `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 blueprint para produzir uma saída diferente, execute o comando a seguir para atualizar os instantâneos de referência:

```
yarn test:update
```

Os instantâneos esperam que as saídas sintetizadas sejam constantes entre cada execução. Se o seu blueprint gerar arquivos que variam, você deve excluir esses arquivos do teste de snapshot. Atualize o `blueprintSnapshotConfiguration` objeto do seu objeto `ProjenBlueprint` de entrada para adicionar a `snapshotGlobs` propriedade. A `snapshotGlobs` propriedade é uma matriz de [globs](#) que determina quais arquivos são incluídos ou excluídos da captura instantânea.

Note

Há uma lista padrão de globos. Se você especificar sua própria lista, talvez seja necessário trazer de volta explicitamente as entradas padrão.

Publicar um blueprint personalizado em um espaço

Antes de inserir um blueprint personalizado no catálogo de blueprints do seu espaço, você deve publicá-lo no espaço. Você também pode visualizar o blueprint no CodeCatalyst console antes de publicar. Você pode publicar uma versão prévia ou uma versão normal do seu blueprint.

Important

Se você quiser usar pacotes de blueprint de fontes externas, considere os riscos que podem vir com esses pacotes. Você é responsável pelos blueprints personalizados que adiciona ao seu espaço e pelo código que eles geram.

Tópicos

- [Visualizando e publicando uma versão prévia de um blueprint personalizado](#)
- [Visualizando e publicando uma versão normal de um blueprint personalizado](#)
- [Publicar e aplicar um blueprint personalizado em espaços e projetos especificados](#)

Visualizando e publicando uma versão prévia de um blueprint personalizado

Você pode publicar uma versão prévia do seu blueprint personalizado no seu espaço se quiser adicioná-la ao catálogo de blueprints do seu espaço. Isso permite que você visualize o blueprint 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 0.0.1 versão, 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 seu blueprint personalizado executando o `package.json` arquivo e visualize suas alterações.

Para visualizar e publicar uma versão prévia de um blueprint personalizado

1. Retome seu ambiente de desenvolvimento. Para mais informações, consulte [Retomar um ambiente de desenvolvimento](#).
2. Abra um terminal funcional em seu ambiente de desenvolvimento.
3. (Opcional) Em um terminal em funcionamento, instale as dependências necessárias para seu projeto, caso ainda não as tenha instalado. Use o seguinte comando:

```
yarn
```

4. (Opcional) Se você fez alterações no `.projenrc.ts` arquivo, gere novamente a configuração do seu projeto antes de criar e visualizar seu blueprint. Use o seguinte comando:

```
yarn projen
```

5. Reconstrua e visualize seu blueprint personalizado usando o comando a seguir. Use o comando a seguir:

```
yarn blueprint:preview
```

Navegue até o `See this blueprint at:` link fornecido para visualizar seu plano personalizado. Verifique se a interface do usuário, incluindo o texto, aparece conforme o esperado com base na sua configuração. Se quiser alterar seu blueprint personalizado, você pode editar o `blueprint.ts` arquivo, resintetizar o blueprint e, em seguida, publicar uma versão prévia novamente. Para ter mais informações, consulte [Ressíntese](#).

6. (Opcional) Você pode publicar uma versão prévia do seu blueprint personalizado, que pode então ser adicionada ao catálogo de blueprints do seu espaço. Navegue até o `Enable version [preview version number]` at: link para publicar uma versão prévia em seu espaço.

Você pode emular a criação do projeto sem precisar criar um projeto no CodeCatalyst. Para sintetizar seu projeto, use o seguinte comando:

```
yarn blueprint:synth
```

Um blueprint é gerado na `synth/synth.[options-name]/proposed-bundle/` pasta. Para ter mais informações, consulte [Síntese](#).

Se você estiver atualizando seu blueprint personalizado, use o comando a seguir para resintetizar seu projeto:

```
yarn blueprint:resynth
```

Um blueprint é gerado na `synth/synth.[options-name]/proposed-bundle/` pasta. Para ter mais informações, consulte [Ressíntese](#).

Depois de publicar sua versão prévia, você pode adicionar o blueprint para que os membros do espaço possam usá-lo para criar novos projetos ou aplicar em projetos existentes. Para ter mais informações, consulte [Adicionar um blueprint personalizado a um catálogo espacial](#).

Visualizando e publicando uma versão normal de um blueprint personalizado

Depois de terminar de desenvolver e visualizar seu blueprint personalizado, você poderá visualizar e publicar a nova versão que deseja adicionar ao catálogo de blueprints 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 blueprint, seu blueprint não será automaticamente disponibilizado para ser adicionado ao catálogo de blueprints do seu espaço. Você ainda pode publicar seu blueprint personalizado em seu espaço depois de executar um `yarn` comando.

Para visualizar e publicar um blueprint personalizado

1. Retome seu ambiente de desenvolvimento. Para mais informações, consulte [Retomar um ambiente de desenvolvimento](#).
2. Abra um terminal funcional em seu ambiente de desenvolvimento.
3. • Se você optou por não gerar o fluxo de trabalho de lançamento ao criar seu blueprint, use o seguinte comando:

```
yarn blueprint:release
```

Você ainda pode navegar até o `See this blueprint at:` link fornecido para ver seu plano personalizado.

Publique a versão atualizada do seu blueprint personalizado, que pode então ser adicionada ao catálogo de blueprints do seu espaço. Navegue até o `Enable version [release version number] at:` link para publicar a versão mais recente em seu espaço.

- Se você optou por um fluxo de trabalho de lançamento ao criar seu blueprint, a versão mais recente do blueprint será publicada automaticamente quando as alterações forem enviadas. Use os seguintes comandos:

```
git add .
```

```
git commit -m "commit message"
```



```
git push
```

Depois de publicar sua versão normal, você pode adicionar o blueprint para que os membros do espaço possam usá-lo para criar novos projetos ou aplicar em projetos existentes. Para ter mais informações, consulte [Adicionar um blueprint personalizado a um catálogo espacial](#).

Publicar e aplicar um blueprint personalizado em espaços e projetos especificados

Por padrão, os `blueprint:release` comandos `blueprint:preview` e são publicados no CodeCatalyst espaço em que você criou o blueprint. Se você tiver vários espaços corporativos, também poderá visualizar e publicar o mesmo esquema nesses espaços. Você também pode aplicar um blueprint a um projeto existente de outro espaço.

Para publicar ou aplicar um blueprint personalizado em um espaço especificado

1. Retome seu ambiente de desenvolvimento. Para ter mais informações, consulte [Retomar um ambiente de desenvolvimento](#).
2. Abra um terminal funcional em seu ambiente de desenvolvimento.
3. (Opcional) Instale as dependências necessárias para seu projeto, caso ainda não as tenha instalado. Use o seguinte comando:

```
yarn
```

4. Use a `--space` tag 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 aplicar uma versão prévia de um blueprint 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
```

Visualizando detalhes, versões e projetos de um blueprint personalizado

Você pode visualizar os blueprints personalizados publicados do seu espaço, incluindo os detalhes, as versões e os projetos criados com ou aplicando o blueprint.

Tópicos

- [Visualizando as plantas personalizadas de um espaço](#)
- [Visualizando projetos criados com ou aplicando um blueprint personalizado](#)

Visualizando as plantas personalizadas de um espaço

Para visualizar as plantas personalizadas de um espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja visualizar um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints para visualizar os blueprints do Space. Os detalhes a seguir são exibidos na tabela:
 - Nome - Nome do blueprint personalizado.
 - Status do catálogo - Se o blueprint personalizado é publicado no catálogo de blueprints do espaço.
 - Versão mais recente - A versão mais recente do plano personalizado.
 - Última modificação - A data em que o plano espacial foi atualizado pela última vez.

Visualizando projetos criados com ou aplicando um blueprint personalizado

Para visualizar projetos criados com ou aplicando um blueprint personalizado

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja visualizar um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Na tabela de blueprints do Space, escolha o nome de um blueprint personalizado para visualizar os Projetos usando blueprint e Projetos que não usam tabelas de blueprint.

Adicionar um blueprint personalizado a um catálogo espacial

Depois de publicar um blueprint personalizado em seu espaço, ele pode ser adicionado ao catálogo de blueprints do seu espaço. Se você adicionar um blueprint personalizado ao catálogo de blueprints do seu CodeCatalyst espaço, o blueprint estará disponível para todos os membros do espaço usarem ao criar um projeto ou aplicá-lo a um projeto existente. Antes de adicionar um blueprint personalizado ao catálogo de blueprints do espaço, a permissão de publicação do blueprint 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 [Definindo](#)

[permissões de publicação para um blueprint personalizado](#) e [Publicar um blueprint personalizado em um espaço](#).

Para adicionar uma planta ao catálogo de plantas do espaço

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. O blueprint só pode ser adicionado a partir da ramificação padrão do repositório de origem. Se você desenvolveu o blueprint em uma ramificação de recurso, mescle sua ramificação de recurso com as alterações na ramificação padrão. Crie uma pull request para mesclar quaisquer alterações na ramificação padrão. Para ter mais informações, consulte [Análise de código com pull requests na Amazon CodeCatalyst](#).
3. No CodeCatalyst console, navegue até o painel do espaço com seu blueprint personalizado.
4. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
5. Escolha o nome do blueprint que você deseja adicionar e, em seguida, escolha Adicionar ao catálogo. Se você tiver mais de uma versão, escolha uma versão no menu suspenso Versão do catálogo
6. Escolha Salvar.

Removendo um blueprint personalizado de um catálogo espacial

Um blueprint personalizado pode ser removido do catálogo de blueprints 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 blueprint personalizado de um catálogo espacial, isso não afetará um projeto criado a partir do blueprint ou um projeto que aplicou o blueprint. Os recursos do blueprint não são removidos do projeto.

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o painel do espaço com seu blueprint personalizado
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Escolha o nome do blueprint que você deseja remover e, em seguida, escolha Remover blueprint do catálogo.

Definindo permissões de publicação para um blueprint personalizado

Por padrão, a permissão de um blueprint personalizado é habilitada se uma versão do fluxo de trabalho for gerada durante a criação do projeto. Quando as permissões de publicação estão habilitadas, o blueprint pode ser publicado no espaço. Você pode desativar a permissão para que o blueprint não possa ser publicado. Quando a permissão é desativada, o fluxo de trabalho de lançamento gerado durante a criação do blueprint não é executado. Novas alterações em um blueprint não podem ser publicadas a menos que as permissões do blueprint estejam habilitadas.

Important

Para ativar ou desativar as permissões de publicação de um projeto de blueprint, você deve ter a função de administrador do Space.

Para definir as permissões de publicação de um projeto blueprint

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja gerenciar as permissões de publicação de um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Escolha a guia Permissões de publicação do projeto para ver as permissões de publicação de todos os blueprints do seu espaço.
5. Escolha o blueprint que você deseja gerenciar e, em seguida, escolha Habilitar ou Desabilitar para alterar as permissões de publicação. Se você estiver habilitando as permissões, revise os detalhes da alteração de permissão e escolha Habilitar publicação de blueprint para confirmar a alteração.

Alterando as versões do catálogo para um blueprint personalizado

Como autor do blueprint, você pode gerenciar a versão que deseja publicar no catálogo de blueprints do espaço. Alterar a versão do catálogo de um blueprint não afeta os projetos que estão usando uma versão diferente do blueprint.

Para gerenciar uma versão personalizada do blueprint

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

2. No CodeCatalyst console, navegue até o espaço em que você deseja alterar a versão de um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Na tabela de blueprints do Space, escolha o botão de rádio para o blueprint personalizado que você deseja gerenciar.
5. Escolha Criar versão do catálogo e, em seguida, escolha a versão a no menu suspenso Versão do catálogo.
6. Escolha Salvar.

Excluindo um blueprint ou uma versão personalizada publicada

Quando você exclui a versão de um blueprint personalizado ou o próprio 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 uma versão do blueprint ou o blueprint, os membros do projeto não conseguirão acessar os recursos do projeto, e qualquer fluxo de trabalho solicitado por repositórios de origem de terceiros será interrompido.

Note

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

Se uma versão do blueprint for publicada no catálogo de blueprints do espaço, escolha uma nova versão para o catálogo antes de excluir a versão publicada.

Para excluir a versão do catálogo de um blueprint personalizado

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja excluir a versão do catálogo de um blueprint personalizado.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Escolha o nome do blueprint com a versão do catálogo que você deseja excluir.
5. Escolha o botão de rádio para a versão do catálogo que você deseja excluir e, em seguida, escolha Excluir versão.

6. Revise os detalhes e, em seguida, escolha outra versão do blueprint no menu suspenso Escolher uma nova versão do catálogo de blueprint.
7. Entre de `lete` para confirmar a exclusão da versão do catálogo de blueprints.
8. Escolha Excluir.

Se uma versão do blueprint não estiver no catálogo do blueprint do espaço, você poderá excluí-la sem escolher uma nova versão.

Para excluir uma versão personalizada do blueprint

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No CodeCatalyst console, navegue até o espaço em que você deseja excluir uma versão personalizada do blueprint.
3. No painel do espaço, escolha a guia Configurações e, em seguida, escolha Blueprints.
4. Escolha o nome do blueprint com a versão que você deseja excluir.
5. Escolha o botão de rádio para a versão que você deseja excluir e, em seguida, escolha Excluir versão.
6. Digite `delete` para confirmar a exclusão da versão do blueprint.
7. Escolha Excluir.

A exclusão de um blueprint do catálogo de blueprints do espaço exclui todas as versões do blueprint. Os projetos do espaço que estão usando o blueprint não são afetados pela exclusão.

Para excluir uma versão personalizada do blueprint

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, em seguida, escolha Blueprints.
4. Na tabela de blueprints do Space, escolha o botão de rádio do blueprint personalizado que você deseja excluir e, em seguida, escolha Excluir blueprint.
5. Digite `delete` para confirmar a exclusão do blueprint personalizado.
6. Escolha Delete (Excluir).

Lidando com dependências, incompatibilidades e ferramentas

Tópicos

- [Adicionando dependências](#)
- [Lidando com incompatibilidades de tipos de dependência](#)
- [Usando yarn e npm](#)
- [Atualizando ferramentas e componentes](#)

Adicionando dependências

Como autor do blueprint, talvez você precise adicionar pacotes ao seu blueprint, como `@amazon-codecatalyst/blueprint-component.environments`. Você precisa atualizar o `projen.ts` arquivo com esse pacote e, em seguida, regenerar a configuração do seu projeto com o [Projen](#). O Projen atua como o modelo de projeto para cada base de código do blueprint, o que fornece a capacidade de enviar atualizações de ferramentas compatíveis com versões anteriores, alterando a forma como o modelo renderiza os arquivos de configuração. O `package.json` arquivo é um arquivo parcialmente pertencente ao modelo 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 `projen.ts` arquivo, navegue até a seção `deps`.
2. Adicione a dependência que você deseja usar em seu blueprint.
3. Use o comando a seguir para regenerar a configuração do seu projeto:

```
yarn projen && yarn
```

Lidando 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 ocorre 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 vendidos pelo blueprint 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 `package.json` arquivo 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
```

Usando yarn e npm

As plantas usam [fios 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, você pode remover o `package-lock.json` arquivo gerado e garantir que seu `.projenrc.ts` arquivo esteja atualizado com as dependências de que você precisa. Você regenera a configuração do seu projeto com o Projen.

Use o seguinte para se regenerar 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
```

Atualizando ferramentas e componentes

Ocasionalmente, talvez você queira atualizar suas ferramentas e componentes para trazer novos recursos disponíveis. É recomendável manter todos os componentes na mesma versão, a menos que 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.

Usando o Yarn workspace monorepo

Use o comando a seguir para atualizar utilitários e componentes da raiz do repositório de um blueprint personalizado:

```
yarn upgrade @amazon-codecatalyst/*
```

Use o comando a seguir se você não estiver usando um monorepo:

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Outras opções que você pode usar para atualizar ferramentas e componentes:

- Use o `npm view @caws-blueprint-component/<some-component>` para obter a versão mais recente.
- Aumente manualmente para a versão mais recente definindo a versão em seu arquivo `package.json` e usando o seguinte comando: `yarn`. Todos os componentes e utilitários devem ter a mesma versão.

Contribua

O kit de desenvolvimento de software (SDK) blueprints é uma biblioteca de código aberto com a qual você pode contribuir. Como colaborador, considere as diretrizes de contribuição, o feedback e os defeitos. Para obter mais informações, consulte o [GitHub repositório de blueprints](#).

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 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 do
- 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, talvez você queira 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, o código de teste.* O Dev Environment é aberto. Li adiciona rapidamente o código, depois confirma e envia a ramificação com as alterações no repositório de origem. CodeCatalyst Em seguida, Li cria uma pull request. Como parte da criação dessa pull request, 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). Esses PATs estão associados à sua identidade de CodeCatalyst usuário. Para ter mais informações, consulte [Conceda 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 adicionar 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 ter mais informações, consulte [Introdução aos fluxos de trabalho](#).

Tópicos

- [Conceitos do 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 do 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 de ter um projeto, você pode adicionar, atualizar ou remover usuários e recursos, personalizar o painel do projeto e monitorar o progresso do trabalho da sua equipe. Você pode ter vários projetos em um espaço.

Os repositórios de origem são específicos para os projetos nos quais você os cria ou os vincula em um espaço. Você não pode compartilhar um repositório entre projetos e não pode vincular um repositório a mais de um projeto em um espaço. Os usuários com a função 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 essas funções. Para ter mais informações, consulte [Concedendo acesso com funções de usuário](#).

Repositórios de origem

Um repositório de origem é onde você armazena com segurança o código e os 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. Você pode 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 aplicativos incluídas em seu ambiente de desenvolvimento são definidas por um devfile no repositório de origem do seu projeto. Se você não tiver um devfile em seu repositório de origem, um devfile padrão será aplicado automaticamente. O devfile 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 pode optar por criar uma nova 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. Os PATs representam você CodeCatalyst e você pode gerenciá-los nas 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, certifique-se de armazená-los com segurança em seu computador local. Por padrão, os PATs expiram após um ano.

Ao trabalhar com ambientes de desenvolvimento integrados (IDEs), as PATs equivalem a uma senha do Git. Forneça o PAT quando for solicitada uma senha ao configurar seu IDE para trabalhar com um repositório Git. Para obter mais informações sobre como conectar seu IDE a um repositório baseado em Git, consulte a documentação do seu IDE.

Ramificações

Uma ramificação é um ponteiro ou referência para um commit no Git e no CodeCatalyst. Você pode usar filiais para organizar seu trabalho. Por exemplo, você pode usar ramificações para trabalhar em uma versão nova ou diferente dos arquivos sem afetar os arquivos em outras ramificações. Você pode usar ramificações 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 main. A ramificação principal é a ramificação padrão do 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 amostra, 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 para você como primeiro commit e uma ramificação padrão será criada para você 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 ter mais informações, consulte [Gerenciando 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

Um commit 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. O commit inclui informações sobre a alteração, incluindo a identidade do usuário que fez a alteração, a hora e a data da alteração, o título do commit e qualquer mensagem incluída sobre a alteração. Para ter 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 aos commits para identificar commits específicos.

Solicitações pull

Uma pull request é a principal forma 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 pull requests para revisar as alterações no código de forma colaborativa em busca de pequenas alterações ou correções, adições de recursos importantes ou novas versões do seu software lançado. Em uma pull request, 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 individuais de código, bem como comentários sobre a pull request em geral.

Tip

Enquanto você cria uma pull request, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a pull request for criada, a diferença exibida será entre a revisão da pull request escolhida e o commit que foi a ponta da ramificação de destino quando você criou a pull request. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

Revisões

Uma revisão é uma versão atualizada de uma pull request. Cada push para a ramificação de origem de uma pull request cria uma revisão que contém as alterações feitas nos commits incluídos nesse push. Você pode ver as diferenças entre as revisões de um pull request, além das diferenças entre as ramificações de origem e de destino. Para ter mais informações, consulte [Análise de código com pull requests na Amazon CodeCatalyst](#).

Fluxos de trabalho

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint 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 blueprint, você 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 melhor prática, recomendamos que você use as versões mais recentes do Git e de outros softwares.

Note

Se você usa Dev Environments, não precisa instalar o Git. Uma versão recente do Git está incluída no seu ambiente de desenvolvimento.

Informações de compatibilidade de versão para CodeCatalyst

Componente	Version (Versão)
Git	mais recente

Instale o Git

Para trabalhar com arquivos, commits, branches e outras informações em repositórios de origem de um cliente Git sem um IDE, instale o Git em sua máquina local.

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

Crie um token de acesso pessoal

Para clonar repositórios de origem em sua máquina local ou em seu IDE preferido, você deve criar 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 seu PAT.
3. Em Data de expiração, deixe a data padrão ou escolha o ícone do calendário para selecionar uma data personalizada. A data de expiração é padronizada para um ano a partir 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. Você não pode 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 aprender a:

- Visualize os repositórios de origem de um projeto e navegue por seus conteúdos
- Crie um ambiente de desenvolvimento com uma nova ramificação onde você pode trabalhar no código
- Altere um arquivo, confirme e envie suas alterações
- Crie uma pull request e revise suas alterações de código com outros membros do projeto
- Veja o fluxo de trabalho do seu projeto, crie e teste automaticamente as alterações na ramificação de origem da pull request
- Mescle suas alterações da ramificação de origem na ramificação de destino e feche a pull request
- Veja as alterações mescladas criadas e implantadas automaticamente

Para aproveitar ao máximo este tutorial, convide outras pessoas para o seu projeto para que vocês possam trabalhar juntos em uma pull request. 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 introdução](#).

Criando um projeto com um blueprint

Criar um projeto é o primeiro passo para poder trabalhar em conjunto. Você pode usar um blueprint para criar seu projeto, que também criará um repositório de origem com código de amostra e um fluxo de trabalho que criará e implantará automaticamente seu código quando você o alterar. Neste tutorial, mostraremos um projeto criado com o esquema de aplicativo de página única, mas você pode seguir os procedimentos de qualquer projeto com um repositório de origem. Certifique-se de escolher uma função do IAM ou adicionar uma função do IAM se

Se você não tiver uma como parte da criação do projeto. Recomendamos que você use a função `CodeCatalystWorkflowDevelopmentRole-spaceName` de serviço para esse projeto.

Se você já tem um projeto, pode pular para [Visualizando 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 essa função e precisa de um projeto para trabalhar neste tutorial, peça a alguém com uma dessas funções que crie um projeto para você e adicione você ao projeto criado. Para ter mais informações, consulte [Concedendo acesso com funções de usuário](#).

Para criar um projeto com um blueprint

1. No CodeCatalyst console, navegue até o espaço em que você deseja criar um projeto.
2. No painel do espaço, escolha Criar projeto.
3. Escolha Começar com um blueprint.
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. Como alternativa, você pode optar por armazenar o código-fonte do blueprint em um repositório de terceiros. Para ter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Siga um destes procedimentos, dependendo do provedor de repositório terceirizado 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, escolha Bitbucket como provedor de repositório e, em seguida, escolha o espaço de trabalho do Bitbucket em que você deseja armazenar o código-fonte criado pelo blueprint.


7. Em Recursos do projeto, configure os parâmetros do blueprint. Dependendo do blueprint, você pode ter a opção de nomear o nome do repositório de origem.
8. (Opcional) Para visualizar arquivos de definição com atualizações com base nas seleções de parâmetros do projeto que você fez, escolha Exibir código ou Exibir fluxo de trabalho em Gerar visualização do projeto.
9. (Opcional) Escolha Exibir detalhes no cartão do blueprint para ver detalhes específicos sobre o blueprint, como uma visão geral da arquitetura do blueprint, conexões e permissões necessárias e o tipo de recursos que o blueprint 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 pull requests. Opcionalmente, você pode escolher 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 [Convidar um usuário para um projeto](#).

Visualizando os repositórios de um projeto

Como membro de um projeto, você pode visualizar os repositórios de origem do projeto. Você também pode optar por criar repositórios adicionais. Se alguém com a função de administrador do Space tiver instalado e configurado os GitHub repositórios ou a extensão do Bitbucket, você também poderá adicionar links para repositórios de terceiros nas GitHub contas ou nos espaços de trabalho do Bitbucket configurados para a extensão. Para obter mais informações, consulte [Criando 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`.

Para navegar até os repositórios de origem de um projeto

1. Navegue até seu projeto e faça o seguinte:
 - Na página de resumo do seu projeto, escolha o repositório que você deseja na lista e, em seguida, escolha Exibir repositório.
 - No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem. Em Repositórios de origem, escolha 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 pull requests e fluxos de trabalho. Por padrão, o conteúdo da ramificação padrão é exibido. Você pode alterar a exibição escolhendo uma ramificação diferente na lista suspensa.

A página de visão geral do repositório inclui informações sobre os fluxos de trabalho e pull requests configurados para as ramificações desse repositório e seus arquivos. Se você acabou de criar o projeto, os fluxos de trabalho iniciais para criar, 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 seu status escolhendo o número abaixo de Fluxos de trabalho relacionados, mas isso abre a página Fluxos de trabalho em CI/CD. Para este tutorial, fique na página de visão geral e explore o código no repositório. O conteúdo do README.md arquivo é renderizado nesta página abaixo dos arquivos do repositório. Em Arquivos, o conteúdo da ramificação padrão é mostrado. Você pode alterar a visualização do arquivo para mostrar o conteúdo de outra ramificação, se você tiver uma. A `.codecatalyst` pasta contém o código usado para outras partes do projeto, como arquivos YAML do fluxo de trabalho.

Para visualizar o conteúdo das pastas, escolha a seta ao lado do nome da pasta para expandi-la. Por exemplo, escolha a seta ao lado de `src` para visualizar os arquivos do aplicativo Web de página

única contido nessa pasta. Para visualizar o conteúdo de um arquivo, escolha-o na lista. Isso abrirá Exibir arquivos, onde você poderá navegar pelo conteúdo de vários arquivos. Você também pode editar arquivos únicos no console, mas para editar vários arquivos, convém criar 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 criar e testar seu código quando você criar uma pull request posteriormente neste tutorial.

Tip

Se você decidir clonar um repositório localmente em vez de ou além de usar um ambiente de desenvolvimento, certifique-se de ter o Git em seu computador local ou de que seu IDE inclua o Git. Para ter 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. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código, escolha Repositórios de origem e escolha o repositório para o qual você deseja criar um ambiente de desenvolvimento.
4. Na página inicial do repositório, escolha Create Dev Environment.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para 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 Dev Environment, você pode editar arquivos, confirmar suas alterações e enviar suas alterações para a **test** ramificação. Para este tutorial, edite o conteúdo entre as `<p>` tags no `App.tsx` arquivo na `src` pasta 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. `src`Expand a e abra `App.tsx`.
2. Faça uma alteração no texto dentro das `<p>` tags.
3. Salve o arquivo e, em seguida, confirme e envie suas alterações usando o menu Git. Como alternativa, na janela do terminal, confirme e envie suas alterações com os `git commit` `git push` comandos e.

```
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.

Criação de uma solicitação pull

Você pode usar pull requests para revisar as alterações no código de forma colaborativa em busca de pequenas alterações ou correções, adições de recursos importantes ou novas versões do seu

software lançado. Neste tutorial, você criará uma pull request para revisar as alterações feitas na ramificação de *teste* em comparação com a ramificação principal. Criar uma pull request em um projeto criado com um modelo também iniciará a execução dos fluxos de trabalho associados, se houver.

Para criar uma pull request

1. Navegue até o projeto.
2. Execute um destes procedimentos:
 - No painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha Create pull request.
 - Na página inicial do repositório, escolha Mais e, em seguida, escolha Criar pull request.
 - Na página do projeto, escolha Criar pull request.
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 pull request na página principal do repositório.
4. Na ramificação de destino, escolha a ramificação na qual mesclar o código depois que ele for revisado.
5. Na ramificação de origem, escolha a ramificação que contém o código confirmado.
6. Em Título do pull request, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
7. (Opcional) Na descrição da pull request, forneça informações como um link para problemas ou uma descrição de suas 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 ao pull request.

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).


8. (Opcional) Em Problemas, escolha Vincular problemas e, em seguida, escolha um problema na lista ou insira sua ID. Para desvincular um problema, escolha o ícone de desvinculação.

9. (Opcional) Em Revisores obrigatórios, escolha Adicionar revisores obrigatórios. Escolha na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que o pull request possa ser mesclado na ramificação de destino.

 Note

Você não pode adicionar um revisor como revisor obrigatório e como revisor opcional. Você não pode se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, escolha Adicionar revisores opcionais. Escolha na lista de membros do projeto para adicioná-los. Os revisores opcionais não precisam aprovar as alterações como um requisito para que a pull request possa ser incorporada à ramificação de destino.
11. Analise as diferenças entre as filiais. A diferença exibida em uma pull request 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 pull request 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 você estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Criar.

 Note

Depois de criar a pull request, você pode adicionar comentários. Os comentários podem ser adicionados à pull request ou a linhas individuais em arquivos, bem como à pull request 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 pull request escolhendo Visão geral e, em seguida, revisando as informações na área de detalhes da pull request em Execuções do fluxo de trabalho. Para visualizar a execução do fluxo de trabalho, escolha a execução.

i Tip

Se você nomeou sua ramificação de forma diferente **develop**, um fluxo de trabalho não será executado automaticamente para criar e testar suas alterações. Se você quiser configurar isso, edite o arquivo YAML para o `onPullRequestBuildAndTest` fluxo de trabalho. Para ter mais informações, consulte [Criação de um fluxo de trabalho](#).

Você pode comentar sobre essa pull request 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 pull request. Para obter mais informações [Analisando uma pull request](#), consulte [Atualizando uma pull request](#) [Análise de código com pull requests na Amazon CodeCatalyst](#), [Visualizando o status e os detalhes da execução do fluxo de trabalho](#) e.

Mesclando uma pull request

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 pull request 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 `onPushToMainDeploy` fluxo de trabalho.

Para mesclar uma pull request (console)

1. Em Pull requests, escolha a pull request que você criou na etapa anterior. No pull request, escolha Mesclar.
2. Escolha entre as estratégias de mesclagem disponíveis para o pull request. Opcionalmente, selecione ou desmarque a opção de excluir a ramificação de origem após mesclar a pull request e, em seguida, escolha Mesclar. Após a conclusão da mesclagem, o status da pull request muda para Mesclado e não aparece mais na visualização padrão das pull requests. A visualização padrão mostra pull requests com o status Aberto. Você ainda pode visualizar uma pull request mesclada, mas não pode aprová-la ou alterar seu status.

i 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 pull request é indicado por um ícone de relógio em Visão geral na área de detalhes da pull request. Se todos os revisores obrigatórios aprovaram o pull request, 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 obter mais informações, consulte [Mesclando uma pull request \(Git\)](#) e sua documentação do Git.

Visualizando o código implantado

Agora é hora de ver o código implantado originalmente que estava na ramificação padrão e suas alterações mescladas quando elas são criadas, testadas e implantadas automaticamente. Para fazer isso, você pode retornar à página de visão geral do repositório e escolher o número ao lado do ícone de fluxos de trabalho relacionados ou, no painel de navegação, escolher CI/CD e, em seguida, escolher Fluxos de trabalho.

Para visualizar o código implantado

1. Em Fluxos de trabalho, em `monPushToMainDeployPipeline`, expanda Execuções recentes.

Note

Esse é o nome padrão do fluxo de trabalho para projetos criados com o blueprint de aplicativo de página única.

2. A execução mais recente é aquela iniciada pela confirmação do pull request mesclado na `main` ramificação e provavelmente mostrará o status Em andamento. Escolha uma execução concluída com êxito na lista para abrir os detalhes dessa execução.
3. Escolha Variáveis. Copie o valor para `appURL`. Esse é o URL do aplicativo web de página única implantado. Abra uma nova guia do navegador e cole o valor para ver o código criado 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 que você abriu para visualizar o aplicativo da web e atualizar seu navegador. Você deve ver as alterações feitas na sua pull request mesclada.

Como excluir recursos

Depois de explorar o trabalho com um repositório de origem e uma pull request, talvez você queira remover todos os recursos desnecessários. Você não pode excluir pull requests, mas pode fechá-las. Você pode excluir qualquer ramificação criada.

Se você não precisar mais do repositório de origem ou do projeto, também poderá excluir esses recursos. Para ter mais informações, consulte [Excluindo um repositório de origem](#) e [Excluir um projeto](#).

Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst

Um repositório de origem é onde você armazena com segurança o código e os arquivos do seu projeto. Ele também armazena seu histórico de origem, desde a primeira confirmação até as últimas alterações. Se você escolher um blueprint 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 que você criar para um projeto. Você também pode optar por vincular um repositório Git hospedado no GitHub ou no Bibucket 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, você deve instalar 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 ter mais informações, consulte [Instalando 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 você 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

- [Criando um repositório de origem](#)
- [Vinculando um repositório de origem](#)
- [Visualizando um repositório de origem](#)
- [Editando as configurações de um repositório de origem](#)
- [Clonando um repositório de origem](#)
- [Excluindo um repositório de origem](#)

Criando 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 escolher criar repositórios para um projeto. Esses repositórios conterão um único arquivo, um arquivo README.md, que você pode editar ou excluir a qualquer momento. Dependendo de suas escolhas ao criar um repositório de origem, os repositórios também podem conter um `.gitignore` arquivo.

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 obter 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. (Opcional) Adicione um `.gitignore` arquivo para o tipo de código que você planeja enviar.
8. Selecione Create (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 alterar ou excluir a ramificação padrão.

Vinculando 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 a função de administrador do Space podem instalar extensões. Depois que a extensão estiver instalada, você poderá vincular-se a repositórios configurados para acesso por essa extensão. Para obter mais informações, consulte [Instalando uma extensão em um espaço](#) ou siga [Vinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

⚠ 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

⚠ 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.

ℹ Note

- Um repositório GitHub ou Bitbucket só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados GitHub ou do Bitbucket com projetos. CodeCatalyst
- Você não pode vincular um repositório GitHub ou um repositório do Bitbucket 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.

Para vincular um repositório de origem

1. Navegue até o projeto ao qual você deseja vincular um repositório.

Note

Antes de vincular um repositório, um usuário com a função de administrador do Space deve primeiro instalar a extensão para o provedor que hospeda o repositório. Para ter mais informações, consulte [Instalando uma extensão em um espaço](#).

2. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
3. Escolha Adicionar repositório e, em seguida, escolha Vincular repositório.
4. No menu suspenso Provedor de repositório, escolha um dos seguintes provedores de repositório terceirizados: ou Bitbucket. GitHub
5. Siga um destes procedimentos, dependendo do provedor de repositório terceirizado que você escolheu 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 que você deseja vincular: CodeCatalyst
 - Para fornecer acesso a todos os repositórios atuais e futuros, escolha 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 do espaço de trabalho do Bitbucket, escolha 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 repositório GitHub ou 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Visualizando 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 filiais no repositório
- O número de pull requests abertos para o repositório
- O número de fluxos de trabalho relacionados para o repositório
- Os arquivos e pastas na ramificação padrão ou na ramificação que você escolher
- 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 os commits, branches e pull requests 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 ver informações sobre repositórios vinculados, escolha o link na lista de repositórios para abrir esse repositório no serviço que o hospeda.

Para navegar até os repositórios de origem de um projeto

1. Navegue até seu projeto e faça o seguinte:

- Na página de resumo do seu projeto, escolha o repositório desejado na lista e escolha Exibir repositório.
- No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem. Em Repositórios de origem, escolha 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 pull requests e fluxos de trabalho. Por padrão, o conteúdo da ramificação padrão é exibido. Você pode alterar a exibição escolhendo uma ramificação diferente na lista suspensa.

Tip

Você também pode navegar rapidamente até os repositórios do seu projeto escolhendo Ver código do projeto na página de resumo do projeto.

Editando 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 integrantes do projeto a entender para que o repositório é usado e a aplicar as melhores práticas e processos usados pela equipe.

Note

Você não pode 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, você deve editá-las no provedor que hospeda o repositório vinculado. Para obter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.

Para 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, escolha o repositório desejado na lista e, em seguida, escolha Exibir repositório. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem. Escolha o nome do repositório na lista de repositórios de origem do projeto.
3. Na página de visão geral do repositório, escolha Mais e, em seguida, selecione Gerenciar configurações.
4. Faça uma ou mais das coisas a seguir:
 - Edite a descrição do repositório e escolha Salvar.
 - Para alterar a ramificação padrão do repositório, em Ramificação padrão, escolha Editar. Para ter mais informações, consulte [Gerenciando a ramificação padrão de um repositório](#).
 - Para adicionar, remover ou alterar uma regra sobre quais funções do projeto têm permissão para realizar determinadas ações em uma ramificação, em Regras de ramificação, escolha Editar. Para ter mais informações, consulte [Gerencie as ações permitidas para uma ramificação com regras de ramificação](#).
 - Para adicionar, remover ou alterar uma regra de aprovação para mesclar pull requests em uma ramificação, em Regras de aprovação, escolha Editar. Para ter mais informações, consulte [Gerenciando requisitos para mesclar uma pull request com regras de aprovação](#).

Clonando um repositório de origem

Para trabalhar de forma eficaz com vários arquivos, ramificações e confirmações em repositórios de origem, clone o repositório de origem em seu computador local e use 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 no código. A criação de um ambiente de desenvolvimento clona automaticamente o repositório e a ramificação que você especifica no ambiente de desenvolvimento.

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, em seguida, clone-o. Para obter mais informações, consulte a documentação do serviço que hospeda o repositório vinculado.


Para criar um ambiente de desenvolvimento a partir de um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
3. Escolha o repositório de origem em que você deseja trabalhar no código.
4. Escolha Criar ambiente de desenvolvimento.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para mais informações.
6. Execute um destes procedimentos:
 - Escolha Trabalhar na ramificação existente e, em seguida, escolha uma ramificação no menu suspenso Ramificação existente.
 - Escolha Trabalhar em 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 suspenso Criar ramificação.
7. Opcionalmente, adicione um nome para o Dev Environment ou edite sua configuração.
8. Selecione Create (Criar).

Para clonar um repositório de origem

1. Navegue até o projeto.

2. Na página de resumo do seu projeto, escolha o repositório que você deseja na lista e, em seguida, escolha Exibir repositório. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem. Escolha 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. Escolha Clonar repositório. Copie o URL do clone para o repositório.


 Note

Se você não tiver um token de acesso pessoal (PAT), escolha Criar token. Copie o token e salve-o em um local 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 git clone comando com a URL do clone após o comando. Por exemplo: .

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Quando for solicitada uma senha, cole o PAT que você salvou anteriormente.

 Note

Se seu sistema operacional fornece gerenciamento de credenciais ou se você instalou um sistema de gerenciamento de credenciais, você só precisa fornecer o PAT uma vez. Caso contrário, talvez seja necessário fornecer o PAT para cada operação do Git. Como prática recomendada, certifique-se de que seu sistema de gerenciamento de credenciais armazene com segurança seu PAT. 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 a URL. Quando solicitado a fornecer uma senha, forneça o PAT.

Excluindo 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 depois que o repositório for excluído. 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 dados dele ou enviar dados para ele. A exclusão de um repositório de origem não exclui nenhuma cópia local desse repositório (repositórios locais). Para excluir um repositório local, use as ferramentas de gerenciamento de arquivos e diretórios do seu computador local.

Note

Você não pode excluir um repositório vinculado no CodeCatalyst console. Para excluir um repositório vinculado, escolha o link na lista de repositórios para abrir esse repositório no serviço que o hospeda e, em seguida, 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 [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Para excluir um repositório de origem

1. Navegue até o projeto que contém o repositório de origem que você deseja excluir.
2. Na página de resumo do seu projeto, escolha o repositório que você deseja na lista e, em seguida, escolha Exibir repositório. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem. Escolha o nome do repositório na lista de repositórios de origem do projeto.
3. Na página inicial do repositório, escolha Mais e, em seguida, escolha Excluir repositório.

4. Revise a ramificação, a pull request 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 você quiser continuar, digite delete e, em seguida, escolha 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. Você pode usar ramificações 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 funções específicas 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 blueprint, um arquivo README.md é adicionado para você como um primeiro commit 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

Você não pode 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 filiais.

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 main. Esse é o nome usado para a ramificação padrão em todos os exemplos desse guia.

Tópicos

- [Criando uma filial](#)
- [Gerenciando a ramificação padrão de um repositório](#)
- [Gerencie as ações permitidas para uma ramificação com regras de ramificação](#)
- [Comandos Git para filiais](#)
- [Visualizando filiais e detalhes](#)
- [Excluindo uma ramificação](#)

Criando uma filial

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 ter mais informações, consulte [Criar um ambiente de desenvolvimento](#).

Você também pode usar o Git para criar ramificações. Para ter mais informações, consulte [Comandos comuns do Git para filiais](#).

Para criar uma ramificação (console)

1. No CodeCatalyst console, navegue até o projeto em que seu repositório de origem reside.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.
3. Escolha o repositório em que você deseja criar uma ramificação.
4. Na página de visão geral do repositório, escolha Mais e, em seguida, escolha Criar ramificação.
5. Insira um nome para a filial.
6. Escolha uma ramificação a partir da qual criar a ramificação e, em seguida, escolha Criar.

Gerenciando 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 blueprint para criar um projeto, a ramificação padrão no repositório de origem criado para esse projeto será denominada main. O conteúdo da ramificação padrão é exibido automaticamente na página de visão geral desse repositório.

A ramificação padrão é tratada de forma um pouco diferente de todas as outras ramificações em um repositório de origem. Ele tem um rótulo especial ao lado do nome, Padrão. A ramificação padrão é aquela usada como ramificação base ou padrão nos repositórios locais (repositórios) quando os usuários clonam o repositório em 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.


Para visualizar e alterar a ramificação padrão de um repositório

1. Navegue até o projeto em que seu repositório reside.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja visualizar as configurações, incluindo a ramificação padrão.

3. Na página de visão geral do repositório, escolha Mais e, em seguida, escolha 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 Filiais.
5. Para alterar a ramificação padrão, escolha Editar.

 Note

Você deve ter a função de administrador do projeto no projeto para alterar a ramificação padrão.

6. Escolha o nome da ramificação que você deseja tornar a ramificação padrão na lista suspensa e escolha Salvar.

Gerencie 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 dessa função. Você pode alterar quais ações são permitidas para uma ramificação específica configurando regras de ramificação. As regras de ramificação são baseadas na função que um usuário tem em seu projeto. Você pode optar por limitar algumas ações predefinidas, como enviar confirmações para uma ramificação, para usuários com uma função específica em um projeto. Isso pode ajudá-lo a proteger ramificações específicas em um projeto, limitando quais funções podem realizar determinadas ações. Por exemplo, se você configurar uma regra de ramificação para permitir que somente usuários com a função de administrador do projeto se mesquem ou enviem para essa ramificação, os usuários com outras funções no projeto não poderão fazer alterações no código dessa ramificação.

Você deve considerar cuidadosamente todas as implicações da criação de uma regra para uma ramificação. Por exemplo, se você optar por limitar os envios para uma ramificação aos usuários com a função de administrador do projeto, os usuários com a função de Colaborador não poderão criar ou 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 para o YAML. Como prática recomendada, teste todas as regras de ramificação depois de criá-las para garantir que elas não tenham nenhum impacto que você não pretendia. Você também pode usar regras de ramificação em conjunto com regras de aprovação para pull requests. Para ter mais informações, consulte [Gerenciando requisitos para mesclar uma pull request 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. Você não pode criar regras de ramificação para repositórios vinculados.

Você só pode criar regras de ramificação que sejam mais restritivas do que as permissões padrão para a função. Você não pode criar regras de ramificação que sejam mais permissivas do que o permitido pela função de um usuário no projeto. Por exemplo, você não pode criar uma regra de ramificação que permita que usuários com a função 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 funções têm quais permissões, consulte [Concedendo acesso com funções de usuário](#). Personalizado indica que uma ou mais ações de ramificação têm ações que têm uma lista específica de funções permitidas para realizar essa ação, diferente das permissões padrão concedidas pela função 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 a função de administrador do projeto excluam essa ramificação.

A tabela a seguir lista as ações e as configurações padrão das funções autorizadas a realizar essas ações em uma ramificação.

Ações e funções da ramificação

Ação da filial	Funções autorizadas a realizar essa ação quando nenhuma regra de ramificação é aplicada
Mesclar com a ramificação (isso inclui mesclar uma pull request com a ramificação)	Administrador do projeto, Colaborador
Empurre até o galho	Administrador do projeto, Colaborador
Excluir a ramificação	Administrador do projeto, Colaborador
Excluir a ramificação (ramificação padrão)	Não permitido

Você não pode excluir regras de ramificação, mas pode atualizá-las para permitir ações de todas as funções que teriam permissão para 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.

Para visualizar e editar regras de ramificação para um repositório

1. Navegue até o projeto em que seu repositório reside.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja visualizar as regras de ramificação.

3. Na página de visão geral do repositório, escolha Branches.
4. Na coluna Regras de ramificação, visualize o status das regras para cada ramificação do repositório. O 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 essas funções 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 funções.

Para ver as especificações das regras de ramificação de uma ramificação, escolha a palavra Padrão ou Personalizada ao lado da ramificação que você deseja revisar.

5. Para criar ou alterar uma regra de ramificação, escolha Gerenciar configurações. Na página de configurações do repositório de origem, em Regras de ramificação, escolha Editar.
6. Em Ramificação, escolha o nome da ramificação para a qual você deseja configurar uma regra na lista suspensa. Para cada um dos tipos de ação permitidos, escolha as funções que você deseja permitir que executem essa ação na lista suspensa e escolha Salvar.

Comandos Git para filiais

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 filiais

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 da ramificação especificado.	<code>git branch <i>branch-name</i></code>

Essa ramificação não aparecerá no repositório remoto até que você confirme e envie a alteração.

Cria uma ramificação no repositório local usando o nome da ramificação especificado e, em seguida, alterna para ela.

```
git checkout -b branch-name
```

Alterna para outra ramificação no repositório local usando o nome de ramificação especificado.

```
git checkout other-branch-name
```

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 especificada. Também configura informações de rastreamento upstream para a filial no repositório local.

```
git push -u remote-name branch-name
```

Mescla alterações de outra ramificação no repositório local à ramificação atual no repositório local.

```
git merge from-other-branch-name
```

Exclui uma ramificação no repositório local, a menos que ela contenha trabalho que ainda não foi mesclado.

```
git branch -d branch-name
```

Exclui uma ramificação no repositório remoto usando o apelido especificado que o repositório local tem para o repositório remoto e o nome da ramificação especificada. (Observe o uso de dois pontos (:).) Como alternativa, especifique `--delete` como parte do comando.

```
git push remote-name :branch-name  
git push remote-name --delete branch-name
```

Para obter mais informações, consulte a documentação do Git.

Visualizando filiais 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 filiais remotas e locais.

Para visualizar filiais (console)

1. No CodeCatalyst console, navegue até o projeto que contém o repositório de origem no qual você deseja visualizar as ramificações. Escolha Código, escolha Repositórios de origem e, em seguida, escolha o repositório de origem.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que 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, escolha-a na lista suspensa de ramificações do repositório.
4. Para visualizar todas as ramificações de um repositório, escolha Exibir tudo. A página Branches exibe informações sobre o nome, a confirmação mais recente e as regras de cada ramificação.

Para obter informações sobre como usar o Git e seu sistema operacional para visualizar ramificações e detalhes, consulte [Comandos comuns do Git para ramificações](#), a documentação do Git e a documentação do sistema operacional.

Excluindo 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 filiais baixo pode ajudar os usuários a encontrar 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.

Para excluir uma ramificação (console)

1. Navegue até o projeto em que seu repositório reside.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja excluir uma ramificação.

3. Na página de visão geral do repositório, escolha o seletor suspenso ao lado do nome da ramificação e, em seguida, escolha Exibir tudo.
4. Escolha a ramificação que você deseja excluir e, em seguida, escolha Excluir ramificação.

Note

Você não pode excluir a ramificação padrão de um repositório.

5. Uma caixa de diálogo de confirmação é exibida. Ele 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 delete na caixa de texto e escolha Excluir.

Você também pode usar o Git para excluir ramificações. Para ter mais informações, consulte [Comandos comuns do Git para filiais](#).

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 de um arquivo em diferentes ramificações do repositório.

Para adicionar ou editar vários arquivos em um repositório de origem, você pode usar 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

- [Criando ou adicionando um arquivo](#)
- [Visualizando um arquivo](#)

- [Visualizando o histórico de alterações em um arquivo](#)
- [Editar um arquivo](#)
- [Renomeando ou excluindo um arquivo](#)

Criando ou adicionando 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, considere [criar um ambiente de desenvolvimento](#).

Para criar um ambiente de desenvolvimento a partir de um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
3. Escolha o repositório de origem em que você deseja trabalhar no código.
4. Escolha Criar ambiente de desenvolvimento.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para mais informações.
6. Execute um destes procedimentos:
 - Escolha Trabalhar na ramificação existente e, em seguida, escolha uma ramificação no menu suspenso Ramificação existente.
 - Escolha Trabalhar em 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 suspenso Criar ramificação.
7. Opcionalmente, adicione um nome para o Dev Environment ou edite sua configuração.
8. Selecione Create (Criar).

Para criar um arquivo no CodeCatalyst console

1. Navegue até o projeto em que você deseja criar um arquivo. Para obter mais informações sobre como navegar até um repositório, consulte [Visualizando um repositório de origem](#).

2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja criar o arquivo.

3. (Opcional) Escolha a ramificação em que você deseja criar o arquivo, se quiser criar o arquivo em uma ramificação diferente da ramificação padrão.
4. Escolha 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, escolha Confirmar.

6. Em Nome do arquivo, revise o nome do arquivo e faça as alterações que desejar nele. Opcionalmente, escolha a ramificação em que você deseja criar o arquivo na lista de ramificações disponíveis na 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 do commit que adiciona o arquivo ao repositório de origem.
7. Escolha 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 obter mais informações sobre os submódulos do Git, consulte a documentação do Git.](#)

Para adicionar um arquivo usando um cliente Git ou um ambiente de desenvolvimento integrado (IDE) conectado

1. Clone seu repositório de origem em seu computador local. Para ter mais informações, consulte [Clonando um repositório de origem](#).
2. Crie arquivos em seu repositório local ou copie arquivos em seu repositório local.
3. Crie e envie um commit fazendo o seguinte:
 - Se você estiver usando um cliente Git, no terminal ou na linha de comando, execute o `git add` comando, especificando os nomes dos arquivos que você deseja adicionar. Como alternativa, para adicionar todos os arquivos adicionados ou alterados, execute o `git add` comando seguido por um ponto único ou duplo para indicar se você deseja incluir todas as alterações no nível do diretório atual (período único) ou todas as alterações no diretório atual e em todos os subdiretórios (período duplo). Para confirmar as alterações, execute o `git commit -m` comando e forneça uma mensagem de confirmação. Para enviar suas alterações para o repositório de origem CodeCatalyst, execute `git push`. Para obter mais informações sobre os comandos do Git, consulte sua documentação do Git e [Comandos Git para filiais](#)
 - Se você estiver usando um ambiente de desenvolvimento ou um IDE, crie arquivos e adicione arquivos no IDE e, em seguida, confirme e envie suas alterações. Para obter mais informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#) ou consulte a documentação do IDE.

Visualizando 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 em que você deseja visualizar os arquivos. Para ter mais informações, consulte [Visualizando um repositório de origem](#).
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que 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, escolha-a na lista de ramificações.
 - Para expandir uma pasta, escolha-a na lista.
5. Para ver o conteúdo de um arquivo específico, escolha-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 escolher arquivos adicionais para visualizar em Exibir arquivos. Para editar um arquivo, escolha 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 obter mais informações, consulte a documentação do seu cliente Git ou IDE.

Note

Você não pode visualizar os submódulos do Git no console. CodeCatalyst [Para obter mais informações sobre os submódulos do Git, consulte a documentação do Git.](#)

Visualizando 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 vários commits na ramificação em que você escolhe visualizar o histórico do arquivo. Por exemplo, se você visualizar o histórico de alterações no **readme.md** arquivo na **main** ramificação 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 em que você deseja visualizar o histórico de um arquivo. Para ter mais informações, consulte [Visualizando um repositório de origem](#).
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.
3. Escolha o repositório em que você deseja visualizar o histórico de um arquivo. Escolha a ramificação em que você deseja visualizar o histórico do arquivo e, em seguida, escolha o arquivo na lista. Escolha View history (Exibir histórico).
4. Examine a lista de confirmações que incluíram alterações nesse arquivo na ramificação especificada. Para ver 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 exibições dessa ramificação, escolha o arquivo na lista de arquivos e escolha Exibir histórico.

Note

Você não pode ver o histórico de alterações nos submódulos do Git no console. CodeCatalyst [Para obter mais informações sobre os submódulos do Git, consulte a documentação do Git.](#)

Editar 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 obter

mais informações, consulte [Escreva e modifique o código com Dev Environments em CodeCatalyst](#) ou [Clonando um repositório de origem](#).

Para editar um arquivo no CodeCatalyst console

1. Navegue até o projeto em que você deseja editar um arquivo. Para obter mais informações sobre como navegar até um repositório, consulte [Visualizando um repositório de origem](#).
2. Escolha o repositório em que você deseja editar o arquivo. Escolha Exibir ramificações e, em seguida, escolha a ramificação na qual você deseja trabalhar. Escolha o arquivo na lista de arquivos e pastas dessa ramificação.

O conteúdo do arquivo é exibido.

3. Selecione a opção Editar.
4. No editor, edite o conteúdo do arquivo e escolha Confirmar. Opcionalmente, em Confirmar alterações, adicione mais informações sobre a alteração na mensagem Confirmar. Quando estiver satisfeito com suas alterações, escolha Confirmar.

Renomeando ou excluindo 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 pull request é a principal forma pela qual você e outros membros do projeto podem revisar, comentar e mesclar alterações de código de uma ramificação para outra. Você pode usar pull requests para revisar as alterações no código de forma colaborativa em busca de pequenas alterações ou correções, adições de recursos importantes ou novas versões do seu software lançado. Se você usa problemas para monitorar o trabalho em seu projeto, você pode vincular problemas específicos a pull requests para ajudá-lo a monitorar quais problemas estão sendo resolvidos pelas alterações de código na pull request. Quando você cria, atualiza, comenta, mescla ou fecha uma pull request, um e-mail é enviado automaticamente para o autor da pull request, bem como para todos os revisores obrigatórios ou opcionais da pull request.

i Tip

Você pode configurar sobre quais eventos de pull request você receberá e-mails como parte do seu perfil. Para ter mais informações, consulte [Gerenciando notificações na Amazon CodeCatalyst](#).

As pull requests 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, na qual 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).

i Note

Enquanto você cria uma pull request, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois de criar a pull request, a diferença exibida será entre a revisão da pull request escolhida e o commit que foi a ponta da ramificação de destino quando você criou a pull request. Para obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

Enquanto uma pull request é criada para um repositório e ramificações de origem específicos, você pode criá-los, visualizá-los, revisá-los e fechá-los como parte do trabalho com seu projeto. Você não precisa visualizar o repositório de origem para visualizar e trabalhar com pull requests. Um estado de pull request é definido como Aberto quando você o cria. 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.

Depois que seu código for revisado, você poderá alterar o estado do pull request de várias maneiras:

- Mescle o pull request no CodeCatalyst console. O código na ramificação de origem da pull request será mesclado na ramificação de destino. O status do pull request mudará para Mesclado. Não pode ser alterado novamente 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.

Ates 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 qualquer fluxo de trabalho executado quando você cria uma pull request. (Essa etapa é opcional, mas recomendada.)

Tópicos

- [Criação de uma solicitação pull](#)
- [Visualizando pull requests](#)
- [Gerenciando requisitos para mesclar uma pull request com regras de aprovação](#)
- [Analisando uma pull request](#)
- [Atualizando uma pull request](#)
- [Mesclando uma pull request](#)
- [Fechando uma pull request](#)

Criação de 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 pull request 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 pull request a partir de uma ramificação específica fornece automaticamente o nome do repositório e a ramificação de origem na página de criação da pull request. Ao criar uma pull request, você receberá automaticamente e-mails sobre quaisquer atualizações na pull request, bem como quando a pull request for mesclada ou fechada.

Note

Enquanto você cria uma pull request, a diferença exibida é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a pull request for criada, a diferença exibida será entre a revisão da pull request escolhida e o commit que foi a ponta da ramificação de destino quando você criou a pull request. 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 Write description for me ao criar pull requests para que o Amazon Q crie automaticamente uma descrição das alterações contidas em um pull request. Quando você escolhe essa opção, o Amazon Q analisa as diferenças entre a ramificação de origem que contém as alterações de código e a filial de destino na qual você deseja mesclar essas alterações. Em seguida, ele cria um resumo do que são essas mudanças, bem como sua melhor interpretação da intenção e do efeito dessas mudanças. Esse recurso está disponível somente na região Oeste dos EUA (Oregon).

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Write description for me e Create content summary foram criados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança e o uso responsável da inteligência artificial (IA).

Para criar uma pull request

1. Navegue até o projeto.
2. Execute um destes procedimentos:
 - No painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha Create pull request.
 - Na página inicial do repositório, escolha Mais e, em seguida, escolha Criar pull request.
 - Na página do projeto, escolha Criar pull request.
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 pull request na página principal do repositório.


4. Em Agência de destino, escolha a ramificação na qual mesclar o código depois que ele for revisado.
5. Na ramificação de origem, escolha a ramificação que contém o código confirmado.
6. Em Título do pull request, insira um título que ajude outros usuários a entender o que precisa ser revisado e por quê.
7. (Opcional) Na descrição da pull request, forneça informações como um link para problemas ou uma descrição de suas 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 ao pull request.

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).

8. (Opcional) Em Problemas, escolha Vincular problemas e, em seguida, escolha um problema na lista ou insira sua ID. Para desvincular um problema, escolha o ícone de desvinculação.
9. (Opcional) Em Revisores obrigatórios, escolha Adicionar revisores obrigatórios. Escolha na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que o pull request possa ser mesclado na ramificação de destino.


 Note

Você não pode adicionar um revisor como revisor obrigatório e como revisor opcional. Você não pode se adicionar como revisor.

10. (Opcional) Em Revisores opcionais, escolha Adicionar revisores opcionais. Escolha na lista de membros do projeto para adicioná-los. Os revisores opcionais não precisam aprovar as alterações como um requisito para que a pull request possa ser incorporada à ramificação de destino.
11. Analise as diferenças entre as filiais. A diferença exibida em uma pull request 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 pull request 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 você estiver convencido de que a pull request contém o código e as alterações que você deseja revisar, escolha Criar.

 Note

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

Para criar uma pull request a partir de uma ramificação

1. Navegue até o projeto em que você deseja criar uma pull request.
2. No painel de navegação, escolha Repositórios de origem e, em seguida, escolha o repositório que contém a ramificação em que você tem alterações de código para revisar.
3. Escolha a seta suspensa ao lado do nome da ramificação padrão e, em seguida, escolha a ramificação desejada na lista. Para visualizar todas as ramificações de um repositório, escolha Exibir tudo.
4. Escolha Mais e, em seguida, escolha Criar pull request.
5. O repositório e a ramificação de origem estão pré-selecionados para você. Na filial 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 pull request serão executados após a criação da pull request, se a ramificação de destino da pull request corresponder a uma das ramificações especificadas no fluxo de trabalho.

6. Analise as diferenças entre as filiais. 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, escolha Vincular problemas e, em seguida, escolha um problema na lista ou insira sua ID. Para desvincular um problema, escolha o ícone de desvinculação.
8. (Opcional) Em Revisores obrigatórios, escolha Adicionar revisores obrigatórios. Escolha na lista de membros do projeto para adicioná-los. Os revisores obrigatórios devem aprovar as alterações antes que o pull request possa ser mesclado na ramificação de destino.

Note

Você não pode adicionar um revisor tanto como obrigatório quanto opcional. Você não pode se adicionar como revisor.

9. (Opcional) Em Revisores opcionais, escolha Adicionar revisores opcionais. Escolha na lista de membros do projeto para adicioná-los. Os revisores opcionais não precisam aprovar as alterações antes que a pull request possa ser mesclada na ramificação de destino.
10. Quando você estiver convencido de que a pull request contém as alterações que você deseja revisar e inclui os revisores necessários, escolha Criar.

Se você tiver algum fluxo de trabalho configurado para execução em que a ramificação corresponda à ramificação de destino na pull request, você verá informações sobre essas execuções de fluxo de trabalho em Visão geral na área de detalhes da pull request após a criação da pull request. Para ter mais informações, consulte [Adicionar um gatilho push, pull ou scheduler](#).

Visualizando pull requests

Você pode visualizar pull requests para um projeto no CodeCatalyst console da Amazon. A página de resumo do projeto exibe todas as pull requests abertas para um projeto. Para ver todas as pull requests, independentemente do estado, navegue até a página de pull requests do seu projeto. Ao visualizar uma pull request, você pode optar por ter um resumo de todos os comentários deixados sobre as alterações na pull request criada para você.

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Write description for me e Create content summary foram criados no

Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança e o uso responsável da inteligência artificial (IA).

Para visualizar pull requests abertos

1. Navegue até o projeto em que você deseja visualizar pull requests.
2. Na página do projeto, as pull requests abertas são exibidas, incluindo informações sobre quem criou a pull request, qual repositório contém as ramificações da pull request e a data em que a pull request foi criada. Você pode filtrar a visualização aberta do pull request por repositório de origem.
3. Para ver todas as pull requests, escolha Exibir tudo. Você pode usar os seletores para escolher entre as opções. Por exemplo, para ver todas as pull requests, escolha Qualquer status e Qualquer autor.

Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Pull requests e use os seletores para refinar sua visualização.

4. Na página de pull requests, você pode classificar pull requests por ID, título, status e muito mais. Para personalizar quais informações e quantas informações são exibidas na página de pull requests, escolha o ícone de roda dentada.
5. Para ver um pull request específico, escolha-o na lista.
6. Para visualizar o status das execuções de fluxos de trabalho associadas a essa pull request, se houver, escolha Visão geral e revise as informações na área de detalhes da pull request 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 pull request e se os requisitos da ramificação de destino no fluxo de trabalho corresponderem à ramificação de destino especificada na pull request. Para ter mais informações, consulte [Adicionar um gatilho push, pull ou scheduler](#).

7. Para visualizar problemas vinculados, se houver, escolha Visão geral e revise as informações nos detalhes da pull request em Problemas. Se você quiser ver um problema vinculado, escolha sua ID na lista.
8. (Opcional) Para criar um resumo dos comentários deixados nas alterações de código nas revisões da pull request, escolha Criar resumo do conteúdo. O resumo não incluirá nenhum comentário deixado no pull request geral.

Note

Essa funcionalidade exige que os recursos generativos de IA estejam habilitados para o espaço e só esteja disponível na região Oeste dos EUA (Oregon). Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).

9. Para ver as alterações de código na pull request, escolha Alterações. Você pode ver rapidamente quantos arquivos têm alterações na pull request e quais arquivos na pull request têm comentários sobre eles, em Arquivos alterados. O número de comentários exibidos 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 pull request 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 pull request. Algumas diferenças podem ser muito grandes para serem exibidas. Para obter mais informações, consulte [Cotas para repositórios de origem em CodeCatalyst](#) e [Visualizando um arquivo](#).

10. Para visualizar os relatórios de qualidade desse pull request, escolha Relatórios.

Note

Um fluxo de trabalho deve ser configurado para gerar relatórios para que eles apareçam em suas pull requests. Para ter mais informações, consulte [Testando com fluxos de trabalho](#).

Gerenciando requisitos para mesclar uma pull request com regras de aprovação

Ao criar uma pull request, você pode optar por adicionar revisores obrigatórios ou opcionais a essa pull request individual. No entanto, você também pode criar requisitos que todas as pull requests 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 filiais em um repositório. Quando você cria uma pull request 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 pull request com essa ramificação. A criação de regras de aprovação pode ajudá-lo a manter os padrões de qualidade para fusões em filiais, como sua filial 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, você deve considerar 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 possa ser mesclada na ramificação de destino, você efetivamente criou uma regra que exige que todos, exceto a pessoa que criou as pull requests, aprovem essa pull request 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.

Para visualizar e editar regras de aprovação para filiais de destino para pull requests

1. Navegue até o projeto em que seu repositório reside.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja visualizar as regras de aprovação.

3. Na página de visão geral do repositório, escolha Branches.

4. Na coluna Regras de aprovação, escolha Exibir para ver o status de todas as regras para cada ramificação do repositório.

Em Número mínimo de aprovações, o número corresponde ao número de aprovações necessárias antes que uma pull request possa ser mesclada a essa filial.

5. Para criar ou alterar uma regra de aprovação, escolha Gerenciar configurações. Na página de configurações do repositório de origem, em Regras de aprovação, escolha Editar.

Note

Você deve ter a função de administrador do projeto para editar as regras de aprovação.

6. Em Filial, escolha o nome da filial para a qual você deseja configurar uma regra de aprovação na lista suspensa. Em Número mínimo de aprovações, insira um número e escolha Salvar.

Analizando uma pull request

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 pull request. Você pode optar por criar um resumo dos comentários deixados sobre as alterações de código na pull request para ajudá-lo a entender rapidamente o feedback deixado por outros usuários. Você também pode optar por criar um ambiente de desenvolvimento para trabalhar no código.

Note

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como os recursos Write description for me e Create content summary foram criados no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança e o uso responsável da inteligência artificial (IA).

i Tip

Você pode configurar sobre quais eventos de pull request você receberá e-mails como parte do seu perfil. Para ter mais informações, consulte [Gerenciando notificações na Amazon CodeCatalyst](#).

As pull requests mostram qual será a diferença entre a revisão da pull request e o commit que foi a ponta da ramificação de destino quando você criou a pull request. 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.

i Tip

Ao trabalhar no console, especialmente se você tem uma pull request aberta há algum tempo, considere atualizar seu navegador para garantir que você tenha a revisão mais recente disponível para uma pull request antes de começar a revisá-la.


Para revisar uma pull request no CodeCatalyst console

1. Navegue até o projeto.
2. Navegue até as pull requests fazendo o seguinte:
 - Se a pull request estiver listada na página do projeto, escolha-a na lista.
 - Se a pull request não estiver listada na página do projeto, escolha Exibir tudo. Use os filtros e classifique para encontrar a pull request e, em seguida, escolha-a na lista.
 - No painel de navegação, escolha Código e, em seguida, escolha Pull requests.
3. Escolha a pull request que você deseja revisar na lista. Você pode filtrar a lista de pull requests digitando parte do nome na barra de filtro.
4. Em Visão geral, você pode revisar o nome e o título da pull request. Você pode criar e visualizar os comentários deixados na própria pull request. Você também pode visualizar os detalhes da pull request, incluindo informações sobre execuções de fluxo de trabalho, problemas vinculados, revisores, o autor da pull request 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 pull request, expanda Comentários na pull request e escolha Criar comentário.
6. (Opcional) Para ver um resumo de todos os comentários deixados sobre as alterações nas revisões desse pull request, escolha Criar resumo de comentários.

 Note

Essa funcionalidade exige que os recursos generativos de IA estejam habilitados para o espaço e só esteja disponível na região Oeste dos EUA (Oregon). Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).


7. Em Alterações, você pode ver as diferenças entre a ramificação de destino e a revisão mais recente da pull request. Se houver mais de uma revisão, você poderá alterar quais revisões são comparadas na diferença entre elas. Para obter mais informações sobre revisões, consulte [Revisões](#).

 Tip

Você pode ver rapidamente quantos arquivos têm alterações na pull request e quais arquivos na pull request têm comentários sobre eles, em Arquivos alterados. O número de comentários mostrados ao lado de uma pasta indica o número de comentários nos arquivos dessa pasta. Expand 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 pull request, vá até a linha na qual você deseja comentar. Escolha o ícone de comentário que aparece para essa linha, insira um comentário e escolha Salvar.
10. Para visualizar as alterações entre as revisões em uma pull request 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 acionadores de pull request, você pode ver as descobertas de cobertura de linha e filial na pull request relevante. Para ocultar as descobertas de cobertura de código, escolha Ocultar cobertura de código. Para ter mais informações, consulte [Relatórios de cobertura de código](#).
12. Se quiser fazer alterações no código da pull request, você pode criar um ambiente de desenvolvimento a partir da pull request. Escolha Criar ambiente de desenvolvimento. Opcionalmente, adicione um nome para o Dev Environment ou edite sua configuração e escolha Create.
13. Em Relatórios, você pode visualizar os relatórios de qualidade nessa pull request. 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 em suas pull requests. Para ter mais informações, consulte [Configurando relatórios de qualidade em uma ação](#).

14. Para ver um relatório específico, escolha-o na lista. Para ter mais informações, consulte [Testando com fluxos de trabalho](#).
15. Se você estiver listado como revisor dessa pull request e quiser aprovar as alterações, verifique se está visualizando a revisão mais recente e escolha Aprovar.

 Note

Todos os revisores obrigatórios devem aprovar uma pull request antes que ela possa ser mesclada.

Atualizando uma pull request

Você pode facilitar a revisão do código por outros membros do projeto atualizando a pull request. Você pode atualizar uma pull request para alterar seus revisores, seus links para problemas, o título da pull request ou sua descrição. Por exemplo, talvez você queira alterar os revisores necessários para um pull request para remover alguém que está de férias e adicionar outra pessoa. Você também pode atualizar uma pull request com mais alterações no código enviando commits para a ramificação de origem de uma pull request 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 pull request.

Para atualizar os revisores para uma pull request

1. Navegue até o projeto em que você deseja atualizar os revisores de uma pull request.
2. Na página do projeto, em Abrir pull requests, escolha a pull request em que você deseja atualizar os revisores. Como alternativa, no painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha a pull request que você deseja atualizar.
3. (Opcional) Em Visão geral, na área de detalhes do pull request, 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 de detalhes da pull request, escolha Vincular problemas para vincular um problema à pull request e, em seguida, escolha um problema na lista ou insira sua 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 pull request

1. Para atualizar vários arquivos, [crie um ambiente de desenvolvimento](#) ou clone o repositório e sua 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 [Clonando 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 ter mais informações, consulte [Editar um arquivo](#).

Para atualizar o título e a descrição de uma pull request

1. Navegue até o projeto em que você deseja atualizar o título ou a descrição de uma pull request.
2. A página do projeto exibe pull requests abertas, incluindo informações sobre quem criou a pull request, qual repositório contém as ramificações da pull request e quando a pull request foi

- criada. Você pode filtrar a visualização aberta do pull request por repositório de origem. Escolha a pull request que você deseja alterar na lista.
3. Para ver todas as pull requests, escolha Exibir tudo. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Pull requests. Use a caixa de filtro ou as funções de classificação para encontrar a pull request que você deseja alterar e, em seguida, escolha-a.
 4. Em Visão geral, escolha Editar.
 5. Altere o título ou a descrição e escolha Salvar.

Mesclando uma pull request

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 pull request com sua escolha de estratégias de mesclagem do Git executando `git merge` o comando em seu 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 pull request.

Se você tiver a função de administrador do projeto, também poderá optar por mesclar uma pull request que ainda não atendeu a todos os requisitos de aprovações e regras de aprovação.

Mesclando uma pull request (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 obter a aprovação

de todos os aprovadores necessários, resolver conflitos localmente, se necessário, e enviar essas alterações antes de poder mesclar. A mesclagem de uma pull request enviará automaticamente um e-mail para o criador da pull request, bem como para quaisquer revisores obrigatórios ou opcionais. Ele não fechará nem alterará automaticamente o status de nenhum problema vinculado à pull request.

Tip

Você pode configurar sobre quais eventos de pull request você receberá e-mails como parte do seu perfil. Para ter mais informações, consulte [Gerenciando notificações na Amazon CodeCatalyst](#).

Para mesclar uma pull request

1. Navegue até o projeto em que você deseja mesclar uma pull request.
2. Na página do projeto, em Abrir pull requests, escolha a pull request que você deseja mesclar. Se você não vê a pull request, escolha Exibir todas as pull requests e, em seguida, escolha-a na lista. Como alternativa, no painel de navegação, escolha Código, escolha Pull requests e, em seguida, escolha a pull request que você deseja mesclar. Escolha Merge (Mesclar).
3. Escolha entre as estratégias de mesclagem disponíveis para o pull request. Opcionalmente, selecione ou desmarque a opção de excluir a ramificação de origem após mesclar a pull request e, em seguida, escolha 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 pull request é indicado por um ícone de relógio na área de detalhes da pull request em Visão geral. Se todos os revisores obrigatórios aprovaram o pull request, 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 pull request 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 obter mais informações, consulte [Mesclando uma pull request \(Git\)](#) e sua documentação do Git.

Substituir requisitos de mesclagem

Se você tiver a função de administrador do projeto, poderá optar por mesclar uma pull request que ainda não atendeu a todos os requisitos das aprovações e regras de aprovação necessárias. Isso é chamado de substituir os requisitos de uma pull request. Você pode optar por fazer isso se um revisor obrigatório não estiver disponível ou se surgir uma necessidade urgente de mesclar um pull request específico em uma ramificação que tenha regras de aprovação que não possam ser atendidas rapidamente.

Para mesclar uma pull request

1. No pull request em que você deseja substituir os requisitos e mesclar, escolha a seta suspensa ao lado do botão Mesclar. Escolha Substituir requisitos de aprovação.
2. Em Substituir motivo, forneça detalhes do motivo pelo qual você está mesclando essa pull request sem que ela atenda às regras de aprovação e aos requisitos obrigatórios do revisor. Embora isso seja opcional, isso é altamente recomendado.
3. Opcionalmente, 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 mantenha a ramificação de origem ao ignorar os requisitos para mesclar uma pull request até ter a chance de revisar a decisão com outros membros da equipe.
5. Escolha Merge (Mesclar).

Mesclando uma pull request (Git)

O Git suporta muitas opções para mesclar e gerenciar ramificações. Os comandos a seguir são algumas das opções que você pode usar. Para obter mais informações, consulte a documentação disponível no site do [Git](#). Depois de mesclar e enviar suas alterações, feche manualmente a pull request. Para ter mais informações, consulte [Fechando uma pull request](#).

Comandos comuns do Git para mesclar ramificações

Mescla as alterações da ramificação de origem no repositório local com a 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 rápida. 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 de 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 de três vias. 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 seu 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 alternati

```
git push remote-name :source-branch-name  
  
git push remote-name --delete source-branch-name
```

va, especifique `--delete` como parte do comando.

Fechando uma pull request

Você pode marcar um pull request como Fechado. Isso não mescla a pull request, mas pode ajudá-lo a determinar quais pull requests exigem ação e quais pull requests não são mais relevantes. Recomendamos fechar uma pull request se você não planeja mais mesclar essas alterações ou se as alterações foram mescladas por outra pull request.

Fechar uma pull request enviará automaticamente um e-mail para o criador da pull request, bem como para quaisquer revisores obrigatórios ou opcionais. Isso não alterará automaticamente o status de nenhum problema vinculado ao pull request.

Note

Você não pode reabrir uma pull request após ela ter sido fechada.

Para fechar uma pull request

1. Navegue até o projeto em que você deseja fechar uma pull request.
2. Na página do projeto, pull requests abertos são exibidos. Escolha o pull request que você deseja fechar.
3. Escolha Fechar.
4. Revise as informações e escolha Fechar pull request.

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. Toda vez que um usuário confirma e envia uma alteração para uma ramificação, essas informações são salvas. As informações do Git commit incluem o autor do commit, a pessoa que cometeu 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 aos commits para ajudar a identificar commits específicos.

Na Amazon CodeCatalyst, você pode:

- Veja uma lista de confirmações de uma ramificação.
- Visualize confirmações individuais, incluindo as alterações feitas em uma confirmação em comparação com seu pai ou pais.

Você também pode visualizar arquivos e pastas. Para ter mais informações, consulte [Gerenciando arquivos de código-fonte na Amazon CodeCatalyst](#).

Tópicos

- [Visualizando confirmações em uma ramificação](#)
- [Alterando a forma como os commits são exibidos \(CodeCatalystconsole\)](#)

Visualizando 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 um commit específico.

Tip

Você também pode ver o histórico de confirmações que fizeram alterações em um arquivo específico. Para obter mais informações, consulte [Visualizando um arquivo](#).

Você também pode ver os commits usando seu cliente Git. Para obter mais informações, consulte a documentação do Git.

Para ver os commits (console)

1. Navegue até o projeto que contém o repositório de origem no qual você deseja visualizar os commits.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja visualizar as confirmações em uma ramificação.

3. A ramificação padrão do repositório é exibida, incluindo informações sobre a confirmação mais recente da ramificação. Escolha Compromissos. Como alternativa, escolha Mais e, em seguida, escolha Exibir confirmações.
4. Para ver as confirmações de uma ramificação diferente, escolha o seletor de ramificação e, em seguida, escolha o nome da ramificação.
5. Para ver detalhes sobre um determinado commit, escolha seu título em Título do commit. 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 um pai, você poderá escolher em qual confirmação principal visualizar as informações e exibir as alterações escolhendo o ícone suspenso ao lado da ID da confirmação principal.

Alterando a forma como os commits são exibidos (CodeCatalystconsole)

Você pode alterar quais informações são exibidas na visualização de Confirmações. Você pode optar por ocultar ou exibir colunas como autor e ID do commit.

Para alterar a forma como os commits são exibidos (console)

1. Navegue até o projeto que contém o repositório de origem no qual você deseja visualizar os commits.
2. Escolha o nome do repositório na lista de repositórios de origem do projeto. Como alternativa, no painel de navegação, escolha Código e, em seguida, escolha Repositórios de origem.

Escolha o repositório em que você deseja alterar a forma como você visualiza os commits.

3. A ramificação padrão do repositório é exibida, incluindo informações sobre a confirmação mais recente da ramificação. Escolha Compromissos.
4. Escolha o ícone de engrenagem.
5. Em Preferências, escolha o número de confirmações a serem exibidas e escolha se deseja exibir informações sobre o autor da confirmação, a data da confirmação e a ID da confirmação.

Note

Você não pode ocultar o título do commit na exibição das informações.

- Depois de fazer suas alterações, escolha 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 CodeCatalyst](#).

Recurso	Informações
Nomes de ramificação	<p>Qualquer combinação de caracteres permitidos entre 1 e 256 caracteres deve ser exclusiva em um repositório. Nomes de ramificação não podem:</p> <ul style="list-style-type: none"> começar nem terminar com uma barra (/) ou um ponto (.) consistir em único caractere @ conter dois ou mais pontos consecutivos (.), barras (//) ou a seguinte combinação de caracteres: @{ conter espaços ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ : <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 Git Internals e. git-check-ref-format</p>
Comentários sobre um pull request	Máximo de 1.000 em uma pull request.

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">• conter sequências de caracteres vazias• ser um caminho de arquivo relativo• incluir qualquer uma das seguintes combinações: <code>./</code> <code>../</code> <code>//</code>• terminar com barra no final ou barra invertida <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 para 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 data-bbox="829 302 1507 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</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, considere fazer várias confirmações menores em vez de uma confirmação grande.</p> </div>
Metadados de uma confirmação	<p>Máximo de 6 MB para os metadados combinados de uma confirmação (por exemplo, a combinação de informações do autor, data, lista de confirmações principal e mensagens de confirmação).</p> <div data-bbox="829 1163 1507 1570" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</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 pull request	50

Recurso	Informações
Número de pull requests abertos em um espaço	Máximo de 1.000 para um CodeCatalyst espaço da Amazon.
Número total de pull requests 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 conter qualquer combinação de letras, números, pontos, sublinhados e traços entre 1 e 100 caracteres. Os nomes não diferenciam maiúsculas de minúsculas. Os nomes dos repositórios não podem terminar em <code>.git</code> , não podem conter espaços e não podem conter nenhum dos seguintes caracteres: <code>! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :</code>
Tamanho do repositório	Os tamanhos dos repositórios são afetados pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte Preços do e Solução de problemas com repositórios de origem .

Recurso	Informações
Revisores de uma pull request	Máximo de 100 revisores no total (opcional ou obrigatório) para uma pull request.
Resumos escritos para pull requests	O número máximo de resumos escritos para pull requests depende do nível de cobrança do seu espaço. Para obter mais informações, consulte Preços do .

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 nele a partir de um repositório de origem e trabalhe nesse código com um IDE compatível.
- Crie um ambiente de desenvolvimento em um IDE e clone 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.

As ferramentas do projeto e as bibliotecas de aplicativos incluídas em seu ambiente de desenvolvimento são definidas pelo devfile no repositório de origem do seu 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 obter mais informações sobre o devfile, consulte <https://devfile.io>.

Depois de criar um ambiente de desenvolvimento, somente você poderá 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 Space, poderá alterar o nível de cobrança do seu espaço para usar diferentes opções de configuração do Dev Environment e gerenciar os limites de computação e armazenamento.

Tópicos

- [Criar um ambiente de desenvolvimento](#)
- [Interromper um ambiente de desenvolvimento](#)
- [Retomar um ambiente de desenvolvimento](#)
- [Editando um ambiente de desenvolvimento](#)
- [Excluir um ambiente de desenvolvimento](#)
- [Conectando-se a um ambiente de desenvolvimento usando SSH](#)
- [Configurando um devfile para um ambiente de desenvolvimento](#)
- [Associando 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
- Crie um ambiente de desenvolvimento no IDE de sua escolha e clone qualquer repositório de origem no ambiente de desenvolvimento

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 IDEs compatíveis. Você deve ter o AWS Toolkit instalado para usar os ambientes de desenvolvimento em seu IDE. Para ter 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 ambientes de desenvolvimento com os seguintes ambientes de desenvolvimento integrado (IDEs) compatíveis:

- [AWS Cloud9](#)
- [JetBrains IDEs](#)
 - [IntelliJ IDEA Ultimate](#)
 - [GoLand](#)
 - [PyCharmProfessional](#)
- [Código do Visual Studio](#)

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](#).

Para criar um ambiente de desenvolvimento a partir de uma 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, faça o seguinte:

- Escolha Visão geral e, em seguida, navegue até a seção Meus ambientes de desenvolvimento.
 - Escolha Código e, em seguida, escolha Dev Environments.
 - Escolha Código, escolha Repositórios de origem e escolha o repositório para o qual você deseja criar um ambiente de desenvolvimento.
4. Escolha Criar ambiente de desenvolvimento.
 5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para 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, em seguida, 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 a partir da página de repositórios de origem ou de um repositório de origem específico, não precisará escolher um repositório. O ambiente de desenvolvimento será criado a partir do repositório de origem que você escolheu na página Repositórios de origem.

8. (Opcional) Em Alias - opcional, insira um alias para o Dev Environment.
9. (Opcional) Escolha o botão de edição da configuração do Dev Environment para editar a configuração de computação, armazenamento ou tempo limite do Dev Environment.
10. (Opcional) Na 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 ao VPC não são compatíveis. AWS Toolkit

 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 CIDR IPv4 não esteja configurado para o intervalo de 172.16.0.0/12 endereços 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, faça o seguinte:
 - Escolha Visão geral e, em seguida, navegue até a seção Meus ambientes de desenvolvimento.
 - Escolha Código e, em seguida, escolha Dev Environments.
4. Escolha Criar ambiente de desenvolvimento.
5. Escolha um IDE compatível no menu suspenso. Consulte [Ambientes de desenvolvimento integrados compatíveis para ambientes de desenvolvimento](#) Para mais informações.
6. Escolha Criar um ambiente de desenvolvimento vazio.
7. (Opcional) Em Alias - opcional, insira um alias para o Dev Environment.
8. (Opcional) Escolha o botão de edição da configuração do Dev Environment para editar a configuração de computação, armazenamento ou tempo limite do Dev Environment.
9. (Opcional) Na 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 ao VPC não são compatíveis. AWS Toolkit


 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 CIDR IPv4 não esteja configurado para o intervalo de 172.16.0.0/12 endereços 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

Criar e abrir um ambiente de desenvolvimento pela primeira vez pode levar de um a dois minutos.

 Note

Depois que o Dev Environment 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.

Criando um ambiente de desenvolvimento em um IDE

Você pode usar Dev Environments para trabalhar rapidamente no código armazenado nos repositórios de origem do seu projeto. Os ambientes de desenvolvimento aumentam sua velocidade de desenvolvimento porque você pode começar a codificar 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 `/projects` diretório de um ambiente de desenvolvimento armazena os arquivos que são retirados do repositório de origem e o devfile usado para configurar o ambiente de desenvolvimento. O `/home` diretório, que fica vazio na criação do Dev Environment, armazena os arquivos que você cria enquanto usa seu Dev Environment. Tudo nos `/home` diretórios `/projects` e 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, certifique-se de fechar todas as instâncias conectadas para evitar custos adicionais.

Um ambiente de desenvolvimento será interrompido automaticamente se ficar inativo pelo tempo selecionado nos campos Tempo limite durante a criação do ambiente de desenvolvimento. Você pode interromper o Dev Environment 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.

Para 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 você deseja interromper um ambiente de desenvolvimento.
3. No painel de navegação, escolha Código.
4. Escolha Dev Environments.
5. Escolha o botão de rádio para o ambiente de desenvolvimento que você deseja interromper.
6. No menu Ações, escolha 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 computacional.

Retomar um ambiente de desenvolvimento

O `/projects` diretório de um ambiente de desenvolvimento armazena os arquivos que são retirados do repositório de origem e o `devfile` usado para configurar o ambiente de desenvolvimento. O `/home` diretório, que fica vazio na criação do Dev Environment, armazena os arquivos que você cria enquanto usa seu Dev Environment. Tudo nos `/home` diretórios `/projects` e 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 e continuar trabalhando em seu ambiente de desenvolvimento mais tarde.

Um ambiente de desenvolvimento será interrompido automaticamente se ficar inativo pelo tempo selecionado nos campos Tempo limite durante a criação do ambiente de desenvolvimento. Você deve fechar a guia do AWS Cloud9 navegador para que o Dev Environment fique ocioso.

Note

O Dev Environment ainda está disponível e em execução mesmo se você excluir a ramificação com a qual criou o Dev Environment. 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 a partir da página de visão geral


1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que você deseja retomar um ambiente de desenvolvimento e navegue até a seção Meus ambientes de desenvolvimento.
3. Escolha Retomar em (IDE).
 - Para JetBrains IDEs, escolha JetBrains Gateway-EAP quando solicitado a Escolher um aplicativo para abrir o link -gateway. JetBrains Escolha Abrir link para confirmar quando solicitado.
 - Para o IDE do VS Code, escolha VS Code quando solicitado a escolher um aplicativo para abrir o link do VS Code. Escolha Abrir link para confirmar.

Para retomar um ambiente de desenvolvimento a partir do repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que você deseja retomar um ambiente de desenvolvimento.
3. No painel de navegação, escolha Código.
4. Escolha Repositórios de origem.
5. Escolha o repositório de origem que contém o ambiente de desenvolvimento que você deseja retomar.
6. Escolha o nome da filial para ver um menu suspenso de suas filiais e, em seguida, escolha sua filial.
7. Escolha Resume Dev Environment.
 - 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 opção 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 você deseja retomar um ambiente de desenvolvimento.
3. No painel de navegação, escolha Código.
4. Escolha Dev Environments.
5. Na coluna IDE, escolha Resume in (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 opção 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.

Editando um ambiente de desenvolvimento

Enquanto seu IDE está em execução, você pode editar o Dev Environment. Se você editar o tempo limite de computação ou inatividade, seu ambiente de desenvolvimento será reiniciado depois que você salvar suas alterações.

Para editar um ambiente de desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que você deseja editar um ambiente de desenvolvimento.
3. No painel de navegação, escolha Código.
4. Escolha Dev Environments.
5. Escolha o ambiente de desenvolvimento que você deseja editar.
6. Selecione a opção 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 excluir o ambiente de desenvolvimento. Crie um novo ambiente de desenvolvimento para trabalhar em novos conteúdos. Se você excluir seu ambiente de desenvolvimento, o conteúdo persistente será excluído permanentemente. Antes de excluir seu ambiente de desenvolvimento, certifique-se de confirmar e enviar suas alterações de código para o repositório de origem original do ambiente de desenvolvimento. Depois de excluir seu ambiente de desenvolvimento, a cobrança de computação e armazenamento do ambiente de desenvolvimento será interrompida.

Depois de excluir seu ambiente de desenvolvimento, pode levar alguns minutos para que a cota de armazenamento seja atualizada. Se você atingiu a cota de armazenamento, não poderá criar um novo 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, você não poderá mais recuperá-lo.

Para excluir um ambiente de desenvolvimento

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o projeto em que você deseja excluir um ambiente de desenvolvimento.
3. No painel de navegação, escolha Código.
4. Escolha Dev Environments.
5. Escolha o ambiente de desenvolvimento que você deseja excluir.
6. Escolha Excluir.
7. Digite **delete** para confirmar a exclusão do Dev Environment.
8. Escolha Excluir.

Note

Antes de excluir uma conexão de VPC em seu espaço, certifique-se de remover 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. Certifique-se de limpar 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.

Conectando-se a 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, certifique-se de definir 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 rpm pacotes 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 você deseja se conectar a um ambiente de desenvolvimento usando SSH.
3. No painel de navegação, escolha Código.
4. Escolha Dev Environments.
5. Escolha um ambiente de desenvolvimento em execução ao qual você deseja se conectar usando SSH.
6. Escolha Connect via SSH, escolha o sistema operacional desejado e faça o seguinte:

- Se você ainda não tiver 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 ter 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 `~/ .aws/codecatalyst-dev-env` diretório 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. Esse 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>
```

- Fazendo o 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>
```

Configurando 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 para o 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 blueprint vazio, poderá criar um devfile manualmente. Se você criar um projeto usando um blueprint diferente, CodeCatalyst cria um devfile automaticamente. O `/projects` diretório de um ambiente de desenvolvimento armazena os arquivos que são extraídos do repositório de origem e do devfile. O `/home` diretório, que está vazio quando você cria um ambiente de desenvolvimento pela primeira vez, armazena os arquivos que você cria enquanto usa seu ambiente de desenvolvimento. Tudo nos `/home` diretórios `/projects` e de um ambiente de desenvolvimento é armazenado de forma persistente.

Note

A `/home` pasta só será alterada se você alterar o nome do devfile ou do componente devfile. Se você alterar o nome do devfile ou do componente devfile, o conteúdo do `/home` diretório será substituído e os dados do `/home` diretório anterior não poderão ser recuperados.

Se você criar um ambiente de desenvolvimento com um repositório de origem que não contém um devfile em sua raiz, ou se você criar um ambiente de desenvolvimento sem um repositório de origem, um devfile universal padrão será aplicado automaticamente ao repositório de origem. A mesma imagem padrão de devfile universal é usada para todos os IDEs. CodeCatalyst atualmente suporta

a versão 2.0.0 do devfile. Para obter mais informações sobre o devfile, consulte [Esquema Devfile - Versão 2.0.0](#).

 Note

Você só pode incluir imagens de contêiner público em seu devfile.

Observe que os ambientes de desenvolvimento conectados ao 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

- [Editando um arquivo de desenvolvimento do repositório 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 devfile do repositório usando o modo de recuperação](#)
- [Especificando imagens de devfile universais para um ambiente de desenvolvimento](#)
- [Comandos Devfile](#)
- [Eventos Devfile](#)
- [Componentes do Devfile](#)

Editando um arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento

Use o procedimento a seguir para editar um arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento.

Editando um devfile de repositório para um ambiente de desenvolvimento no CodeCatalyst

Para editar o arquivo devfile do repositório

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, escolha Código.
4. Escolha Repositórios de origem.
5. Escolha o repositório de origem que contém o devfile que você deseja editar.
6. Na lista de arquivos, escolha o `devfile.yaml` arquivo.
7. Selecione a opção Editar.
8. Edite o devfile.
9. Escolha Confirmar ou crie uma pull request para que um membro da equipe possa revisar e aprovar as alterações.

Note

Se você editar seu devfile, precisará reiniciá-lo para que as alterações entrem em vigor. Isso pode ser feito executando `/aws/mde/mde start --location devfile.yaml`. Se houver algum problema ao iniciar seu 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 `/projects` pasta do ambiente.

```
{
    "status": "STABLE",
    "location": "devfile.yaml"
}
```

Você também pode mover o devfile padrão para o seu repositório `/projects/devfile.yaml` de código-fonte. Para atualizar a localização do devfile, use o seguinte comando: `/aws/mde/mde start --location repository-name/devfile.yaml`.

Editando um devfile de repositório 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 depois 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, deverá confirmar e enviar suas alterações para o repositório de origem ou criar uma pull request para que um membro da equipe possa revisar e aprovar as edições do devfile.

- [Editando o arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento no AWS Cloud9](#)
- [Editando o devfile do repositório 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 obter mais informações sobre o devfile, consulte [Esquema Devfile - Versão 2.0.0](#).

Atributo	Tipo
<code>exec</code>	Command
<code>postStart</code>	Evento
<code>container</code>	Componente
<code>args</code>	Propriedades do componente
<code>env</code>	Propriedades do componente
<code>mountSources</code>	Propriedades do componente
<code>volumeMounts</code>	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 registros 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 registros.

Solução de problemas de um devfile do repositório 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 seu ambiente e corrigir seu devfile. No modo de recuperação, a execução `/aws/mde/mde status` não conterá a localização do seu devfile.

```
{
    "status": "STABLE"
}
```

Você pode verificar o erro nos registros abaixo `/aws/mde/logs`, corrigir o devfile e tentar executar `/aws/mde/mde start` novamente.

Especificando 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, CodeCatalyst fornece essa imagem e contém ferramentas que são mantidas pelo CodeCatalyst. Para permanecer notificado sobre o lançamento de novas imagens, consulte [Assinando notificações de imagem universais com o SNS](#).

Note

A `public.ecr.aws/aws-mde/universal-image:latest` imagem obtém a `public.ecr.aws/aws-mde/universal-image:3.0` imagem.

A Amazon CodeCatalyst oferece suporte às seguintes imagens de devfile.

Versão de imagem	Identificador da imagem
Universal image 1.0	<code>public.ecr.aws/aws-mde/universal-image:1.0</code>
Universal image 2.0	<code>public.ecr.aws/aws-mde/universal-image:2.0</code>

Versão de imagem	Identificador da imagem
Universal image 3.0	public.ecr.aws/aws-mde/universal-image:3.0

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

- [Assinando notificações de imagem universais com o SNS](#)
- [Versões de tempo de execução do Universal Image 1.0](#)
- [Versões de tempo de execução do Universal Image 2.0](#)
- [Versões de tempo de execução do Universal Image 3.0](#)

Assinando notificações de imagem universais com o SNS

CodeCatalyst fornece um serviço universal de notificação por imagem. Você pode usá-lo para assinar um tópico do Amazon Simple Notification Service (SNS) que notifica você quando atualizações CodeCatalyst universais de imagens forem lançadas. Para obter 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 atualizações CodeCatalyst universais de imagens.

Mensagem de amostra

```
{
  "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 assinar atualizações de imagens CodeCatalyst universais usando o console do Amazon SNS

1. [Abra o console do Amazon SNS no painel.](#)
2. Na barra de navegação, escolha sua Região da AWS.
3. No painel de navegação, escolha Subscriptions (Assinaturas) e, em seguida, 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. Esse endereço de e-mail será usado para receber notificações.
7. Selecione Criar assinatura.
8. Você receberá um e-mail de confirmação com o assunto "AWS Notificação - Confirmação de assinatura". Abra o e-mail e escolha Confirmar assinatura.

Para cancelar a assinatura de atualizações de imagens CodeCatalyst universais usando o console do Amazon SNS

1. [Abra o console do Amazon SNS no painel.](#)
2. Na barra de navegação, escolha sua Região da AWS.
3. No painel de navegação, escolha Assinaturas e selecione a assinatura da qual você deseja cancelar a assinatura.
4. Escolha Ações e, em seguida, escolha Excluir assinaturas.

5. Escolha Excluir.

Versões de tempo de execução do Universal Image 1.0

A tabela a seguir lista os tempos de execução disponíveis para `universal-image:1.0`.

universal-image:1.0 versões de tempo de execução

Nome do tempo de execução	Version (Versão)	Versão secundária mais recente e principal específica
clipe da AWS	2.11	<code>aws-cli: 2.x</code>
docker compose	2,16	<code>docker-compose: 2.x</code>
dotnet	6.0	<code>dotnet: 6.x</code>
	7.0	<code>dotnet: 7.x</code>
golang	1,19	<code>golang: 1.x</code>
java	corretto11	<code>java: corretto11.x</code>
	corretto17	<code>java: corretto17.x</code>
nodejs	14.20	<code>nodejs: 14.x</code>
	16.19	<code>nodejs: 16.x</code>
openssl	1,0	<code>openssl: 1.x</code>
	1.1	
php	7.2	<code>php: 7.x</code>
python	3.9	<code>python: 3.x</code>
	3.10	
ruby	3.1	<code>ruby: 3.x</code>
terraform	1.4	<code>terraform: 1.x</code>

Versões de tempo de execução do Universal Image 2.0

A tabela a seguir lista os tempos de execução disponíveis para `universal-image:2.0`.

universal-image:2.0 versões de tempo de execução

Nome do tempo de execução	Version (Versão)	Versão secundária mais recente e principal específica
clipe da AWS	2.11	<code>aws-cli: 2.x</code>
docker compose	2,17	<code>docker-compose: 2.x</code>
dotnet	6.0	<code>dotnet: 6.x</code>
	7.0	<code>dotnet: 7.x</code>
golang	1,20	<code>golang: 1.x</code>
java	corretto11	<code>java: corretto11.x</code>
	corretto17	<code>java: corretto17.x</code>
nodejs	16.19	<code>nodejs: 16.x</code>
openssl	1,0	<code>openssl: 1.x</code>
	1.1	
php	7.2	<code>php: 7.x</code>
python	3.9	<code>python: 3.x</code>
	3.10	
ruby	3.2	<code>ruby: 3.x</code>
terraform	1.4	<code>terraform: 1.x</code>

Versões de tempo de execução do Universal Image 3.0

A tabela a seguir lista os tempos de execução disponíveis para `universal-image:3.0`.

universal-image:3.0 versões de tempo de execução

Nome do tempo de execução	Version (Versão)	Versão secundária mais recente e principal específica
clipe da AWS	2.11	aws-cli: 2.x
docker compose	2,17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x
golang	1,21	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	18.17	nodejs: 18.x
	20.6	nodejs: 20.x
openssl	3.0	openssl: 3.x
php	8.2	php: 8.x
python	3.9	python: 3.x
	3.11	
ruby	3.2	ruby: 3.x
terraform	1.5	terraform: 1.x

Comandos Devfile

Atualmente, CodeCatalyst só oferece suporte a exec comandos em seu devfile. Para obter mais informações, consulte [Adicionar comandos na documentação](#) do Devfile.io.

O exemplo a seguir mostra como especificar exec comandos em seu devfile.

```
commands:
- id: setupscript
  exec:
    component: test
    commandLine: "chmod +x script.sh"
    workingDir: /projects/devfiles
- id: executescrpt
  exec:
    component: test
    commandLine: "./projects/devfiles/script.sh"
- id: updateyum
  exec:
    component: test
    commandLine: "yum -y update --security"
```

Depois de se conectar ao seu ambiente de desenvolvimento, você pode executar comandos definidos por meio do terminal.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescrpt
```

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 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 `postStart` eventos em seu 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 `postStart` comandos especificados na ordem em que foram definidos. Se um comando falhar, o Dev Environment continuará em execução e a saída da execução será armazenada nos registros abaixo/`aws/mde/logs`.

Componentes do Devfile

Atualmente, CodeCatalyst só oferece suporte a `container` componentes em seu devfile. Para obter mais informações, consulte [Adicionar componentes na documentação](#) do Devfile.io.

O exemplo a seguir mostra como adicionar um comando de inicialização ao seu contêiner em seu 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.

Associando 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 de 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.

Você só pode associar um ambiente de desenvolvimento a uma conexão de VPC após a criação do ambiente de desenvolvimento. Você não pode alterar a conexão VPC associada ao seu ambiente de desenvolvimento depois de criá-la. Se quiser usar uma conexão VPC diferente, você precisa excluir seu ambiente de desenvolvimento atual e criar um novo.

Important

Ambientes de desenvolvimento com uma conexão VPC não oferecem suporte a [repositórios de origem de terceiros](#) vinculados a CodeCatalyst

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 a função de administrador de espaço ou a função de usuário avançado para gerenciar conexões de VPC no nível do espaço. Para obter mais informações sobre VPCs, consulte [Gerenciando 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 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 [Preços do](#) e [Solução de problemas com ambientes de desenvolvimento](#).

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 [Preços do](#) e [Solução de problemas com ambientes de desenvolvimento](#).

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 [Preços do](#) e [Solução de problemas com ambientes de desenvolvimento](#).

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.

CodeCatalyst suporta os seguintes formatos de pacote:

- npm

Os pacotes em um repositório de pacotes podem ser descobertos e compartilhados entre os membros do projeto que contém o repositório.

Para adicionar pacotes a um repositório, configure um gerenciador de pacotes para usar o endpoint (URL) do repositório. Em seguida, você pode usar o gerenciador de pacotes para publicar pacotes no repositório.

Você pode configurar CodeCatalyst fluxos de trabalho para publicar pacotes e consumir pacotes de repositórios de CodeCatalyst pacotes. Para obter mais informações sobre o uso de pacotes em fluxos de trabalho, consulte [Publicar e importar pacotes usando um fluxo de trabalho](#).

Você pode disponibilizar pacotes em um repositório de pacotes para outro repositório no mesmo projeto adicionando-o como um repositório upstream. Todas as versões do pacote disponíveis para o repositório upstream também estão disponíveis para o repositório downstream.

Você pode disponibilizar pacotes de código aberto para seu CodeCatalyst repositório conectando repositórios públicos externos a ele. Para obter mais informações sobre repositórios upstream e conexão com repositórios externos, incluindo uma lista de repositórios compatíveis, consulte.

[Configurando e usando repositórios upstream](#)

Tópicos

- [Conceitos de pacotes](#)
- [Configurando e usando repositórios de pacotes](#)
- [Configurando e usando repositórios upstream](#)

- [Conectando-se a repositórios externos públicos](#)
- [Publicando e modificando pacotes](#)
- [Uso de npm](#)
- [Cotas para pacotes](#)

Conceitos de 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 popular)
- 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 pacotes

Alguns formatos de pacotes oferecem suporte a nomes de pacotes hierárquicos para organizar 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 de. 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. Se você não tem como agrupar nomes de pacotes, pode ser mais difícil evitar colisões de nomes.

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 no CodeCatalyst qual está associado a uma versão do pacote, como um `.tgz` arquivo npm.

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](#). Cada repositório de pacotes fornece endpoints para buscar e publicar pacotes usando ferramentas como a CLI (`npm`) do Node.js. Você pode criar até 1.000 repositórios de pacotes em cada espaço.

Você pode vincular um repositório de pacotes a outro repositório usando repositórios upstream. Ao vincular um repositório de pacotes como um repositório upstream, você pode usar os pacotes no repositório vinculado por meio do repositório configurado. Para ter mais informações, consulte [Repositórios upstream](#).

Os repositórios Gateway são um tipo especial de repositório de pacotes que extrai e armazena pacotes de autoridades externas oficiais de pacotes. Para ter mais informações, consulte [Repositórios de gateway](#).

Repositórios de gateway

Um repositório de gateway é um tipo especial de repositório de pacotes conectado a uma autoridade de pacotes oficial e externa suportada. 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 do gateway. Os pacotes consumidos dessa maneira são armazenados no repositório do gateway e no repositório downstream que recebeu a solicitação original.

Os repositórios do 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-gateway` fornece pacotes npm de `npmjs.com`.

Repositórios upstream

Você pode usar CodeCatalyst para criar um relacionamento upstream entre dois repositórios de pacotes. Um repositório de pacotes é um upstream de outro quando as versões do pacote que ele contém podem ser acessadas a partir do endpoint do repositório downstream do repositório. Com um relacionamento inicial, o conteúdo dos dois repositórios de pacotes é efetivamente mesclado 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.

Configurando e usando 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

- [Criando um repositório de pacotes](#)
- [Conectando-se a um repositório de pacotes](#)
- [Editando um repositório de pacotes](#)
- [Excluindo um repositório de pacotes](#)

Criando 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 no qual você deseja criar um repositório de pacotes.
3. No painel de navegação, escolha Pacotes.
4. Na página Package repositories, escolha Create package repository.

5. Na seção Detalhes do repositório Package, 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 do repositório. Uma descrição do repositório é especialmente útil quando você tem vários repositórios em várias equipes em um projeto.
6. Na seção Editar repositórios upstream, adicione qualquer repositório de pacotes que você queira acessar por meio do seu CodeCatalyst repositório de pacotes. Você pode adicionar repositórios do Gateway para se conectar a repositórios de pacotes externos ou outros repositórios de CodeCatalyst pacotes.
 - Quando um pacote é solicitado de 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 ou usar os botões Reordenar.
7. Escolha Criar para criar seu repositório de pacotes.

Conectando-se a um repositório de pacotes

Para publicar pacotes CodeCatalyst ou para consumir 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, você pode fazer isso seguindo as instruções em [Criando um repositório de pacotes](#).

Para obter instruções sobre como conectar o gerenciador de pacotes npm a um repositório de CodeCatalyst pacotes, consulte [Configurando e usando o npm](#)

Editando um repositório de pacotes

Execute as etapas a seguir para editar a descrição e os repositórios upstream de um repositório de pacotes.

Para editar 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 você deseja editar.
3. No painel de navegação, escolha Pacotes.

4. Na página Package repositories, escolha o repositório que você deseja excluir.
5. Escolha o menu suspenso Ações e escolha Editar.
6. Edite a descrição do repositório e os repositórios upstream. Para obter mais informações sobre como repositórios upstream, consulte [Configurando e usando repositórios upstream](#).
7. Escolha Salvar.

Excluindo 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 você deseja excluir.
3. No painel de navegação, escolha Pacotes.
4. Na página Package repositories, escolha o repositório que você deseja excluir.
5. Escolha o menu suspenso Ações e escolha Excluir.
6. Analise as informações fornecidas sobre os efeitos da exclusão de um repositório de pacotes.
7. Entre de1ete no campo de entrada e escolha Excluir.

Configurando e usando repositórios upstream

Você pode conectar os repositórios de gateway e outros repositórios de pacotes como upstreams aos seus repositórios de pacotes. Isso permite que um cliente gerenciador de pacotes acesse os pacotes que estão contidos em mais de um repositório de pacotes usando um único endpoint do repositório de pacotes. A seguir estão os principais benefícios do uso de repositórios upstream:

- Você só precisa configurar seu 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.

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 públicos correspondente do repositório de gateway. Para obter uma lista dos repositórios de pacotes públicos compatíveis, consulte [Repositórios de pacotes externos compatíveis e seus 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 seu gerenciador de pacotes para usar `project-repo` para extrair pacotes de `project-repo` e `team-repo` e `npm-public-registry-gateway`, `npmjs.com` e.

Tópicos

- [Adicionar um repositório upstream](#)
- [Editando a ordem de pesquisa dos repositórios upstream](#)
- [Solicitar uma versão do pacote com repositórios upstream](#)
- [Removendo um repositório upstream](#)


Adicionar 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 Package repositories, escolha o repositório de pacotes ao qual você deseja adicionar um repositório upstream.
3. Escolha o menu suspenso Ações e escolha Editar.
4. Na seção Repositórios upstream, escolha Adicionar repositório upstream.

- Escolha a barra de pesquisa para abrir e pesquisar uma lista de repositórios disponíveis. Você pode adicionar repositórios de pacotes públicos compatíveis ou outros repositórios como CodeCatalyst repositórios upstream. Quando você encontrar o repositório que deseja adicionar, escolha-o na lista.

 Note

Nos repositórios do Gateway, há um npm-public-registry-gateway repositório. Para se conectar a autoridades públicas externas de pacotes, como npmjs.com, CodeCatalyst use um repositório 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 do gateway.

- Depois de selecionar todos os repositórios que você deseja adicionar como repositórios upstream, escolha Adicionar.
- Para obter mais informações sobre como alterar a ordem de pesquisa dos repositórios upstream, consulte. [Editando a 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 obter mais informações sobre como solicitar versões de pacotes de um repositório upstream, consulte. [Solicitar uma versão do pacote com repositórios upstream](#)

Editando a 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

- No painel de navegação, selecione Packages (pacotes).
- Na página Repositórios, escolha o repositório de pacotes cuja ordem de pesquisa do repositório upstream você deseja editar.
- Escolha o menu suspenso Ações e escolha Editar.

4. Na seção Repositórios upstream, você pode ver os repositórios upstream e sua ordem de pesquisa. Para alterar a ordem de pesquisa, arraste e solte os repositórios na lista ou use os botões Reordenar.
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, comonpm, 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 nenhum dos `downstream` repositórios upstream contiver a versão do pacote, uma `Not Found` resposta HTTP 404 será retornada 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.

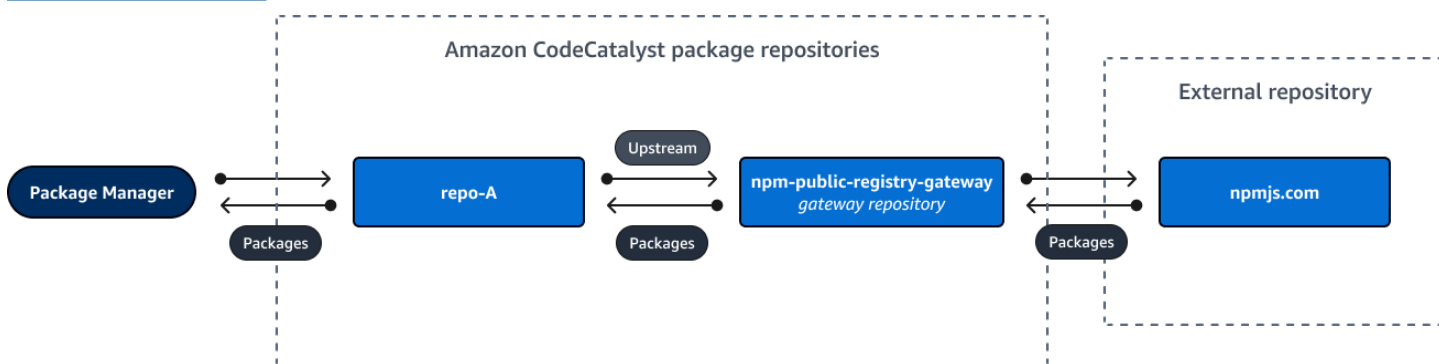
Retenção de pacotes de repositórios upstream

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).

Buscando pacotes por meio de um relacionamento 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> para `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
  
```

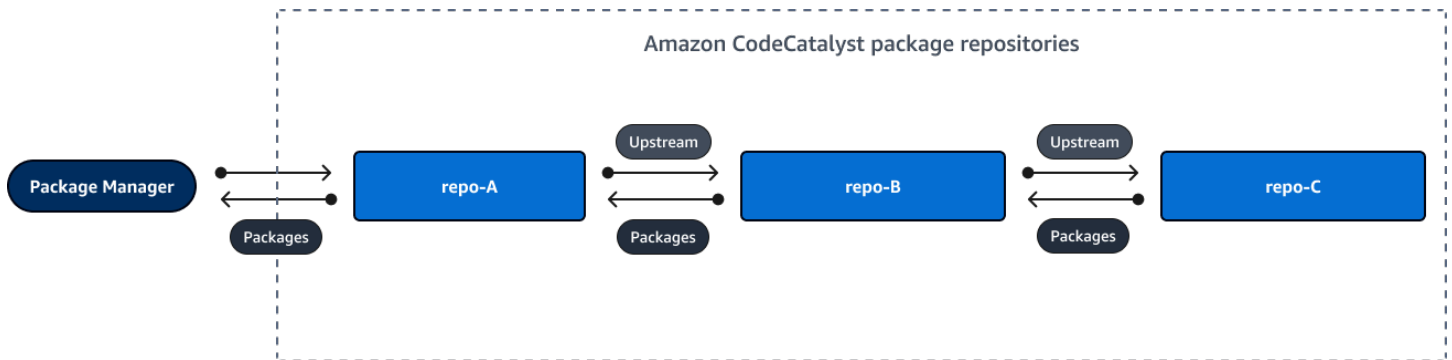
Depois de executar `npm install`, `repo-A` contém somente a versão mais recente (`lodash 4.17.20`) porque essa é a versão que foi obtida por `npm` from `repo-A`.

Como `npm-public-registry-gateway` tem uma conexão upstream externa com <https://npmjs.com>, todas as versões do pacote importadas de <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 com o `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 de <https://npmjs.com> ao longo do tempo.

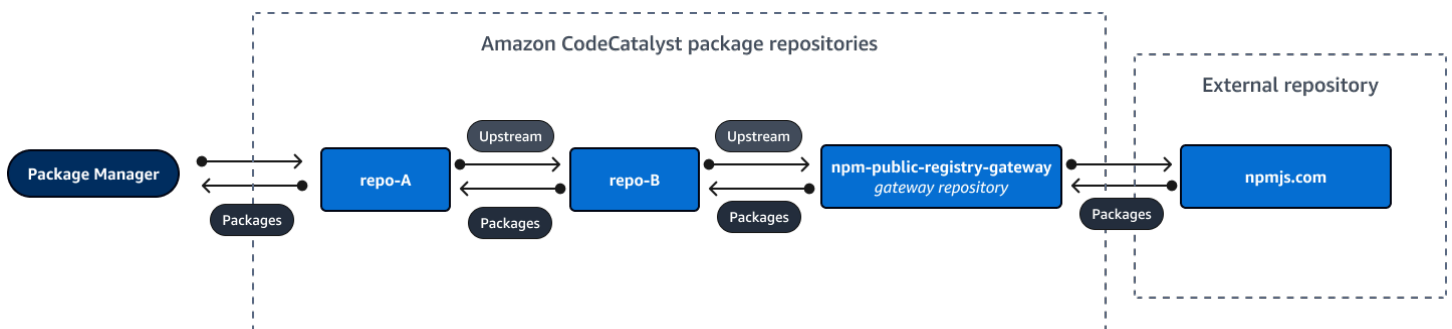
Retenção de pacotes em repositórios intermediários

CodeCatalyst permite que você encadeie repositórios upstream. Por exemplo, `repo-A` pode ter `repo-B` como um repositório upstream e `repo-B` pode ter `repo-C` como um 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 é retida no repositório `repo-B`. A versão do pacote só é mantida no repositório downstream mais distante, que neste exemplo é `repo-A`. Ele não é retido 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 para `repo-A` obter uma versão do pacote `repo-D`, a versão do pacote seria retida em `repo-A` mas não em `repo-B` ou `repo-C`.

O comportamento de retenção de pacotes é semelhante ao extrair uma versão de pacote de um repositório público de pacotes, exceto que a versão do pacote é sempre retida no repositório do gateway que tem a conexão direta upstream com o repositório público. Por exemplo, `repo-A` tem `repo-B` como um 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 **repo-A** solicitar uma versão específica do pacote, lodash 4.17.20, por exemplo, 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 é obtido, ele é retido, **repo-A** pois é o repositório mais distante a jusante e **npm-public-registry-gateway** tem a conexão upstream com o repositório externo público, npmjs.com. lodash 4.17.20 não é retido **repo-B** porque é um repositório intermediário.

Removendo 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 construções.

Para remover um repositório upstream

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Package repositories, escolha o repositório de pacotes do qual você deseja remover um repositório upstream.
3. Escolha o menu suspenso Ações e escolha Editar.
4. Na seção Repositórios upstream, encontre o repositório upstream que você deseja remover e escolha Remover.
5. Quando terminar de remover os repositórios upstream, escolha Salvar.

Conectando-se 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 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 usam pacotes do repositório do gateway.

Para se conectar a um repositório público usando repositórios de gateway

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Package repositories, nos repositórios do Gateway, você pode ver uma lista dos repositórios de gateway compatíveis e suas descrições. Para usar um repositório de gateway, primeiro você deve criá-lo. Se o repositório do gateway tiver sido criado, a data e a hora em que ele foi criado serão mostradas. Caso contrário, escolha Criar para criá-lo.
3. Escolha o repositório de pacotes que você deseja conectar ao repositório público.
4. Escolha o menu suspenso Ações e escolha Editar.
5. Para se conectar ao repositório público, adicione o repositório gateway que corresponde ao repositório público ao qual você deseja se conectar como um repositório upstream.

Na seção Editar repositórios upstream, escolha Adicionar CodeCatalyst repositório.

6. A seção Repositórios do Gateway lista todos os repositórios do gateway disponíveis. Quando você encontrar o repositório de gateway que corresponde ao repositório externo público ao qual você deseja se conectar, escolha-o na lista e escolha Adicionar.
7. Quando um pacote é solicitado de um repositório, CodeCatalyst pesquisa os repositórios upstream na ordem em que eles aparecem na lista Editar repositórios upstream. Quando um pacote é encontrado, CodeCatalyst interrompe a pesquisa. Para alterar a ordem dos repositórios upstream, arraste e solte os repositórios na lista ou use as setas de reordenação.
8. Quando terminar de adicionar e solicitar repositórios upstream, escolha Salvar.

Depois de adicionar um repositório de gateway como repositório upstream, você pode usar um gerenciador de pacotes conectado ao seu 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 do gateway e no repositório de pacotes local. Para obter 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 seus 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	Descrição	Nome do repositório do gateway
npm	Registro npm público	npm-public-registry-gateway

Publicando e modificando pacotes

Um pacote CodeCatalyst é um pacote de software e os metadados necessários para resolver dependências e instalar o software. CodeCatalyst suporta o formato de pacote npm. Esta seção fornece informações sobre como publicar, visualizar e excluir pacotes e atualizar o status da versão de um pacote.

Tópicos

- [Publicando pacotes em um repositório de CodeCatalyst pacotes](#)
- [Visualizando detalhes da versão do pacote](#)
- [Excluindo uma versão do pacote](#)
- [Atualizando o status da versão de um 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 seu gerenciador de pacotes ao seu repositório de pacotes. Para obter instruções sobre como conectar o gerenciador de pacotes npm a um repositório de CodeCatalyst pacotes, consulte [Configurando e usando o npm](#)
3. Use seu gerenciador de pacotes conectado para publicar suas 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 `lodash@1.0`, em um repositório de pacotes e `myrepo` esteja conectado a `npmjs.com` por meio de um repositório de gateway configurado como um repositório upstream. `myrepo` 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 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 não `lodash@1.1` 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`. 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 contido nele. Como o `npm` suporta apenas um único ativo por versão do pacote, para modificar uma versão publicada do pacote, você deve primeiro excluí-la.

Visualizando detalhes da versão do pacote

Você pode usar o CodeCatalyst console para ver detalhes sobre uma versão específica do pacote.

Para ver os detalhes da versão do pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Package repositories, escolha o repositório que contém a versão do pacote da qual você deseja ver os detalhes.
3. Pesquise a versão do pacote na tabela Pacotes. Você pode usar a barra de pesquisa para filtrar pacotes pelo nome do pacote. Escolha o pacote na lista.
4. Na página Package details, escolha Versions e, em seguida, escolha a versão que você deseja visualizar.

Excluindo uma versão do pacote

Você pode excluir uma versão do pacote na página de detalhes da versão do pacote no CodeCatalyst console.

Para excluir uma versão do pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Package repositories, escolha o repositório que contém a versão do pacote que você deseja excluir.
3. Pesquise e escolha o pacote na tabela.
4. Na página de detalhes do Package, escolha Versions e escolha a versão que você deseja excluir.
5. Na página de detalhes da versão do Package, escolha Version actions e, em seguida, escolha Delete.
6. Insira delete no campo de texto e escolha Excluir.

Atualizando o status da versão de um 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 obter mais informações sobre os possíveis valores de status das versões do pacote e seus significados, consulte [Status da versão do pacote](#).

Para atualizar o status da versão de um pacote

1. No painel de navegação, selecione Packages (pacotes).
2. Na página Package repositories, escolha o repositório que contém a versão do pacote cujo status você deseja atualizar.
3. Pesquise e escolha o pacote na tabela.
4. Na página Package details, escolha Versions e, em seguida, escolha a versão que você deseja visualizar.
5. Na página de detalhes da versão do Package, escolha Actions e escolha Unlist, Archive ou Dispose. Para obter informações sobre o status de cada versão do pacote, consulte [Status da versão do pacote](#).
6. Insira o texto de confirmação no campo de texto e escolha Deslistar, Arquivar ou Descartar, dependendo do status para o qual você está atualizando.

Status da versão do pacote

A seguir estão os valores possíveis para o status da versão do pacote. Você pode alterar o status da versão do pacote no console. Para ter mais informações, consulte [Atualizando o status da versão de um 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 gerenciadores de pacotes; por exemplo, na saída de `denpm view <package-name> versions`. Todos os ativos da versão do pacote estão disponíveis no repositório.
- **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 `npm package-lock.json` arquivo, 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 do seu aplicativo 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 compilação 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.

- **Descartado:** 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 é marcada como descartada, você não é cobrado pelo armazenamento dos ativos do pacote.

Além dos status na lista anterior, uma versão do pacote também pode ser excluída. Depois de excluída, a versão do pacote não está no repositório e você pode republicá-la livremente usando um gerenciador de pacotes ou uma ferramenta de compilação.

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. Se você permitir que versões de um pacote sejam adicionadas por publicação direta e ingestão de repositórios públicos, estará vulnerável a um ataque de substituição de dependências. Para ter 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.

Você deve considerar a configuração dos controles de origem do pacote para fazer com 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 de pacote padrão para 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

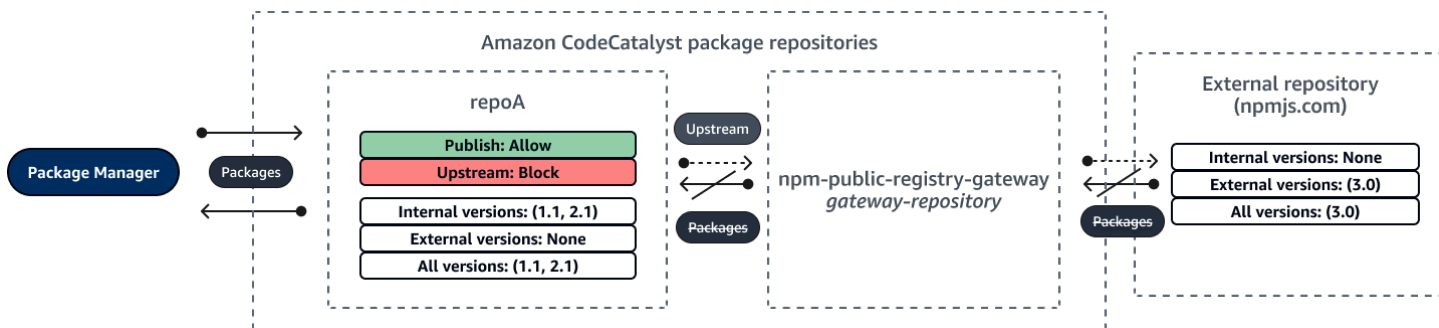
Esta seção descreve alguns cenários comuns de quando uma versão de pacote é adicionada a um repositório de CodeCatalyst pacotes. As configurações de controle de origem do pacote são definidas para novos pacotes, dependendo de como a primeira versão do pacote é adicionada.

Nos cenários a seguir, um pacote interno é publicado diretamente 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 com uma conexão externa.

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 externa com um repositório público. Seu repositório contém as versões 1.1 e 2.1 de packageA, mas a versão 3.0 é publicada no repositório público. Normalmente, o RepoA ingere a versão 3.0 depois que o pacote é 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.

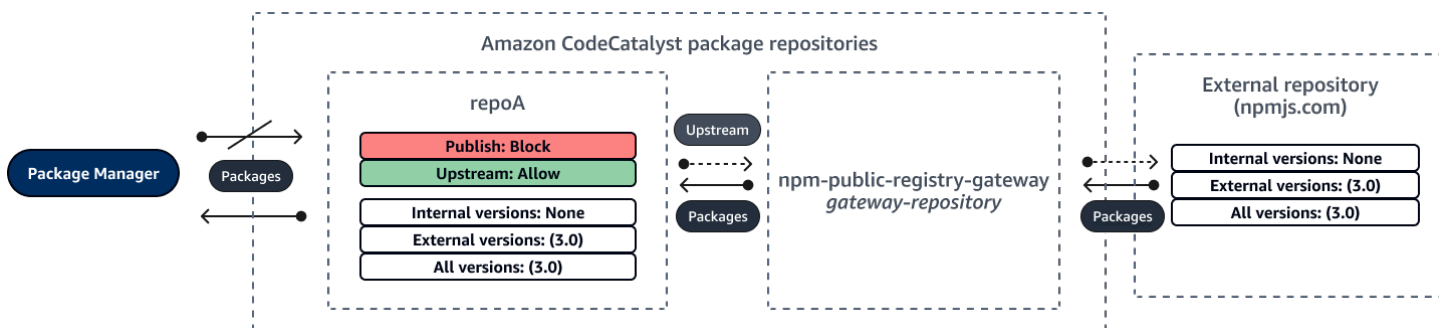


Uma versão de pacote interno é publicada para um pacote externo existente

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 tentar 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 externa com um repositório público. 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 no 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 ter 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.

Para 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, pesquise e escolha o pacote que você deseja editar.
4. Na página de resumo do pacote, escolha Editar controles de origem.
5. Em Controles de origem, escolha os controles de origem do pacote que você deseja definir para esse pacote. As duas configurações de controle de origem do pacote, Publish 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. Selecione Save (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 `lodash@1.0` npm em um repositório e `myrepo` tenha um repositório upstream com uma conexão externa com `npmjs.com`. `myrepo` 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 `loda` 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, em seguida, 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 “confusão” ou “substituição” entre o pacote desejado e o pacote malicioso, o que leva ao comprometimento do código.

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 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

- [Configurando e usando o npm](#)
- [Tratamento de tags npm](#)

Configurando e usando o 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 obter 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 seu repositório de pacotes na lista.
5. Escolha Conectar ao repositório.
6. Em Detalhes da configuração, no cliente do Package Manager, escolha npm client.
7. Escolha seu 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 tem um token, pode 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. Escolha Criar token.
 - c. Copie e armazene seu 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 seu projeto para configurar o npm com seu repositório de pacotes. Os comandos farão o seguinte:
 - Crie um `.npmrc` arquivo no nível do projeto se o seu projeto não tiver um.
 - Adicione as informações do endpoint do repositório de pacotes ao seu arquivo no nível do projeto `.npmrc`.
 - Adicione suas credenciais (PAT) ao seu arquivo em nível de usuário `.npmrc`.

Substitua os valores a seguir.

Note

Se você estiver copiando as instruções do console, os valores nos comandos a seguir serão atualizados para você e não precisarão ser alterados.

- Substitua o *nome de* CodeCatalyst usuário pelo seu nome de usuário.
- Substitua o *PAT pelo* seu CodeCatalyst PAT.
- Substitua *space_name pelo nome do* seu CodeCatalyst espaço.
- Substitua *proj_name pelo nome do* seu CodeCatalyst projeto.
- Substitua *repo_name pelo nome do* repositório do CodeCatalyst 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 npm comandos 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 a partir do npmjs são ingeridos e armazenados no repositório do gateway e no repositório de pacotes 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 [Adicionar um repositório upstream](#) e escolhendo o npm-public-registry-gateway repositório.
3. Instale pacotes com o `npm install` comando.

```
npm install package_name
```

Para obter mais informações sobre a solicitação de 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 npm comandos.

Você pode publicar um pacote npm em um repositório de CodeCatalyst pacotes com o `npm publish` comando.

```
npm publish
```

Para obter informações sobre como criar pacotes npm, consulte [Criação de módulos Node.js](#) no npm Docs.

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 npm os comandos nos quais o npm cliente faz uma ou mais solicitações ao registro no qual está configurado (por exemplo, `npm config set registry`). Verificou-se que esses comandos funcionam corretamente quando invocados em um repositório de CodeCatalyst pacotes.

Command	Descrição
bugs	Adivinha a localização do URL do rastreador de bugs de um pacote e, em seguida, ele tenta abri-lo.
ci	Instala um projeto do zero.
deprecate	Deprecia uma versão de um pacote.
dist-tag	Modifica as tags de distribuição do pacote.
docs	Adivinha a localização do URL de documentação de um pacote e, em seguida, tenta abri-lo usando o parâmetro <code>--browser config</code> .
doctor	Executa um conjunto de verificações para validar se sua instalação do npm pode gerenciar seus JavaScript pacotes.
install	Instala um pacote.
install-ci-test	Instala um projeto do zero e executa testes. Alias: <code>npm ci</code> . Esse comando executa <code>umnpm ci</code> , seguido imediatamente por <code>umnpm test</code> .

Command	Descrição
install-test	Instala o pacote e executa testes. Executa <code>umnpm install</code> , seguido imediatamente por <code>umnpm test</code> .
outdated	Verifica o registro configurado para determinar se algum pacote instalado está desatualizado.
ping	Faz ping no registro npm configurado ou fornecido e verifica a autenticação.
publish	Publica uma versão do pacote no registro.
update	Adivinha a localização da URL do repositório de um pacote e, em seguida, tenta abri-la usando o parâmetro <code>--browser config</code> .
view	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	Descrição
bin (legado)	Exibe o <code>bin</code> diretório npm.
build	Cria um pacote.
cache	Manipula o cache de pacotes.
completion	Ativa o preenchimento de guias em todos os comandos npm.

Command	Descrição
config	Atualiza o conteúdo do usuário e dos arquivos <code>npmrc</code> globais.
dedupe	Pesquisa a árvore de pacotes local e tenta simplificar a estrutura movendo as dependências mais para cima na árvore, onde elas podem ser compartilhadas de forma mais eficaz por vários pacotes dependentes.
edit	Edita um pacote instalado. Seleciona uma dependência no diretório de trabalho atual e abre o diretório do pacote no editor padrão.
explore	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.
help	Recebe ajuda no npm.
help-search	Pesquisa a documentação de ajuda do npm.
init	Cria um arquivo <code>package.json</code> .
link	Vincula simbolicamente um diretório de pacotes.
ls	Lista pacotes instalados.
pack	Cria um tarball a partir de um pacote.
prefix	Exibe um prefixo. Esse é o diretório pai mais próximo que contém um <code>package.json</code> arquivo, a menos que também <code>-g</code> seja especificado.

Command	Descrição
prune	Remove pacotes que não estão listados na lista de dependências do pacote pai.
rebuild	Executa o comando <code>npm build</code> nas pastas correspondentes.
restart	Executa os scripts de parada, reinicialização e inicialização de um pacote e os pré-scripts e pós-scripts associados.
raiz	Imprime o <code>node_modules</code> diretório efetivo para saída padrão.
run-script	Executa scripts de pacotes arbitrários.
shrinkwrap	Bloqueia as versões de dependência para publicação.
uninstall	Desinstala um pacote.

Comandos incompatíveis

Esses npm comandos não são compatíveis com repositórios de CodeCatalyst pacotes.

Command	Descrição	Observações
access	Define o nível de acesso nos pacotes publicados.	CodeCatalyst usa um modelo de permissão diferente do repositório público npmjs.
adduser	Adiciona uma conta de usuário de registro	CodeCatalyst usa um modelo de usuário diferente do repositório público npmjs.

Command	Descrição	Observações
audit	Executa uma auditoria de segurança.	CodeCatalyst atualmente não vende dados de vulnerabilidade de segurança.
hook	Gerencia hooks do npm, incluindo adição, remoção, listagem e atualização.	CodeCatalyst atualmente não oferece suporte a nenhum mecanismo de notificação de alterações.
login	Autentica um usuário. Este é um alias para <code>npm adduser</code> .	CodeCatalyst usa um modelo de autenticação diferente do repositório público npmjs. Para mais informações, consulte Configurando o npm com CodeCatalyst .
logout	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.
owner	Gerencia proprietários de pacotes.	CodeCatalyst usa um modelo de permissões diferente do repositório público npmjs.
profile	Altera as configurações no seu perfil de registro.	CodeCatalyst usa um modelo de usuário diferente do repositório público npmjs.

Command	Descrição	Observações
pesquisa	Pesquisa no registro por pacotes que correspondam aos termos de pesquisa.	CodeCatalyst não suporta o <code>search</code> comando.
star	Marca seus pacotes favoritos.	CodeCatalyst atualmente não suporta nenhum mecanismo de favoritos.
stars	Exibe pacotes marcados como favoritos.	CodeCatalyst atualmente não suporta nenhum mecanismo de favoritos.
team	Gerencia equipes e membros de equipes.	CodeCatalyst usa um modelo de associação de usuário e grupo diferente do repositório público npmjs.
token	Gerencia seus tokens de autenticação.	CodeCatalyst usa um modelo diferente para obter tokens de autenticação. Para mais informações, consulte Configurando o npm com CodeCatalyst .
unpublish	Remove um pacote do registro.	CodeCatalyst não suporta a remoção de uma versão de pacote de um repositório usando o cliente npm. Você pode excluir um pacote no console.
whoami	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 fornecer 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 para cada fluxo (por exemplo, `stable`, `betadev`, `canary`). Para obter mais informações, consulte [dist-tag](#) no npm Docs.

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 apenas a tag mais recente para versões de lançamento estáveis. Outras tags são usadas para versões instáveis ou de pré-lançamento.

Editando tags com o cliente npm

Os três npm `dist-tag` comandos (`add`, `rm`, `els`) 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` que está 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 `web-helper` pacote no repositório R, ele recebe as tags mais recentes e alfa. 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)	1 de janeiro de 2023
U	web-helper	latest (alias para a versão 1.0.1)	1.º de junho de 2023

Nesse caso, quando o cliente npm busca as tags do pacote web-helper no repositório R, a tag mais recente terá um alias para a versão 1.0.1 porque foi atualizada pela 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. `npm update`

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 CodeCatalyst](#).

Recurso	Cota padrão
Repositórios de pacotes	Máximo de 1000 por espaço.
Repositórios upstream diretos	Máximo de 10 por repositório de pacotes.
Repositórios de pacotes upstream pesquisados	Máximo de 25 repositórios upstream pesquisados por versão de pacote solicitada.
Tamanho do arquivo do ativo Package	Máximo de 5 GB por ativo do pacote.
Ativos do Package	Máximo de 150 por versão do pacote.

Crie, teste e implante com fluxos de trabalho em CodeCatalyst

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.

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas durante a execução de um fluxo de trabalho. Um fluxo de trabalho também define os eventos, ou gatilhos, que causam o início do fluxo de trabalho. 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Sobre o arquivo de definição do fluxo de trabalho

Um arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve seu fluxo de trabalho. O arquivo é armazenado em uma `~/ .codecatalyst/workflows/` pasta na raiz do seu [repositório de origem](#). O arquivo pode ter uma extensão `.yaml` ou `.yml`.

Veja a seguir um exemplo de um arquivo simples de definição de fluxo de trabalho. 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	Descrição
<pre>Name: MyWorkflow</pre>	<p>Especifica o nome do fluxo de trabalho. Para obter mais informações sobre a <code>Name</code> propriedade, consulte Propriedades de nível superior.</p>
<pre>SchemaVersion: 1.0</pre>	<p>Especifica a versão do esquema do fluxo de trabalho. Para obter mais informações sobre a <code>SchemaVersion</code> propriedade, consulte Propriedades de nível superior.</p>
<pre>RunMode: QUEUED</pre>	<p>Indica como CodeCatalyst lida com várias execuções. Para obter mais informações sobre o modo de execução, consulte Configurando o comportamento de enfileiramento das execuções.</p>
<pre>Triggers:</pre>	<p>Especifica a lógica que fará com que a execução de um fluxo de trabalho seja iniciada. Para obter mais informações sobre gatilhos, consulte Iniciando um fluxo de trabalho executado automaticamente com gatilhos.</p>
<pre>- Type: PUSH Branches: - main</pre>	<p>Indica que o fluxo de trabalho deve ser iniciado sempre que você envia o código para a <code>main</code> ramificação do repositório de origem padrão. Para obter mais informações sobre a fonte do fluxo de trabalho, consulte Conectando um fluxo de trabalho a um repositório de origem.</p>

Linha	Descrição
<pre>Actions:</pre>	<p>Define as tarefas a serem executadas durante a execução de um fluxo de trabalho. Neste exemplo, a Actions seção define uma única ação chamada <code>Build</code>. Para obter mais informações sobre ações, consulte Configurando as ações que um fluxo de trabalho executa.</p>
<pre>Build:</pre>	<p>Define as propriedades da <code>Build</code> ação. Para obter mais informações sobre a ação de criação, consulte Construindo com fluxos de trabalho.</p>
<pre>Identifier: aws/builddev1</pre>	<p>Especifica o identificador exclusivo e codificado para a ação de criação.</p>
<pre>Inputs: Sources: - WorkflowSource</pre>	<p>Indica que a ação de criação deve procurar no repositório de <code>WorkflowSource</code> origem os arquivos necessários para concluir o processamento. Para ter mais informações, consulte Conectando um fluxo de trabalho a um repositório de origem.</p>
<pre>Configuration:</pre>	<p>Contém as propriedades de configuração que são específicas para a ação de construção.</p>
<pre>Steps: - Run: docker build -t MyApp:latest .</pre>	<p>Diz à ação de construção que crie uma imagem do Docker chamada <code>MyApp</code> e a marque com <code>latest</code>.</p>

Para obter uma lista completa de todas as propriedades disponíveis no arquivo de definição do fluxo de trabalho, consulte [Definição de fluxo de trabalho YAML](#).

Usando os editores visual e YAML do CodeCatalyst console

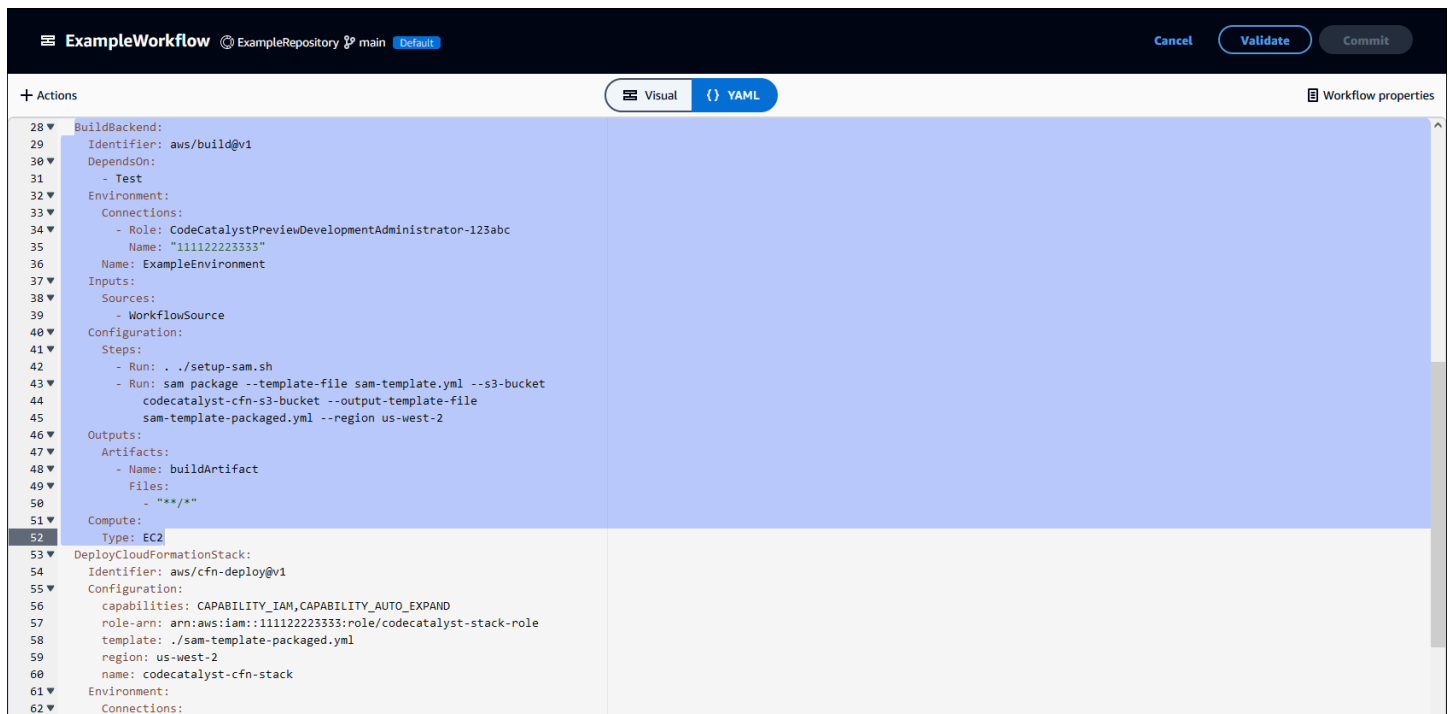
Para criar e editar o arquivo de definição do fluxo de trabalho, você pode usar seu editor preferido, mas recomendamos usar o editor visual ou o editor YAML do CodeCatalyst console. Esses editores oferecem validação útil de arquivos para ajudar a garantir que os nomes, valores, aninhamento, espaçamento, capitalização e assim por diante 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 CodeCatalyst console interface. At the top, there's a header with 'ExampleWorkflow', 'ExampleRepository', and 'main' tabs, along with 'Cancel', 'Validate', and 'Commit' buttons. Below the header, there are tabs for 'Visual' and 'YAML'. The main area is divided into two parts:

- Workflow diagram (1):** A visual flowchart showing a sequence of actions: 'Source' (workflowSource ExampleRepository main) with 'Triggers: Push', followed by 'Test' (aws/managed-test@v1), 'BuildBackend' (aws/build@v1, Environment: ExampleEnvironment, Production), and 'DeployCloudFormationStack' (aws/cfn-deploy@v1, Environment: ExampleEnvironment, Production).
- Configuration area (2):** A panel for configuring the 'BuildBackend' action. It includes sections for 'Inputs', 'Configuration', and 'Outputs'. The 'Configuration' section is active, showing fields for 'Action name' (BuildBackend), 'Compute type' (EC2), 'Compute fleet - optional' (Choose compute), 'Environment/account/role - optional' (Environment: ExampleEnvironment), and 'AWS account connection' (111122223333).

Como alternativa, você pode 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.



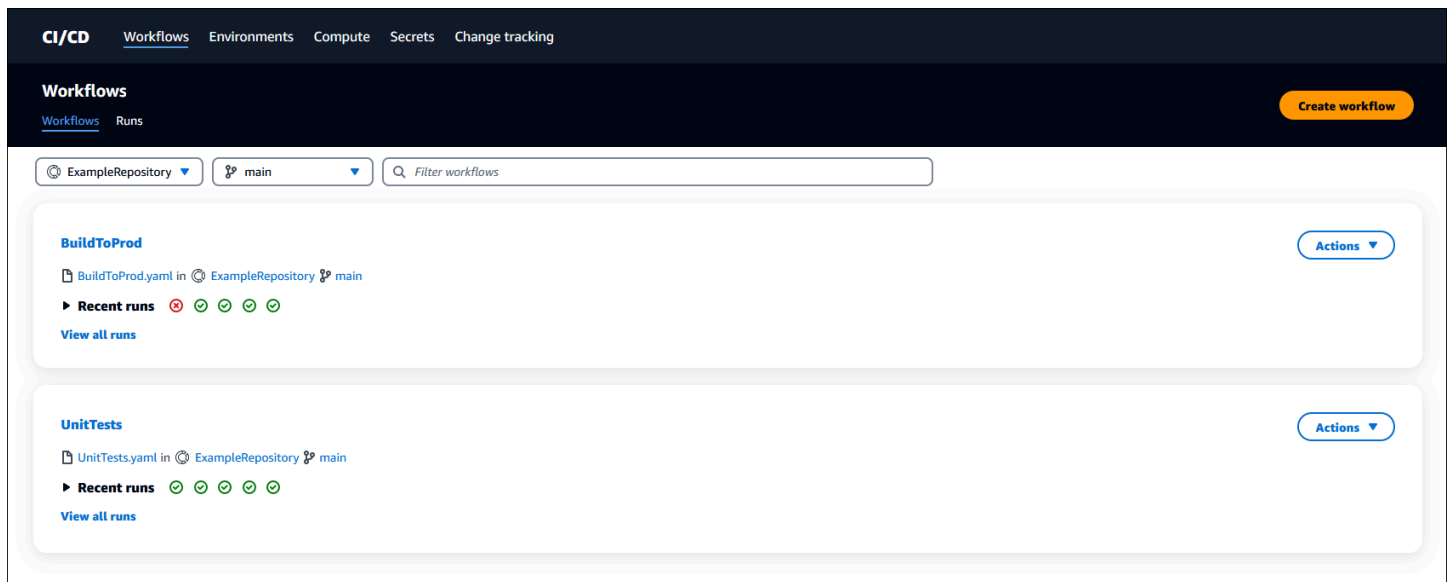
```
28 BuildBackend:
29   Identifier: aws/build@v1
30   DependsOn:
31     - Test
32   Environment:
33     Connections:
34       - Role: CodeCatalystPreviewDevelopmentAdministrator-123abc
35         Name: "111122223333"
36     Name: ExampleEnvironment
37   Inputs:
38     Sources:
39       - WorkflowSource
40   Configuration:
41     Steps:
42       - Run: ./setup-sam.sh
43       - Run: sam package --template-file sam-template.yml --s3-bucket
44             codecatalyst-cfn-s3-bucket --output-template-file
45             sam-template-packaged.yml --region us-west-2
46   Outputs:
47     Artifacts:
48       - Name: buildArtifact
49     Files:
50       - "*"/*"
51   Compute:
52     Type: EC2
53 DeployCloudFormationStack:
54   Identifier: aws/cfn-deploy@v1
55   Configuration:
56     capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
57     role-arn: arn:aws:iam::111122223333:role/codecatalyst-stack-role
58     template: ./sam-template-packaged.yml
59     region: us-west-2
60     name: codecatalyst-cfn-stack
61   Environment:
62     Connections:
```

Você pode alternar do editor visual para o editor YAML para ver o efeito que suas configurações têm no código YAML subjacente.

Descobrimo fluxos de trabalho

Você pode visualizar seu fluxo de trabalho na página de resumo dos fluxos de trabalho, junto com outros fluxos de trabalho que você configurou no mesmo projeto.

A imagem a seguir 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. Você pode escolher Execuções recentes para ver rapidamente o histórico de execuções ou escolher o nome do fluxo de trabalho para ver o código YAML do fluxo de trabalho e outras informações detalhadas.



Visualizando detalhes da execução do fluxo de trabalho

Você pode ver os detalhes da execução de um fluxo de trabalho escolhendo 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 que foi iniciada automaticamente em uma confirmação na origem. O diagrama do fluxo de trabalho indica que uma ação falhou (1). Você pode navegar até os registros (2) para ver as mensagens de registro detalhadas e solucionar problemas. Para obter mais informações sobre execuções de fluxo 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. The 'BuildBackend' action is highlighted as the failed step. A detailed log for 'BuildBackend' shows an error: 'Error: Template file not found at /codecatalyst/output/src3862/src/git-codecommit.us'.

Status	Run mode	Trigger	Start time	Duration	End time
Failed	Queued	Started by 0f8bf654	11 minutes ago	3 minutes 17 seconds	8 minutes ago

BuildBackend
 Failed | Start time: 9 minutes ago | Duration: 1 minute 10 seconds

- Restore cache < 1 second
- ./setup-sam.sh 14 seconds
- sam package --template-file sam-temple.yml --s3-bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml --region us-west-2 3 seconds

Detailed log messages

```

1 [Container] 2023/04/05 15:04:37 Running command sam package --template-file sam-temp
2
3 SAM CLI now collects telemetry to better understand customer needs.
4
5 You can OPT OUT and disable telemetry collection by setting the
6 environment variable SAM_CLI_TELEMETRY=0 in your shell.
7 Thanks for your help!
8
9 Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/develop
10
11 Error: Template file not found at /codecatalyst/output/src3862/src/git-codecommit.us

```

Próximas etapas

Para saber mais sobre os conceitos de fluxos de trabalho, consulte [Conceitos de fluxos de trabalho](#).

Para criar seu primeiro fluxo de trabalho, consulte [Introdução aos 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

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Arquivos de definição de fluxo de trabalho

Um arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve seu fluxo de trabalho. O arquivo é armazenado em uma `~/ .codecatalyst/workflows/` pasta na raiz do seu [repositório de origem](#). O arquivo pode ter uma extensão `.yml` ou `.yaml`.

Para obter mais informações sobre o arquivo de definição do fluxo de trabalho, consulte [Definição de fluxo de trabalho YAML](#).

Ações

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

Para obter mais informações sobre ações, consulte [Configurando as ações que um fluxo de trabalho executa](#).

Grupos de ação

Um grupo de ações contém uma ou mais ações. O agrupamento de ações em grupos de ações ajuda a manter seu fluxo de trabalho organizado e também permite configurar dependências entre grupos diferentes.

Para obter mais informações sobre grupos de ação, consulte [Agrupando ações em grupos de ação](#).

Artefatos

Um artefato é a saída de uma ação de fluxo de trabalho e normalmente consiste em uma pasta ou arquivo de arquivos. Os artefatos são importantes porque permitem que você compartilhe arquivos e informações entre as ações.

Para obter mais informações sobre artefatos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

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 obter mais informações sobre computação, consulte [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#).

Ambientes do

Um ambiente, que não deve ser confundido com um [ambiente de desenvolvimento](#), é onde o código é implantado. Geralmente, ele contém uma instância de um aplicativo em execução junto com sua infraestrutura associada. Você pode dar um nome ao seu ambiente, como desenvolvimento, teste, preparação ou produção. Todas as implantações CodeCatalyst geradas por um ambiente aparecerão na página Ambientes. Para configurar um ambiente, você dá a ele um nome, como `my-production-environment`, e depois o associa ao seu Conta da AWS.

Além de exibir informações de implantação, um ambiente também serve como mecanismo para atribuir funções AWS do IAM às [ações](#) do fluxo de trabalho.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

Portões

Um portão é um componente do fluxo de trabalho que você pode usar 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.

Você pode adicionar portas entre sequências de ações em um fluxo de trabalho ou antes da primeira ação (que é executada imediatamente após o download do código-fonte). Você também pode adicionar portas após a última ação, se precisar fazer isso.

Para obter mais informações sobre portões, consulte [Bloqueando a execução de um fluxo de trabalho](#).

Portões

Um portão é um componente do fluxo de trabalho que você pode usar 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.

Você pode adicionar portas entre sequências de ações em um fluxo de trabalho ou antes da primeira ação (que é executada imediatamente após o download do código-fonte). Você também pode adicionar portas após a última ação, se precisar fazer isso.

Para obter mais informações sobre portões, consulte [Bloqueando a execução de um fluxo de trabalho](#).

Relatórios

Um relatório contém detalhes sobre os testes que ocorrem durante a execução de um fluxo de trabalho. Você pode criar relatórios como um relatório de teste, um relatório de cobertura de código, um relatório de análise de composição de software e um relatório de análise estática. Você pode 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 ajudá-lo a otimizar seus aplicativos e configurações de implantação.

Para obter mais informações sobre os relatórios, consulte [Tipos de relatório de qualidade](#).

Execuções

Uma 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 obter mais informações sobre execuções, consulte [Executar um fluxo de trabalho](#).

Origens

Uma fonte, também chamada de fonte de entrada, é um repositório de origem ao qual uma [ação de fluxo](#) de trabalho se conecta para obter os arquivos necessários para realizar suas operações. Por exemplo, uma ação de fluxo de trabalho pode se conectar a um repositório de origem para obter arquivos de origem do aplicativo a fim de criar um aplicativo.

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

Variáveis

Uma variável é um par de valores-chave que contém informações que você pode referenciar em seu CodeCatalyst fluxo de trabalho.

Para obter mais informações sobre variáveis, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

Acionadores de fluxo de trabalho

Um acionador 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.

Você pode usar três tipos de gatilho:

- Push — Um gatilho de envio de código faz com que a execução de um fluxo de trabalho seja iniciada sempre que um commit é enviado.
- Pull request — Um gatilho de pull request faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma pull request é criada, revisada ou fechada.
- Agendamento — Um gatilho de agendamento faz com que a execução de um fluxo de trabalho comece em um cronograma definido por você. Considere usar um gatilho de agendamento para executar compilações noturnas de seu software para que a versão mais recente esteja pronta para seus desenvolvedores de software trabalharem na manhã seguinte.

Você pode usar gatilhos push, pull request e agendar sozinhos ou em combinação no mesmo fluxo de trabalho.

Para obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).

Introdução aos fluxos de trabalho

Neste tutorial, você aprenderá a criar e configurar seu primeiro fluxo de trabalho.

i Tip

Prefere começar com um fluxo de trabalho pré-configurado? Consulte [Criando um projeto com um blueprint](#), que inclui instruções para configurar um projeto com um fluxo de trabalho funcional, um aplicativo de amostra e outros recursos.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: criar e configurar seu fluxo de trabalho](#)
- [Etapa 2: salve seu fluxo de trabalho com um commit](#)
- [Etapa 3: Exibir os resultados da execução](#)
- [\(Opcional\) Etapa 4: Limpar](#)

Pré-requisitos

Antes de começar

- Você precisa de um CodeCatalyst espaço. Para ter mais informações, consulte [Criando um espaço](#).
- Em seu CodeCatalyst espaço, você precisa de um CodeCatalyst projeto vazio, Start from scratch, chamado:

```
codecatalyst-project
```

Para ter mais informações, consulte [Criando um projeto vazio na Amazon CodeCatalyst](#).

- No seu projeto, você precisa de um CodeCatalyst repositório chamado:

```
codecatalyst-source-repository
```

Para ter mais informações, consulte [Criando um repositório de origem](#).

Note

Se você tiver um projeto e um repositório de código-fonte existentes, poderá usá-los; no entanto, criar novos facilita a limpeza no final deste tutorial.

Etapa 1: criar e configurar seu fluxo de trabalho

Nesta etapa, você cria e configura um fluxo de trabalho que cria e testa automaticamente seu código-fonte quando as alterações são feitas.

Para criar seu fluxo de trabalho

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha Criar fluxo de trabalho.

O arquivo de definição do fluxo de trabalho aparece no editor YAML do CodeCatalyst console.

Para configurar seu fluxo de trabalho

Você pode configurar seu 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 seu fluxo de trabalho.
2. Na ação Criar, escolha + para adicionar o YAML da ação ao seu arquivo de definição de fluxo de trabalho. Seu fluxo de trabalho agora é semelhante ao seguinte.

```
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
      - 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. Escolha Visual para visualizar o arquivo de definição do fluxo de trabalho no editor visual. Os campos no editor visual permitem que você configure as propriedades YAML mostradas no editor YAML.

Etapa 2: salve seu fluxo de trabalho com um commit

Nesta etapa, você salva suas alterações. Como os fluxos de trabalho são armazenados como `.yaml` arquivos no seu repositório, você salva suas alterações com confirmações.

Para confirmar suas alterações no fluxo de trabalho

1. (Opcional) Escolha Validar para garantir que o código YAML do fluxo de trabalho seja válido.
2. Escolha Commit (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, escolha o repositório no qual você deseja salvar o fluxo de trabalho (`codecatalyst-repository`).
6. Em Nome da ramificação, escolha a ramificação na qual você deseja salvar o fluxo de trabalho (`main`).
7. Escolha Commit (Confirmar).

Seu novo fluxo de trabalho aparece na lista de fluxos de trabalho. Pode levar alguns instantes para aparecer.

Como os fluxos de trabalho são salvos com confirmações e porque o fluxo de trabalho tem um gatilho de envio de código configurado, salvar o fluxo de trabalho inicia a execução automática do fluxo de trabalho.

Etapa 3: Exibir os resultados da execução

Nesta etapa, você navega até a execução que foi iniciada a partir do seu commit e visualiza os resultados.

Para ver os resultados da execução

1. Escolha o nome do seu 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, escolha a ação de criação (por exemplo, `Build_F0`).
3. Analise o conteúdo das guias Registros, Relatórios, Configuração e Variáveis. Essas guias mostram os resultados da sua ação de criação.

Para ter mais informações, consulte [Visualizando os resultados de uma ação de construção](#).

(Opcional) Etapa 4: Limpar

Nesta etapa, você limpa os recursos que criou neste tutorial.

Para excluir recursos

- Se você criou um novo projeto para este tutorial, exclua-o. Para obter instruções, consulte [Excluir um projeto](#). A exclusão do projeto também exclui o repositório de origem e o fluxo de trabalho.

Construindo com fluxos de trabalho

Usando [CodeCatalyst fluxos de trabalho](#), você pode criar aplicativos e outros recursos.

Tópicos

- [Como faço para criar um aplicativo?](#)
- [Benefícios da ação de construção](#)
- [Alternativas à ação de construção](#)
- [Adicionando a ação de construção](#)
- [Visualizando os resultados de uma ação de construção](#)
- [Tutorial: Faça upload de artefatos para o Amazon S3](#)
- [Crie e teste a definição YAML da ação](#)

Como faço para criar um aplicativo?

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.

Uma ação de construção é um componente básico 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 de alto nível para criar um aplicativo ou recurso são as seguintes.

Para criar um aplicativo (tarefas de alto nível)

1. Em CodeCatalyst, você adiciona o código-fonte de um aplicativo que deseja criar. Para ter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#).
2. Em CodeCatalyst, você cria um fluxo de trabalho. O fluxo de trabalho é onde você define como criar, testar e implantar seu aplicativo. Para ter mais informações, consulte [Introdução aos 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 mais informações, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
4. No fluxo de trabalho, você adiciona uma ação de compilação que compila e empacota o código-fonte do seu aplicativo ou recurso. Opcionalmente, você também pode fazer com que a ação de criação execute testes de unidade, gere relatórios e implante seu aplicativo se você não quiser usar uma ação de teste ou implantação para essas finalidades. Para saber mais sobre as ações de teste e implantação, consulte [Adicionando a ação de construção](#).
5. (Opcional) No fluxo de trabalho, você adiciona uma ação de teste e uma ação de implantação para testar e implantar seu aplicativo ou recurso. Você pode escolher entre várias ações pré-configuradas para implantar seu aplicativo em diferentes destinos, como o Amazon ECS. Para obter mais informações, consulte [Testando com fluxos de trabalho](#) e [Implantação 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 compilação, teste e implantação em sequência para criar, testar e implantar seu aplicativo e seus recursos no destino. Para ter mais informações, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

Benefícios da ação de construção

Usar a ação de criação em um fluxo de trabalho tem os seguintes benefícios:

- Totalmente gerenciado — a ação de compilação elimina a necessidade de configurar, corrigir, atualizar e gerenciar seus próprios servidores de compilação.
- Sob demanda — A ação de construção é dimensionada sob demanda para atender às suas necessidades de construção. Você paga somente pela quantidade de minutos de compilação que consumir. Para ter mais informações, consulte [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#).

- 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 ter mais informações, consulte [Especificando imagens do Docker do ambiente de execução](#).

Alternativas à ação de construção

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 obter 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 CodeBuild?](#) .

Adicionando a ação de construção

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, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha Visual.
6. Escolha Ações.
7. Em Ações, escolha Construir.

8. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para obter uma descrição de cada campo, consulte [Crie e teste a definição YAML da ação](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar uma ação de construção usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha YAML.
6. Escolha Ações.
7. Em Ações, escolha Construir.
8. Modifique as propriedades no código YAML de acordo com suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Crie e teste a definição YAML da ação](#).
9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Definição de ação de construção

A ação de criação é definida como um conjunto de propriedades YAML dentro do arquivo de definição do fluxo de trabalho. Para obter informações sobre essas propriedades, consulte [Crie e teste a definição YAML da ação](#) no [Definição de fluxo de trabalho YAML](#).

Visualizando os resultados de uma ação de construção

Use as instruções a seguir para visualizar os resultados de uma ação de criação, incluindo os registros, relatórios e variáveis gerados.

Para ver os resultados de uma ação de criação

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
3. No diagrama do fluxo de trabalho, escolha o nome da sua ação de criação, por exemplo, Construir.
4. Para ver os registros da execução da compilação, escolha Logs. Os registros das várias fases de construção são exibidos. Você pode expandir ou reduzir os registros conforme necessário.
5. Para visualizar os relatórios de teste produzidos pela ação de criação, escolha Relatórios ou, no painel de navegação, escolha Relatórios. Para ter mais informações, consulte [Tipos de relatório de qualidade](#).
6. Para visualizar a configuração usada para a ação de compilação, escolha Configuração. Para ter mais informações, consulte [Adicionando a ação de construção](#).
7. Para visualizar as variáveis usadas pela ação de criação, escolha Variáveis. Para ter mais informações, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

Tutorial: Faça upload de artefatos para o Amazon S3

Neste tutorial, você aprende a fazer upload de artefatos em um bucket do Amazon S3 usando CodeCatalyst [um](#) fluxo de trabalho que inclui algumas ações [de](#) criação. Essas ações são executadas em série quando o fluxo de trabalho é iniciado. A primeira ação de construção gera dois arquivos `Hello.txt` e `Goodbye.txt` os agrupa em um artefato de construção. A segunda ação de construção carrega o artefato para o Amazon S3. Você configurará o fluxo de trabalho para ser executado sempre que enviar um commit para seu 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 ter mais informações, consulte [Criando um espaço](#).
- Em seu espaço, você precisa de um CodeCatalyst projeto vazio, Start from scratch, chamado:

```
codecatalyst-artifact-project
```

Para ter 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 ter mais informações, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

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. A função é chamada de função Build.

Note

Se você já tem uma função de criação criada para outro tutorial, você também pode usá-la neste tutorial. Apenas certifique-se de 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.

Para criar uma função de criação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Criar política.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```


Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-s3-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

- 2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    }  
  ]  
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-s3-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

```
codecatalyst-s3-build-role
```

- i. Em Descrição da função, insira:

```
CodeCatalyst build role
```

- j. Selecione Criar função.

Agora você criou uma função 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 onde os `Goodbye.txt` artefatos `Hello.txt` e serão carregados.

Para criar um bucket do Amazon S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel principal, escolha Criar bucket.
3. Em Nome do bucket, insira:

```
codecatalyst-artifact-bucket
```

4. Em AWS Region (Região da AWS), escolha uma região. Este tutorial pressupõe que você escolheu US West (Oregon) `us-west-2`. Para obter informações sobre regiões suportadas pelo Amazon S3, consulte [endpoints e cotas do Amazon Simple Storage Service](#) no. Referência geral da AWS

5. Na parte inferior da página, escolha Criar bucket.
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 us-west-2 do Oeste dos EUA (Oregon).

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 obter mais informações sobre repositórios de origem, consulte [Criando um repositório de origem](#).

Como criar um repositório de origem

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

```
codecatalyst-artifact-source-repository
```

6. Selecione Create (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 blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#)

- Uma ação de construção chamada `GenerateFiles` — No gatilho, a `GenerateFiles` ação 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 `GenerateFiles` ação, a `Upload` ação executa o AWS CLI comando `aws s3 sync` para carregar os arquivos no `codecatalystArtifact` e no seu repositório de origem para o bucket do Amazon S3. AWS CLI Ele vem pré-instalado e pré-configurado na plataforma de CodeCatalyst computação, para que você não precise instalá-lo ou configurá-lo.

Para obter mais informações sobre o software pré-empacotado na plataforma de CodeCatalyst computação, consulte [Especificando imagens do Docker do ambiente de execução](#) Para obter mais informações sobre o `aws s3 sync` comando AWS CLI's, consulte [sync](#) na Referência de AWS CLI Comandos.

Para obter mais informações sobre a ação de criação, consulte [Construindo com fluxos de trabalho](#).

Para criar um fluxo de trabalho

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha Criar fluxo de trabalho.
3. Exclua o código de amostra YAML.
4. Adicione o seguinte código YAML:

```
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 em que você criou [Pré-requisitos](#).
- *codecatalyst-s3-build-role* com o nome da função de compilação que você criou 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 obter informações sobre as propriedades desse arquivo, consulte [Crie e teste a definição YAML da ação](#) o.

5. (Opcional) Escolha Validar para garantir que o código YAML seja válido antes de confirmar.
6. Escolha Commit (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, digite:

```
add initial workflow file
```

- c. Para Repositório, escolha `codecatalyst-artifact-source-repository`.
- d. Em Nome da filial, escolha principal.
- e. Escolha Commit (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 `codecatalyst-artifact-workflow.yaml` arquivo ao seu 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, em seguida, escolha 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. Escolha Carregar para ver o progresso da segunda ação de compilação.
5. Quando a ação de upload 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, escolha 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 `Hello.txt` arquivos `Goodbye.txt` e foram enviados porque faziam parte do `codecatalystArtifact` artefato. Os `README.md` arquivos `.aws/` `.git/`, e foram enviados porque estavam no seu 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 `codecatalyst-artifact-source-repository` origem.
3. Exclua o `codecatalyst-artifact-workflow` fluxo de trabalho.

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 `codecatalyst-artifact-bucket` bucket.
 - c. Exclua o `codecatalyst-artifact-bucket` bucket.
2. Limpe no IAM, da seguinte forma:
 - a. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - b. Exclua o `codecatalyst-s3-build-policy`.
 - c. Exclua o `codecatalyst-s3-build-role`.

Crie e teste a definição YAML da ação

Veja a seguir a definição YAML das ações de criação e teste. Há uma referência para duas ações porque suas propriedades YAML 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 de fluxo de trabalho YAML](#).

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 do usuário, use Ctrl+F. O elemento será listado com sua 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"
```

nome da ação

(Obrigatório)

Especifique o nome da ação. Todos os nomes das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

UI correspondente: guia de configuração/nome da ação

Identifier

(nome da ação/) Identifier

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Use `aws/build@v1` para criar ações.

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 label`

DependsOn

(nome da ação/) DependsOn

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*nome da ação*/) Compute

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(tipo de *nome da ação*/Compute/)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)
Otimizado para flexibilidade durante as corridas 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: guia de configuração/tipo de computação

Fleet

(nome da *ação*) /Compute/ **Fleet**

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 é concluída. Exemplos de frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`

UI correspondente: guia de configuração/frota de computação

Timeout

(nome da ação) 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Environment

(nome da ação) Environment

(Optional)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/ambiente

Name

(nome da ação) /Environment/ Name

(Optional)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/nome do ambiente

Connections

(nome da *ação*) */Environment/ Connections*

(Optional)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia Configuração/

Name

(nome da *ação*) */Environment/Connections/ Name*

(Optional)

Especifique o nome da conexão da conta.

UI correspondente: guia Configuração/

Role

(nome da *ação*) */Environment/Connections/ Role*

(Optional)

Especifique o nome da função do IAM que essa ação usa para acessar e operar em AWS serviços como Amazon S3 e Amazon ECR. Certifique-se de que essa função seja adicionada à conexão da sua conta. Para adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, desde que ela tenha permissões suficientes. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

Warning

Limite as permissões às exigidas pelas ações de criação e teste. Usar uma função com permissões mais amplas pode representar um risco de segurança.

UI correspondente: guia Configuração/

Caching

(*nome da ação*/) Caching

(Optional)

Uma seção na qual você pode especificar um cache para salvar arquivos em disco e restaurá-los desse cache em execuções subsequentes do fluxo de trabalho.

Para obter mais informações sobre o armazenamento em cache de arquivos, consulte [Armazenando arquivos em cache entre execuções de fluxo de trabalho](#).

UI correspondente: guia de configuração/cache de arquivos - opcional

FileCaching

(nome da *ação*) /Caching/ **FileCaching**

(Optional)

Uma seção que especifica a configuração de uma sequência de caches.

UI correspondente: guia de configuração/Cache de arquivos - opcional/Adicionar cache

nome-chave-1

(nome da ação nome-chave 1/Caching/FileCaching/)

(Optional)

Especifique o nome da sua propriedade de cache principal. Os nomes das propriedades de cache devem ser exclusivos em seu fluxo de trabalho. Cada ação pode ter até cinco entradasFileCaching.

UI correspondente: guia de configuração/Cache de arquivos - opcional/Adicionar cache/chave

Path

(nome da ação /Caching/FileCaching/ nome-chave 1/) Path

(Optional)

Especifique o caminho associado para seu cache.

UI correspondente: guia de configuração/Cache de arquivos - opcional/Adicionar cache/caminho

RestoreKeys

(nome da ação /Caching/FileCaching/ nome-chave 1/) RestoreKeys

(Optional)

Especifique a chave de restauração a ser usada como alternativa quando a propriedade primária do cache não puder ser encontrada. Os nomes das chaves de restauração devem ser exclusivos em seu fluxo de trabalho. Cada cache pode ter até cinco entradasRestoreKeys.

UI correspondente: guia Configuração/Cache de arquivos - opcional/Adicionar cache/Restaurar chaves - opcional

Inputs

(nome da ação/) Inputs

(Optional)

A Inputs seção define os dados 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 ação de construção ou ação de teste. As variáveis não contam para esse total.

Se você precisar se referir a arquivos que residem em entradas diferentes (digamos, uma fonte e um artefato), a entrada de origem é a entrada primária e o artefato é a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferenciá-las das primárias. Para obter detalhes, consulte [Exemplo: referenciando arquivos em vários artefatos](#).

UI correspondente: guia Entradas

Sources

(nome da *ação*) */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 seu arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma fonte, deverá especificar pelo menos um artefato de entrada abaixo. *action-name/Inputs/Artifacts*

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: nenhuma

Artifacts - input

(nome da *ação*) */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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Note

Se a lista suspensa Artefacts - opcional não estiver disponível (editor visual) ou se você receber erros ao validar seu YAML (editor YAML), talvez seja porque a ação suporta apenas uma entrada. Nesse caso, tente remover a entrada da fonte.

UI correspondente: guia Entradas/Artefatos - opcional

Variables - input

(nome da *ação*) */Inputs/ Variables*

(Optional)

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para habilitar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de entradas/variáveis - opcional

Outputs

(*nome da ação*/) Outputs

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

UI correspondente: guia Saídas

Artifacts - output

(nome da *ação*) */Outputs/ Artifacts*

(Optional)

Especifique o nome de um artefato gerado pela ação. Os nomes dos artefatos devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (_). Espaços, hífen (-) e outros caracteres especiais não são permitidos. Você não pode usar aspas para ativar espaços, hífen e outros caracteres especiais nos nomes dos artefatos de saída.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos

Name

(nome da *ação*) */Outputs/Artifacts/ Name*

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique o nome de um artefato gerado pela ação. Os nomes dos artefatos devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (_). Espaços, hífen (-) e outros caracteres especiais não são permitidos. Você não pode usar aspas para ativar espaços, hífen e outros caracteres especiais nos nomes dos artefatos de saída.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos/Nova saída/Nome do artefato de construção

Files

(nome da *ação*) */Outputs/Artifacts/ Files*

(Obrigatório se [Artifacts - output](#) estiver 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 seu 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. Você pode usar padrões globais 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 obter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou fonte encontrá-lo. Para obter mais informações, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#).

UI correspondente: guia Saídas/Artefatos/Nova saída/Arquivos produzidos pela compilação

Variables - output

(nome da *ação*) */Outputs/ Variables*

(Optional)

Especifique as variáveis que você deseja que a ação exporte para que estejam disponíveis para uso em ações subsequentes.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia Saídas/Variáveis/Adicionar variável

nome da variável-1

(nome da ação nome da variável-1/Outputs/Variables/)

(Optional)

Especifique o nome de uma variável que você deseja que a ação exporte. Essa variável já deve estar definida na Steps seção Inputs ou da mesma ação.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de saídas/variáveis/Adicionar variável/nome

AutoDiscoverReports

*(nome da ação) /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. Você pode limitar quais arquivos são inspecionados usando as [ExcludePaths](#) propriedades [IncludePaths](#) ou.

UI correspondente: guia Saídas/relatórios/relatórios de descoberta automática

Enabled

(nome da *ação*) */Outputs/AutoDiscoverReports/ Enabled*

(Optional)

Ative ou desative o recurso de descoberta automática.

Os valores válidos são true ou false.

Se Enabled for omitido, o padrão será. true

UI correspondente: guia Saídas/relatórios/relatórios de descoberta automática

ReportNamePrefix

(nome da *ação*) */Outputs/AutoDiscoverReports/ ReportNamePrefix*

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e ativado)

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: guia de saídas/relatórios/nome do prefixo

IncludePaths

(nome da *ação*) */Outputs/AutoDiscoverReports/ IncludePaths*

Ou

(nome da *ação nome do* */Outputs/Reports/ relatório-1/*) IncludePaths

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e ativado, 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 obter 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

Para relatórios configurados manualmente, `IncludePaths` deve ser um padrão global que corresponda a um único arquivo.

UI correspondente:

- Guia de saídas/Relatórios/Descoberta automática de relatórios/Incluir/Excluir caminhos/Incluir caminhos
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Incluir/Excluir caminhos/ Incluir caminhos***

ExcludePaths

(nome da *ação*) `/Outputs/AutoDiscoverReports/ ExcludePaths`

Ou

(nome da *ação nome do relatório-1*) `/Outputs/Reports/ relatório-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 `**/* glob`.

Note

Se o caminho do arquivo incluir um ou mais asteriscos (*) ou outros caracteres especiais, coloque o caminho entre aspas duplas (). "" Para obter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

UI correspondente:

- Guia de saídas/Relatórios/Descoberta automática de relatórios/Incluir/Excluir caminhos/Excluir caminhos
- **Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Incluir/Excluir caminhos/ Excluir caminhos**

SuccessCriteria

(nome da *ação*) `/Outputs/AutoDiscoverReports/ SuccessCriteria`

Ou

(nome da *ação nome do relatório-1*) `/Outputs/Reports/ relatório-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 ter mais informações, consulte [Configurando critérios de sucesso para relatórios](#).

UI correspondente: guia Saída/relatórios/critérios de sucesso

PassRate

(nome da *ação*) `/Outputs/AutoDiscoverReports/SuccessCriteria/ PassRate`

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para obter mais informações sobre relatórios de teste, consulte [Relatórios de teste](#).

UI correspondente: guia Saída/Relatórios/Critérios de sucesso/Taxa de aprovação

LineCoverage

*(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage***

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para obter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente: guia de saída/relatórios/critérios de sucesso/cobertura de linha

BranchCoverage

*(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage***

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura da filial são aplicados somente aos relatórios de cobertura de código. Para obter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente: guia de saída/relatórios/critérios de sucesso/cobertura de filiais

Vulnerabilities

(nome da *ação*) */Outputs/AutoDiscoverReports/SuccessCriteria/ **Vulnerabilities***

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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, você deve especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH CRITICAL as vulnerabilidades serão contabilizadas.

- O número máximo de vulnerabilidades da severidade 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 da SCA. Para obter mais informações sobre relatórios SCA, consulte [Relatórios de análise de composição de software](#).

Para especificar a severidade mínima, use a Severity propriedade. Para especificar o número máximo de vulnerabilidades, use a Number propriedade.

UI correspondente: guia Saída/relatórios/critérios de sucesso/vulnerabilidades

StaticAnalysisBug

(nome da ação) */Outputs/AutoDiscoverReports/SuccessCriteria/**StaticAnalysisBug***

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) */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, você deve especificar:

- A gravidade mínima dos bugs que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH os CRITICAL bugs serão computados.

- O número máximo de bugs da severidade 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 aos PyLint relatórios da ESLint SA. Para obter mais informações sobre relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a severidade mínima, use a Severity propriedade. Para especificar o número máximo de vulnerabilidades, use a Number propriedade.

UI correspondente: guia de saída/relatórios/critérios de sucesso/bugs

StaticAnalysisSecurity

(nome da ação) */Outputs/AutoDiscoverReports/SuccessCriteria/**StaticAnalysisSecurity***

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) */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, você deve especificar:

- A gravidade mínima das vulnerabilidades de segurança que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH as vulnerabilidades CRITICAL de segurança serão computadas.

- O número máximo de vulnerabilidades de segurança da severidade 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 relatórios PyLint da ESLint SA. Para obter mais informações sobre relatórios SA, consulte[Relatórios de análise estática](#).

Para especificar a severidade mínima, use a Severity propriedade. Para especificar o número máximo de vulnerabilidades, use a Number propriedade.

UI correspondente: guia Saída/Relatórios/Critérios de sucesso/Vulnerabilidades de segurança

StaticAnalysisQuality

(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/
StaticAnalysisQuality

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) /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, você deve especificar:

- A gravidade mínima dos problemas de qualidade que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH os problemas CRITICAL de qualidade serão computados.

- O número máximo de problemas de qualidade da severidade 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 aos PyLint relatórios da ESLint SA. Para obter mais informações sobre relatórios SA, consulte[Relatórios de análise estática](#).

Para especificar a severidade mínima, use a Severity propriedade. Para especificar o número máximo de vulnerabilidades, use a Number propriedade.

UI correspondente: guia Saída/Relatórios/Critérios de sucesso/Problemas de qualidade

StaticAnalysisFinding

(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/
StaticAnalysisFinding

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) /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 as descobertas, você deve especificar:

- A severidade mínima das descobertas que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH as CRITICAL descobertas serão computadas.

- O número máximo de descobertas da severidade 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 da SARIF SA. Para obter mais informações sobre relatórios SA, consulte [Relatórios de análise estática](#).

Para especificar a severidade mínima, use a `Severity` propriedade. Para especificar o número máximo de vulnerabilidades, use a `Number` propriedade.

UI correspondente: guia Saída/relatórios/critérios de sucesso/descobertas

Reports

(nome da *ação*) `/Outputs/ Reports`

(Optional)

Uma seção que especifica a configuração dos relatórios de teste.

UI correspondente: guia Saídas/Relatórios

nome do relatório-1

(nome da *ação nome do relatório-1*) `Outputs/Reports/`

(Obrigatório se [Reports](#) estiver incluído)

O nome que você deseja dar ao CodeCatalyst relatório que será gerado a partir de seus relatórios brutos.

UI correspondente: guia Saídas/Relatórios/Configurar manualmente os relatórios/Nome do relatório

Format

(nome da *ação nome do*) `/Outputs/Reports/ relatório-1/` Format

(Obrigatório se [Reports](#) estiver incluído)

Especifique o formato de arquivo que você está usando para seus relatórios. Os valores possíveis são.

- 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: guia Saídas/Relatórios/Configurar relatórios manualmente/Adicionar/configurar relatórios/nome-do-relatório-1/Tipo de relatório e formato do relatório

Configuration

(nome da ação/) Configuration

(Obrigatório) Uma seção na qual você pode definir as propriedades de configuração da ação.

UI correspondente: guia Configuração

Container

(nome da ação) /Configuration/ Container

(Optional)

Especifique a imagem do Docker, ou contêiner, que a ação usa para concluir seu processamento. Você pode especificar uma das [imagens ativas](#) que vêm com CodeCatalyst ela ou usar sua própria imagem. Se você optar por usar sua 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 seu processamento. Para obter informações sobre qual imagem ativa é usada por padrão, consulte [Imagens ativas](#).

Para obter mais informações sobre como especificar sua própria imagem do Docker, consulte [Atribuindo uma imagem Docker de um ambiente de tempo de execução personalizado a uma ação](#)

UI correspondente: imagem Docker do ambiente de execução - opcional

Registry

(nome da *ação*) */Configuration/Container/ Registry*

(Obrigatório se Container estiver incluído)

Especifique o registro em que sua 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, você deve criar uma [função do IAM](#) que inclua as seguintes permissões e a política de confiança personalizada. (Você pode modificar uma função existente para incluir as permissões e a política, se quiser.)

A função do IAM deve incluir as seguintes permissões em sua política de função:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

A função do IAM deve incluir a seguinte política de confiança personalizada:

```
{
```



```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

Para obter mais informações sobre a criação de funções do IAM, consulte [Como criar uma função usando políticas de confiança personalizadas \(console\)](#) no Guia do usuário do IAM.

Depois de criar a função, você deve atribuí-la à ação por meio de um ambiente. Para ter mais informações, consulte [Associando um ambiente, uma conexão de conta e uma função do IAM a uma ação de fluxo de trabalho](#).

UI correspondente: Amazon Elastic Container Registry, Docker Hub e outras opções de registro

Image

(nome da *ação*) */Configuration/Container/ Image*

(Obrigatório se Container estiver incluído)

Especifique um dos seguintes:

- Se você estiver usando um CODECATALYST registro, 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 com o nome da imagem do Docker Hub e a tag opcional.

Exemplo: `postgres:latest`

- Se você 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 você estiver usando um registro personalizado, defina a imagem com o valor esperado pelo registro personalizado.

UI correspondente: imagem docker do ambiente de tempo de execução (se o registro for **CODECATALYST**), imagem do Docker Hub (se o registro for Docker Hub), URL da imagem ECR (se o registro for Amazon Elastic Container Registry) e URL da imagem (se o registro for Outro registro).

Steps

(nome da *ação*) */Configuration/ Steps*

(Obrigatório)

Especifique os comandos do shell que você deseja executar durante a ação para instalar, configurar e executar suas ferramentas de compilação.

Aqui está um exemplo de como criar um projeto npm:

Steps:

- Run: `npm install`
- Run: `npm run build`

Aqui está 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 obter mais informações sobre a especificação de caminhos de arquivo, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#)

UI correspondente: guia de configuração/comandos Shell

Packages

(nome da *ação*) */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 aplicativos.

Para obter mais informações sobre pacotes, consulte [Publique e compartilhe pacotes de software no CodeCatalyst](#).

UI correspondente: guia Configuração/ Pacotes

NpmConfiguration

(nome da *ação*) */Configuration/Packages/ NpmConfiguration*

(Obrigatório se [Packages](#) estiver 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 obter mais informações sobre a configuração do pacote npm, consulte [Uso de npm](#).

UI correspondente: guia de configuração/pacotes/Adicionar configuração/ npm

PackageRegistries

(nome da *ação*) */Configuration/Packages/NpmConfiguration/ PackageRegistries*

(Obrigatório se [Packages](#) estiver incluído)

Uma seção na qual você pode definir as propriedades de configuração de uma sequência de repositórios de pacotes.

UI correspondente: guia de configuração/pacotes/Adicionar configuração/npm/Adicionar repositório de pacotes

PackagesRepository

(nome da *ação*) */Configuration/Packages/NpmConfiguration/PackageRegistries/**PackagesRepository***

(Obrigatório se [Packages](#) estiver 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 obter mais informações sobre repositórios de pacotes, consulte [Repositórios de pacotes](#).

UI correspondente: guia Configuração/Pacotes/Adicionar configuração/npm/Adicionar repositório de pacotes/Repositório de pacotes

Scopes

(nome da *ação*) */Configuration/Packages/NpmConfiguration/PackageRegistries/**Scopes***

(Optional)

Especifique uma sequência de escopos que você deseja definir no registro do pacote. Ao definir 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 ser prefixado com “@”.

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 obter mais informações sobre escopos, consulte [Namespaces de pacotes](#) Pacotes com [escopo definido](#).

UI correspondente: guia de configuração/pacotes/Adicionar configuração/npm/Adicionar repositório de pacotes/escopos - opcional

Testando 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](#)
- [Adicionando a ação de teste](#)
- [Visualizando os resultados de uma ação de teste](#)
- [Ignorando testes que falharam em uma ação](#)
- [Integração universal-test-runner em uma ação de teste](#)
- [Configurando relatórios de qualidade em uma ação](#)
- [Tentando novamente os casos de teste de um relatório](#)
- [Melhores práticas para testes em CodeCatalyst](#)
- [Propriedades SARIF suportadas em análises de composição de software e relatórios de análise estática](#)

Tipos de relatório de qualidade

A ação CodeCatalyst de teste da Amazon oferece suporte aos seguintes tipos de relatórios de qualidade. Para obter 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 contenham os resultados dos seus testes.

Você pode usar um relatório de teste para ajudar a solucionar problemas com seus testes. Se você tiver muitos relatórios de teste de várias compilações, poderá usar seus relatórios de teste para visualizar as taxas de falha e ajudá-lo a otimizar suas compilações.

Você pode usar os seguintes formatos de arquivo de relatório de teste:

- 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 afirmações seus testes cobrem. Uma declaração é uma instrução única, sem incluir comentários.

```
line coverage = (total lines covered)/(total number of lines)
```

Cobertura de ramificação

Mede quantas ramificações seus testes abrangem de todas as ramificações possíveis de uma estrutura de controle, como uma case declaração `if` ou.

```
branch coverage = (total branches covered)/(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 severidade válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,LOW,,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 severidade válidos, do mais para o menos severo CRITICALHIGH, são:MEDIUM,LOW,, INFORMATIONAL e.

CodeCatalyst fornece as seguintes métricas de SA:

Insetos

Identifica vários possíveis bugs encontrados em seu código-fonte. Esses bugs podem incluir problemas relacionados à segurança da memória. Veja a seguir um exemplo de 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 possíveis vulnerabilidades de segurança encontradas em seu código-fonte. 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-fonte. 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 possíveis vulnerabilidades encontradas em seu código-fonte.

CodeCatalyst suporta os seguintes formatos de arquivo de relatório SA:

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

Adicionando a ação de teste

Use o procedimento a seguir para adicionar uma ação de teste ao seu CodeCatalyst fluxo de trabalho.

Visual

Para 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, em seguida, escolha Fluxos de trabalho.

3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha Visual.
6. Escolha Ações.
7. Em Ações, escolha Testar.
8. Nas guias Entradas e Configuração, preencha os campos de acordo com suas necessidades. Para obter uma descrição de cada campo, consulte [Crie e teste a definição YAML da ação](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar uma ação de construção usando o editor YAML

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha YAML.
6. Escolha Ações.
7. Em Ações, escolha Testar.
8. Modifique as propriedades no código YAML de acordo com suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Crie e teste a definição YAML da ação](#).
9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

10. Escolha 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 obter informações sobre essas propriedades, consulte [Crie e teste a definição YAML da ação](#) no [Definição de fluxo de trabalho YAML](#).

Visualizando os resultados de uma ação de teste

Use as instruções a seguir para visualizar os resultados de uma ação de teste, incluindo os registros, relatórios e variáveis gerados.

Para ver os resultados de uma ação de teste

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
3. No diagrama do fluxo de trabalho, escolha o nome da sua ação de teste, por exemplo, Teste.
4. Para ver os registros gerados por uma ação, escolha Registros. Os registros das várias fases da ação são exibidos. Você pode expandir ou reduzir os registros conforme necessário.
5. Para visualizar os relatórios de teste produzidos pela ação de teste, escolha Relatórios ou, no painel de navegação, escolha Relatórios. Para ter mais informações, consulte [Tipos de relatório de qualidade](#).
6. Para visualizar a configuração usada para a ação de teste, escolha Configuração. Para ter mais informações, consulte [Adicionando a ação de teste](#).
7. Para visualizar as variáveis usadas pela ação de teste, escolha Variáveis. Para ter mais informações, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

Ignorando testes que falharam 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 `test2` executar sempre, mesmo que `test1` falhe.

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`. Você pode fazer isso adicionando `2>/dev/null` ao comando. Com essa modificação, o exemplo anterior ficaria parecido com o seguinte.

Steps:

```
- Run: npm install
- Run: npm run test1 2>/dev/null
- Run: npm run test2
```

No segundo trecho de código, o status do `npm install` comando será respeitado, mas qualquer erro retornado pelo `npm run test1` comando será ignorado. Como resultado, o `npm run test2` comando é executado. Ao fazer isso, você pode visualizar os dois relatórios ao mesmo tempo, independentemente de ocorrer um erro.

Integração universal-test-runner em uma ação de teste

As ações de teste se integram à ferramenta `universal-test-runner` de linha de comando de código aberto. Essa ferramenta fornece recursos de teste avançados, como repetir um ou mais casos de teste a partir de um relatório de teste. `universal-test-runner` usa o [Protocolo de Execução de Teste](#) para executar seus testes para qualquer linguagem em uma determinada estrutura. `universal-test-runner` suporta as seguintes estruturas:

- [Gradle](#)
- [Brincadeira](#)
- [Maven](#)
- [pytest](#)
- [.NET](#)

`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 `universal-test-runner` para habilitar recursos de teste avançados. Para fazer isso, instale o Node.js (14 ou superior) na imagem e, em seguida, instale `universal-test-`

runner npm usando o comando shell- Run: `npm install -g @aws/universal-test-runner`. Para obter mais informações sobre como instalar o Node.js em seu contêiner por meio de comandos shell, consulte [Instalando e atualizando o Node Version Manager](#).

Para obter mais informações sobre `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, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha Visual.
6. Escolha Ações.
7. Em Ações, escolha Testar.
8. Na guia Configuração, preencha o campo Comandos do Shell atualizando o código de amostra com as estruturas suportadas de sua escolha. Por exemplo, para usar uma estrutura compatível, você usaria um Run comando 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 obter uma descrição do campo de comandos do Shell, consulte [Steps](#).

9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha 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, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha YAML.
6. Escolha Ações.
7. Em Ações, escolha Testar.
8. Modifique o código YAML de acordo com suas necessidades. Por exemplo, para usar uma estrutura compatível, você usaria um Run comando 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 obter uma descrição da propriedade Steps, consulte [Steps](#).

9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Configurando relatórios de qualidade em uma ação

Esta seção descreve como configurar um relatório de qualidade em uma ação.

Tópicos

- [Descoberta automática e relatórios manuais](#)
- [Configurando 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 ter mais informações, consulte [Tipos de relatório de qualidade](#).

Configurando critérios de sucesso para relatórios

Você pode definir 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,,MEDIUMLOW,INFORMATIONAL. Se você definir os critérios para verificar uma vulnerabilidade em HIGH gravidade, o relatório falhará se houver pelo menos uma vulnerabilidade em HIGH gravidade ou nenhuma vulnerabilidade em HIGH gravidade, mas pelo menos uma vulnerabilidade em um nível de gravidade mais alto, como uma vulnerabilidade em CRITICAL gravidade.

Se você não especificar 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 é aprovada ou falhada.

Visual

Para configurar critérios de sucesso

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha um fluxo de trabalho contendo uma ação que gere um relatório. Esse é o relatório ao qual você deseja aplicar os critérios de sucesso. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.

3. Selecione a opção Editar.
4. Escolha 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, escolha Critérios de sucesso.

Os critérios de sucesso aparecem. Dependendo de 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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para obter mais informações sobre 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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para obter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

Cobertura de agências

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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura da filial são aplicados somente aos relatórios de cobertura de código. Para obter 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, você deve especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH CRITICAL as vulnerabilidades serão contabilizadas.

- O número máximo de vulnerabilidades da severidade 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 da SCA. Para obter mais informações sobre relatórios SCA, consulte[Relatórios de análise de composição de software](#).

Insetos

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, você deve especificar:

- A gravidade mínima dos bugs que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH os CRITICAL bugs serão computados.

- O número máximo de bugs da severidade 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 aos PyLint relatórios da ESLint SA. Para obter mais informações sobre 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, você deve especificar:

- A gravidade mínima das vulnerabilidades de segurança que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH as vulnerabilidades CRITICAL de segurança serão computadas.

- O número máximo de vulnerabilidades de segurança da severidade 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 relatórios PyLint da ESLint SA. Para obter mais informações sobre 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, você deve especificar:

- A gravidade mínima dos problemas de qualidade que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH os problemas CRITICAL de qualidade serão computados.

- O número máximo de problemas de qualidade da severidade 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 aos PyLint relatórios da ESLint SA. Para obter mais informações sobre relatórios SA, consulte[Relatórios de análise estática](#).

8. Escolha Commit (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, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

YAML

Para configurar critérios de sucesso

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha um fluxo de trabalho contendo uma ação que gere um relatório. Esse é o relatório ao qual você deseja aplicar os critérios de sucesso. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
3. Selecione a opção Editar.
4. Escolha YAML.
5. No diagrama do fluxo de trabalho, escolha a ação que você configurou para gerar CodeCatalyst relatórios.
6. No painel de detalhes, escolha a guia Saídas.
7. Na ação, na AutoDiscoverReports seção ou na Reports seção, adicione uma SuccessCriteriaproriedade junto com PassRate, LineCoverage, BranchCoverage, Vulnerabilities, StaticAnalysisBugStaticAnalysisSecurity, e StaticAnalysisQuality propriedades.

Para obter uma explicação de cada uma dessas propriedades, consulte [Crie e teste a definição YAML da ação](#) o.

8. Escolha Commit (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 sobre como iniciar um fluxo de trabalho, consulte [Iniciando 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

Tentando novamente os casos de teste de um relatório

Se seu relatório falhar devido a vários casos de teste, você poderá repetir somente esses testes individuais. Isso permite que você verifique rapidamente a qualidade dos seus casos de teste e determine as próximas etapas para resolver seus problemas, como engajar uma dependência quebrada ou iniciar uma nova execução do fluxo de trabalho. Sua ação de teste incorpora `universal-test-runner` a opção de repetir somente casos de teste selecionados, em vez de toda a ação. Você só pode repetir um conjunto de casos de teste selecionados por ação por vez e apenas cinco vezes por relatório de teste. Para ter mais informações, consulte [Integração universal-test-runner em uma ação de teste](#).

Note

Se você repetir os casos de teste em um relatório, isso não afetará o status do fluxo de trabalho que gerou o relatório original.

Use as instruções a seguir para repetir os casos de teste em seus relatórios.

Para repetir os casos de teste de um relatório

1. No painel de navegação, escolha Reports.
2. Escolha o nome do seu relatório. Você pode filtrar por nome, status, repositório, ramificação ou tipo de relatório.
3. Abaixo do nome do relatório, escolha Resultados.
4. Selecione os casos de teste que você deseja repetir, escolha Executar novamente e, em seguida, escolha Casos de teste selecionados.
5. Quando sua nova tentativa for concluída, escolha Atualizar no banner e veja os resultados atualizados.

Melhores práticas para testes em CodeCatalyst

Ao usar os recursos de teste fornecidos pelo CodeCatalyst, recomendamos que você siga essas melhores práticas.

Tópicos

- [Descoberta automática](#)
- [Critérios de sucesso](#)
- [Incluir/excluir caminhos](#)

Descoberta automática

Ao configurar ações CodeCatalyst, a descoberta automática permite que você descubra automaticamente os resultados de várias ferramentas, como relatórios de teste JUnit, e gere relatórios relevantes CodeCatalyst a partir deles. A descoberta automática ajuda a garantir que os relatórios continuem sendo gerados mesmo que os nomes ou caminhos para as saídas descobertas mudem. Quando novos arquivos são adicionados, os descobre CodeCatalyst automaticamente e produz relatórios relevantes. No entanto, se você usa a descoberta automática, é importante considerar alguns dos seguintes aspectos desse recurso:

- Quando você ativa 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, deverá configurar 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 tempo de execução. No caso de você querer que um relatório específico seja sempre produzido, você deve configurar 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, você deve configurar 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 ter mais informações, consulte [Configurando 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 os dois 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 ter mais informações, consulte [Configurando critérios de sucesso para relatórios](#).

Incluir/excluir caminhos

Ao analisar os resultados da ação, você pode ajustar a lista de relatórios que são gerados CodeCatalyst pela configuração `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

Para relatórios configurados manualmente, `IncludePaths` deve 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 `**/* glob`.

A seguir estão exemplos de possíveis padrões globais.

Padrão	Descrição
<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>
<code>testFolder</code>	Corresponde a um objeto chamado <code>testFolder</code> , tratando-o como um arquivo

Padrão	Descrição
<code>testFolder/*</code>	Combina objetos em um nível da subpasta <code>testFolder</code> , como <code>testFolder/file.xml</code>
<code>testFolder/**</code>	Combina objetos em dois níveis da subpasta <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 slash (/) 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

`ExcludePathstem` precedência sobre `IncludePaths` Se ambos `IncludePaths` `ExcludePaths` incluírem a mesma pasta, essa pasta não será examinada em busca de relatórios.

Propriedades SARIF suportadas em análises de composição de software e relatórios de análise estática

O SARIF (Static Analysis Results Interchange Format) é um formato de arquivo de saída que está disponível na análise de composição de software e nos relatórios de análise estática em.

CodeCatalyst O exemplo a seguir mostra como configurar manualmente o SARIF em um relatório de análise estática:

```
Reports:
MySARreport:
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
\$schema	Sim	O URI do esquema SARIF JSON para a versão 2.1.0.
version	Sim	CodeCatalyst suporta apenas a versão 2.1.0 do SARIF.

Nome	Obrigatório	Descrição
<code>runs[]</code>	Sim	Um arquivo SARIF contém uma matriz de uma ou mais execuções, cada uma representando uma única execução da ferramenta de análise.

Objeto `run`

Nome	Obrigatório	Descrição
<code>tool.driver</code>	Sim	Um <code>toolComponent</code> objeto 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.

Nome	Obrigatório	Descrição
<code>rules[]</code>	Sim	Uma matriz de reporting Descriptor objetos 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 usada para referenciar uma descoberta. Tamanho máximo: 1.024 caracteres
<code>name</code>	Não	O nome de exibição da regra. Tamanho máximo: 1.024 caracteres
<code>shortDescription.text</code>	Não	Uma descrição resumida da regra. Tamanho máximo: 3.000 caracteres
<code>fullDescription.text</code>	Não	Uma descrição completa da regra. Tamanho máximo: 3.000 caracteres

Nome	Obrigatório	Descrição
<code>helpUri</code>	Não	Uma string que pode ser localizada para conter o URI absoluto da documentação primária da regra. Tamanho máximo: 3.000 caracteres
<code>properties.unscore</code>	Não	Uma bandeira que indica se a descoberta do escaneamento foi pontuada.
<code>properties.score.severity</code>	Não	Um conjunto fixo de sequências de caracteres que especificam o nível de severidade da descoberta. Tamanho máximo: 1.024 caracteres
<code>properties.cvssv3_baseSeverity</code>	Não	Uma classificação de severidade qualitativa do Common Vulnerability Scoring System v3.1 .
<code>properties.cvssv3_baseScore</code>	Não	Uma pontuação básica do CVSS v3 variando de 0,0 a 10,0 .
<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 0,0 a 10,0 .

Nome	Obrigatório	Descrição
<code>properties.severity</code>	Não	Um conjunto fixo de sequências de caracteres que especificam o nível de severidade da descoberta. Tamanho máximo: 1.024 caracteres
<code>defaultConfiguration.level</code>	Não	A severidade padrão de uma regra.

Objeto **result**

Nome	Obrigatório	Descrição
<code>ruleId</code>	Sim	O identificador exclusivo da regra usada para referenciar uma descoberta. 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 mensagem de cada descoberta. Tamanho máximo: 3.000 caracteres
<code>rank</code>	Não	Um valor entre 0,0 e 100,0, inclusive, que representa a

Nome	Obrigatório	Descrição
		prioridade ou importância do resultado. Essa escala valoriza 0,0 sendo a prioridade e mais baixa e 100,0 sendo a prioridade mais alta.
level	Não	A gravidade do resultado. Tamanho máximo: 1.024 caracteres
properties.unscore	Não	Uma bandeira que indica se a descoberta do escaneamento foi pontuada.
properties.score.severity	Não	Um conjunto fixo de sequências de caracteres que especificam o nível de severidade da descoberta. Tamanho máximo: 1.024 caracteres
properties.cvssv3_baseSeverity	Não	Uma classificação de severidade qualitativa do Common Vulnerability Scoring System v3.1 .
properties.cvssv3_baseScore	Não	Uma pontuação básica do CVSS v3 variando de 0,0 a 10,0 .
properties.cvssv2_severity	Não	Se os valores do CVSS v3 não estiverem disponíveis, CodeCatalyst pesquisa os valores do CVSS v2.

Nome	Obrigatório	Descrição
<code>properties.cvssv2_score</code>	Não	Uma pontuação básica do CVSS v2 variando de 0,0 a 10,0 .
<code>properties.severity</code>	Não	Um conjunto fixo de sequências de caracteres que especificam o nível de severidade da descoberta. 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. Número máximo de <code>location</code> objetos: 10
<code>relatedLocations[]</code>	Não	Uma lista de referências de localizações adicionais na descoberta. Número máximo de <code>location</code> objetos: 50

Nome	Obrigatório	Descrição
<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.
<code>fileLocation.uri</code>	Não	O URI de retorno indicando a localização do arquivo. Isso é usado se <code>artifactLocation.uri</code> retornar vazio.

Nome	Obrigatório	Descrição
<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	Descrição
<code>fullyQualifiedName</code>	Não	Informações adicionais que descrevem a localização do resultado. Tamanho máximo: 1.024 caracteres

Objeto **fix**

Nome	Obrigatório	Descrição
<code>description.text</code>	Não	Uma mensagem que exibe uma recomendação para cada descoberta. Tamanho máximo: 3.000 caracteres

Nome	Obrigatório	Descrição
<code>artifactChanges.[0].artifactLocation.uri</code>	Não	O URI que indica a localização do artefato que precisa ser atualizado.

Implantação 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 um aplicativo?

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 esquema azul/verde). Você adiciona uma ação de implantação ao seu fluxo de trabalho usando o editor visual do CodeCatalyst console ou o editor YAML.

As etapas de alto nível para implantar um aplicativo ou recurso são as seguintes.

Para implantar um aplicativo (tarefas de alto nível)

1. No seu CodeCatalyst projeto, você adiciona o código-fonte de um aplicativo que deseja implantar. Para ter mais informações, consulte [Armazenando o código-fonte em repositórios para um projeto no CodeCatalyst](#).
2. No seu CodeCatalyst projeto, você cria um fluxo de trabalho. O fluxo de trabalho é onde você define como criar, testar e implantar seu aplicativo. Para ter mais informações, consulte [Introdução aos fluxos de trabalho](#).
3. No fluxo de trabalho, você adiciona um gatilho, uma ação de criação e, opcionalmente, uma ação de teste. Para obter mais informações, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#), [Adicionando a ação de construção](#) e [Adicionando a ação de teste](#).
4. 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 implantar ações](#).)

5. 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 seu aplicativo e seus recursos no destino. Para ter mais informações, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

Lista de ações de implantação

As seguintes ações de implantação estão disponíveis:

- Implantar AWS CloudFormation pilha — Essa ação cria uma CloudFormation pilha AWS com base em um [AWS CloudFormation modelo ou AWS Serverless Application Model modelo](#) fornecido por você. Para ter mais informações, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#).
- Implantar no Amazon ECS — Essa ação registra um arquivo de [definição de tarefa](#) que você fornece. Para ter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) com um fluxo de trabalho](#).
- Implantar no cluster Kubernetes — Essa ação implanta um aplicativo em um cluster do Amazon Elastic Kubernetes Service. Para ter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#).
- AWS CDK implantar — Essa ação implanta um [AWS CDK aplicativo](#) em AWS. Para ter mais informações, consulte [Implantando um AWS Cloud Development Kit \(AWS CDK\) 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 Contas da AWS e VPCs com ambientes CodeCatalyst](#) e [Visualização do status da implantação, confirmações e pull requests](#)

Benefícios das ações de implantação

Usar ações de implantação em um fluxo de trabalho tem os seguintes benefícios:

- **Histórico de implantação** — Visualize um histórico de suas implantações para ajudar a gerenciar e comunicar as mudanças 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. Você também pode configurar alarmes para ativar reversões de implantação.
- **Monitoramento** — Observe sua implantação à medida que ela progride nos vários estágios do seu 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 implantar ações

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 construção.

Normalmente, você usa ações de compilação se quiser implantar em um destino para o qual não existe uma ação de implantação correspondente ou se quiser ter mais controle sobre o procedimento de implantação. Para obter mais informações sobre o uso de ações de compilação para implantar recursos, consulte [Construindo 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 [Integrando GitHub ações em um fluxo de trabalho](#)

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?](#)
- [AWS CloudFormation](#) — veja [O que é AWS CloudFormation?](#)

Use CodeDeploy, CodeBuild CodePipeline, e CloudFormation serviços para implantações corporativas complexas.

Tópicos


- [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) com um fluxo de trabalho](#)
- [Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#)
- [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#)
- [Implantando um AWS Cloud Development Kit \(AWS CDK\) 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 Contas da AWS e VPCs com ambientes CodeCatalyst](#)
- [Exibindo a URL do aplicativo implantado no diagrama do fluxo de trabalho](#)
- [Removendo um alvo de implantação](#)
- [Rastreando o status de implantação por confirmação](#)
- [Visualizando os registros de implantação](#)
- [Visualização do status da implantação, confirmações e pull requests](#)

Implantação de um aplicativo no Amazon Elastic Container Service (ECS) com um fluxo de trabalho


Esta seção descreve como implantar um aplicativo em contêineres em um cluster do Amazon ECS usando um fluxo de trabalho. CodeCatalyst Para fazer isso, você deve adicionar a ação Deploy to 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 Amazon ECS](#) executado no cluster do [Amazon](#) ECS. “Instanciar uma definição de tarefa” é equivalente a implantar um aplicativo no Amazon ECS.

Para usar essa ação, você deve ter um cluster, serviço e arquivo de definição de tarefas do Amazon ECS pronto.

Para obter 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 Deploy to Amazon ECS, consulte [Tutorial: Implantar um aplicativo no Amazon ECS](#).

 Tip


Para obter um exemplo prático da ação Deploy to Amazon ECS, crie um projeto com a API Node.js AWS Fargate ou a API Java com AWS Fargate blueprint. Para ter mais informações, consulte [Criando um projeto com um blueprint](#).

Tópicos

- [Tutorial: Implantar um aplicativo no Amazon ECS](#)
- [Adicionando a ação “Implantar no Amazon ECS”](#)
- [Variáveis produzidas pela ação “Deploy to Amazon ECS”](#)
- [Definição YAML da ação “Implantar no Amazon ECS”](#)

Tutorial: Implantar um aplicativo 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 O aplicativo implantado é um site simples do Hello World construído em uma imagem Docker do servidor 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 o aplicativo.

 Tip

Em vez de seguir este tutorial, você pode usar um plano que faz uma configuração completa do Amazon ECS para você. Você precisará usar a API Node.js com AWS Fargate ou a API

Java com o AWS Fargate blueprint. Para ter mais informações, consulte [Criando um projeto com um blueprint](#).

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: configurar um AWS usuário e AWS CloudShell](#)
- [Etapa 2: implantar um aplicativo de espaço reservado no Amazon ECS](#)
- [Etapa 3: Criar um repositório de imagens 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: faça uma alteração nos arquivos de origem](#)
- [Limpeza](#)

Pré-requisitos

Antes de começar

- Você precisa de um CodeCatalyst espaço com uma AWS conta conectada. Para ter mais informações, consulte [Criando um espaço](#).
- Em seu espaço, você precisa de um CodeCatalyst projeto vazio, Start from scratch, chamado:

```
codecatalyst-ecs-project
```

Para ter 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 ter mais informações, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

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. AWS IAM Identity Center Durante este tutorial, CloudShell é o seu computador de desenvolvimento e é onde você configura AWS recursos e serviços. Exclua esse usuário após concluir o tutorial.

Note

Não use seu usuário root para este 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 AWS IAM Identity Center do usuário e o Guia AWS CloudShell do usuário.

Para criar um usuário do IAM Identity Center

1. Faça login no AWS Management Console e abra o AWS IAM Identity Center 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 ter mais informações, consulte [Criando um espaço](#).

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em Nome de usuário, digite:

CodeCatalystECSUser

4. Em Senha, escolha 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. Selecione Next (Próximo).
9. Na página Adicionar usuário aos grupos, escolha Avançar.
10. Na página Revisar e adicionar usuário, revise as informações e escolha Adicionar usuário.

Uma caixa de diálogo de senha de uso único é exibida.

11. Escolha Copiar e cole as informações de login, incluindo a URL do portal de AWS acesso e a senha de uso único.
12. Escolha Fechar.

Para criar um conjunto de permissões

Você atribuirá esse conjunto de permissões para `CodeCatalystECSUser` mais tarde.

1. No painel de navegação, escolha Conjuntos de permissões e, em seguida, escolha Criar conjunto de permissões.
2. Escolha Conjunto de permissões predefinido e, em seguida, selecione AdministratorAccess. Essa política fornece permissões completas para todos 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 ao 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 usuário do CodeCatalyst ECS

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 paraCodeCatalystECSUser. Você deveria ter copiado essas informações para um editor de texto anteriormente.

Note

Se você não tiver essas informações, acesse a página de CodeCatalystECSUser detalhes no IAM Identity Center, escolha Redefinir senha, Gerar uma senha de uso único [...] e redefina a 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 paraCodeCatalystECSUser.
5. Em Nova senha, insira uma senha e escolha 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 AWS Management Console aparece. Agora você está conectado 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 AWS Management Console 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. Verifique se uma versão aparece.

O já AWS CLI está configurado para o usuário atualCodeCatalystECSUser, portanto, não há necessidade de configurar AWS CLI chaves e credenciais, como é normalmente o caso.

Etapa 2: implantar um aplicativo de espaço reservado no Amazon ECS

Nesta seção, você implanta manualmente um aplicativo de espaço reservado no Amazon ECS. Esse aplicativo de espaço reservado será substituído pelo aplicativo Hello World implantado pelo seu fluxo de trabalho. O aplicativo de espaço reservado é o Apache Web Server.

Para obter 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 o aplicativo de espaço reservado.

Para criar a função de execução da tarefa

Essa função concede ao Amazon ECS AWS Fargate (Fargate) permissão para fazer chamadas de API em seu nome.

1. Crie uma política de confiança:
 - a. Em AWS CloudShell, insira o seguinte comando:

```
cat > codecatalyst-ecs-trust-policy.json
```

Um aviso piscando aparece no CloudShell terminal.

- b. Insira o código a seguir no prompt:

```
{
  "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 uma função 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 da função:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-task-execution-role
```

5. Observe o "Arn": valor da função, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role`. Você precisará desse Amazon Resource Name (ARN) posteriormente.

Para criar um cluster do Amazon ECS

Esse cluster conterá o aplicativo de espaço reservado Apache e, posteriormente, o aplicativo Hello World.

1. Como `CodeCatalystECSUser`, em AWS CloudShell, crie um cluster vazio:

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

2. (Opcional) Verifique se o cluster foi criado com êxito:

```
aws ecs list-clusters
```

O ARN do `codecatalyst-ecs-cluster` 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. Cole o código a seguir no prompt:

```
{
  "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/ -role codecatalyst-ecs-task-execution`

com o ARN da função de execução da tarefa em que você anotou. [Para criar a função de execução da tarefa](#)

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 de definição de tarefas no Amazon ECS

1. Como `CodeCatalystECSUser`, 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 `codecatalyst-ecs-task-def` tarefa deve aparecer na lista.

Para criar o serviço Amazon ECS

O serviço Amazon ECS executa as tarefas (e os contêineres Docker associados) do aplicativo de espaço reservado Apache e, posteriormente, do aplicativo Hello World.

1. Assim como `CodeCatalystECSUser`, mude para o console do Amazon Elastic Container Service, caso ainda não tenha feito isso.
2. Escolha o cluster que você criou anteriormente, `codecatalyst-ecs-cluster`.
3. Na guia Serviços, escolha 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. Para 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 em sua VPC no Guia do usuário da Amazon VPC](#).

- i. Em Grupo de segurança, escolha Criar um novo grupo de segurança e faça o seguinte:
 - i. Em Nome do grupo de segurança, digite:
`codecatalyst-ecs-security-group`
 - ii. Para Descrição do grupo de segurança, digite:
`CodeCatalyst ECS security group`
 - iii. Escolha Adicionar regra. Em Tipo, escolha HTTP e, em Origem, escolha Qualquer lugar.
 - j. Na parte inferior, escolha Criar.
 - k. Aguarde enquanto o serviço é criado. Isso pode levar alguns minutos.
5. Escolha a guia Tarefas e, em seguida, escolha o botão Atualizar. Verifique se todas as três tarefas têm a coluna Último status definida como Em execução.

(Opcional) Para verificar se seu aplicativo 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 `It Works!` página é exibida. Isso indica que o serviço Amazon ECS iniciou com sucesso uma tarefa que lançou um contêiner 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 um aplicativo de espaço reservado do Apache. Com todos esses itens prontos, agora você está pronto para criar um fluxo de trabalho que substituirá o aplicativo de espaço reservado Apache pelo aplicativo Hello World do tutorial.

Etapa 3: Criar um repositório de imagens do Amazon ECR

Nesta seção, você cria um repositório privado de imagens no Amazon Elastic Container Registry (Amazon ECR). Esse repositório armazena a imagem 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 Amazon ECR:

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-ecs-image-repo
```

3. Observe o `"repositoryUri"`: valor, 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. Essas funções são:

- Função de criação — concede permissão à ação de CodeCatalyst criação (no fluxo de trabalho) para acessar sua AWS conta e gravar no Amazon ECR e no 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 obter mais informações sobre as funções do IAM, consulte [Funções do IAM](#) no Guia AWS Identity and Access Management do usuário.

Note

Para economizar tempo, você pode criar uma única função, chamada de CodeCatalystWorkflowDevelopmentRole-*spaceName* função, em vez das duas funções listadas anteriormente. Para ter mais informações, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a CodeCatalystWorkflowDevelopmentRole-*spaceName* função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando as duas funções listadas anteriormente.

Para criar as funções de criação e implantação, você pode usar AWS Management Console o. ou AWS CLI o.

AWS Management Console


Para criar as funções de criação e implantação, conclua a série de procedimentos a seguir.

Para criar uma função de criação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.

- c. No painel de navegação, escolha Políticas.
- d. Escolha Criar política.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

Na primeira vez em que a função `for` usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-ecs-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-ecs-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

codecatalyst-ecs-build-role

- i. Em Descrição da função, digite:

CodeCatalyst ECS build role

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de permissões e uma política de confiança.

3. Obtenha o ARN da função de construção, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (codecatalyst-ecs-build-role).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN. Você precisará disso mais tarde.

Para criar uma função de implantação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Create Policy.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "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 a função for usada 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: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-ecs-deploy-policy

```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```

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

```

```
        "Service": [
            "codecatalyst-runner.amazonaws.com",
            "codecatalyst.amazonaws.com"
        ],
        "Action": "sts:AssumeRole"
    }
]
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-ecs-deploy-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

`codecatalyst-ecs-deploy-role`

- i. Em Descrição da função, digite:

`CodeCatalyst ECS deploy role`

- j. Selecione Criar função.

Agora você criou uma função de implantação com uma política de confiança.

3. Obtenha o ARN da função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-ecs-deploy-role`).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN. Você precisará disso mais tarde.

AWS CLI

Para criar as funções de criação e implantação, conclua a série de procedimentos a seguir.

Para criar uma política de confiança para ambas as funções

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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

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 a função 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:

```
{
```

```

    "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 "arn": valor, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy`. Você precisará desse ARN mais tarde.
4. Crie a função de criação e anexe a política de confiança a ela:

```

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/* é substituído pelo ARN da política *codecatalyst-ecs-build-policy* de compilação que você observou anteriormente.

6. Exiba os detalhes da função de criação:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-build-role
```

7. Observe o "Arn": valor da função, por exemplo,arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role. Você precisará desse ARN mais tarde.

Para criar a política de implantação e a função 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:

```
{  
  "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 a função `for` for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em

seguida, 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 "arn": valor da política de implantação, por exemplo,arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy. Você precisará desse ARN mais tarde.
4. Crie a função de implantação e anexe a política de confiança a ela:

```
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, em que *arn:aws:iam::111122223333:policy/* é substituído pelo ARN da política *codecatalyst-ecs-deploy-policy* de implantação que você observou 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 da função de implantação:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-deploy-role
```

7. Observe o "Arn": valor da função, por exemplo,arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role. Você precisará desse ARN mais tarde.

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 funções de criação e implantação à conexão da sua conta

1. Dentro CodeCatalyst, navegue até seu espaço.
2. Escolha AWS contas. 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 acessar a página.

5. Selecione Adicionar uma função existente que você criou no IAM.

Uma lista suspensa é exibida. A lista exibe todas as funções do IAM com uma política de confiança que inclui os diretores `codecatalyst-runner.amazonaws.com` e `codecatalyst.amazonaws.com` de serviço.

6. Na lista suspensa `codecatalyst-ecs-build-role`, escolha e escolha Adicionar função.

Note

Se você ver `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

7. Escolha Adicionar função do IAM, escolha Adicionar uma função existente que você criou no IAM e, na lista suspensa, escolha. `codecatalyst-ecs-deploy-role` Escolha Add role (adicionar função).

Agora você adicionou as funções de criação e implantação ao seu espaço.

8. Copie o valor do nome de CodeCatalyst exibição da Amazon. Você precisará desse valor 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 obter mais informações sobre repositórios de origem, consulte [Criando um repositório de origem](#).

Como criar um repositório de origem

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu projeto, `codecatalyst-ecs-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, digite:

```
codecatalyst-ecs-source-repository
```

6. Selecione Create (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 `index.html` arquivo — Exibe uma mensagem Hello World no navegador.
- Um `Dockerfile` — descreve a imagem base a ser usada para sua imagem do Docker e os comandos do Docker a serem aplicados a ela.
- Um `taskdef.json` arquivo — Define a imagem do Docker a ser usada ao iniciar tarefas em seu cluster.

A estrutura de pastas é a seguinte:

```
.
|- 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 [Clonando um repositório de origem](#).

Tópicos

- [index.html](#)
- [Dockerfile](#)
- [taskdef.json](#)

index.html

O `index.html` arquivo exibe uma mensagem Hello World no navegador.

Para adicionar o arquivo `index.html`

1. No CodeCatalyst console, acesse seu repositório de origem, `codecatalyst-ecs-source-repository`.
2. Em Arquivos, escolha Criar arquivo.
3. Em Nome do arquivo, digite:

```
public-html/index.html
```

Important

Certifique-se de incluir o `public-html/` prefixo para criar uma pasta com o mesmo nome. `index.html` Espera-se que o esteja nessa pasta.

4. Na caixa de texto, insira o seguinte código:

```
<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. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

O `index.html` é adicionado ao seu repositório em uma `public-html` pasta.

Dockerfile

O Dockerfile descreve a imagem base do Docker a ser usada e os comandos do Docker a serem aplicados a ela. Para obter mais informações sobre o Dockerfile, consulte a Referência do [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 `EXPOSE` instrução no Dockerfile diz ao Docker que o contêiner está escutando na porta 80.

Para adicionar o Dockerfile

1. No seu repositório de origem, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

Dockerfile

Não inclua uma extensão de arquivo.

Important

O Dockerfile deve residir na pasta raiz do seu repositório. O `Docker build` comando do fluxo de trabalho espera que ele esteja lá.

3. Na caixa de texto, insira o seguinte código:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

O Dockerfile é adicionado ao seu repositório.

taskdef.json

O `taskdef.json` arquivo que você adiciona nessa etapa é o mesmo que você já especificou, [Etapa 2: implantar um aplicativo de espaço reservado no Amazon ECS](#) com a seguinte diferença:

Em vez de especificar um nome de imagem do Docker codificado no `image:` campo (`httpd:2.4`), a definição da tarefa aqui usa algumas variáveis para denotar a imagem: e. `$REPOSITORY_URI` `$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 obter detalhes sobre os parâmetros de definição de tarefas, consulte [Parâmetros de definição de tarefas](#) no Amazon Elastic Container Service Developer Guide.

Para adicionar o arquivo `taskdef.json`

1. No seu repositório de origem, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

```
taskdef.json
```

3. Na caixa de texto, insira o seguinte código:

```
{
  "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/ -role codecatalyst-ecs-task-execution`

com o ARN da função de execução da tarefa em que você anotou. [Para criar a função de execução da tarefa](#)

4. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

O `taskdef.json` arquivo é adicionado ao seu 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, em seguida, implanta a imagem no seu cluster do Amazon ECS. Essa implantação substitui o aplicativo de espaço reservado Apache existente.

O fluxo de trabalho consiste nos seguintes blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com 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 `image` campo correto e, em seguida, cria um artefato de saída desse arquivo. Esse artefato é usado como entrada para a ação de implantação, que será a seguir.

Para obter mais informações sobre a ação de criação, consulte [Construindo 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 construção (TaskDefArtifact), encontra o `taskdef.json` interior dele e o registra no seu serviço Amazon ECS. Em seguida, o serviço segue as instruções no `taskdef.json` arquivo para executar três tarefas do Amazon ECS — e os contêineres Hello World Docker 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. Escolha Criar fluxo de trabalho.
3. Em Repositório de origem, escolha `codecatalyst-ecs-source-repository`.
4. Para Branch, escolha `main`.
5. Selecione Create (Criar).
6. Exclua o código de amostra YAML.
7. Adicione o seguinte código YAML:

```
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 ter 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 propriedade `Value:`) com o URI do repositório Amazon ECR em que você criou [Etapa 3: Criar um repositório de imagens do Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (no comando `Run: aws ecr`) 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 de *us-west-2* com AWS seu código de região. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no Referência geral da AWS.

Note

Se você decidiu não criar e implantar funções, `codecatalyst-ecs-deploy-roles` substitua `codecatalyst-ecs-build-roles` 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 `sed` comandos `find` e 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 `Render Amazon ECS` para essa finalidade. Para ter mais informações, consulte [Modificar um arquivo de definição de tarefas do Amazon ECS usando um fluxo de trabalho](#).

8. (Opcional) Escolha Validar para garantir que o código YAML seja válido antes da confirmação.
9. Escolha Commit (Confirmar).
10. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:
 - a. Em Confirmar mensagem, remova o texto e digite:

Add first workflow

- b. Para Repositório, escolha `codecatalyst-ecs-source-repository`.
- c. Em Nome da filial, escolha principal.
- d. Escolha Commit (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 `workflow.yaml` arquivo ao seu repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para ver o progresso 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 construção.
4. Escolha o DeployToECS para ver o progresso da implantação.

Para obter mais informações sobre a visualização dos detalhes da execução, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

Para 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 seu aplicativo com sucesso.

Etapa 9: faça uma alteração nos arquivos de origem

Nesta seção, você faz uma alteração no `index.html` arquivo no seu repositório de origem. Essa alteração faz com que o fluxo de trabalho crie uma nova imagem do Docker, marque-a com um ID de confirmação, envie-a para o Amazon ECR e a implemente 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. Selecione a opção Editar.
4. Na linha 14, altere o `Hello World` texto para `Tutorial complete!`.

- Escolha Confirmar e, em seguida, escolha Confirmar novamente.

A confirmação faz com que uma nova execução do fluxo de trabalho seja iniciada.

- (Opcional) Acesse a página principal do repositório de origem, escolha Exibir confirmações e anote a ID da confirmação da `index.html` alteração.
- Assista ao progresso da implantação:
 - No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
 - Escolha `codecatalyst-ecs-workflow` ver a última execução.
 - Escolha BuildBackende `DeployToECS` para ver o progresso da execução do fluxo de trabalho.
- Verifique se seu aplicativo foi atualizado, da seguinte forma:
 - Abra o console clássico do Amazon ECS em <https://console.aws.amazon.com/ecs/>.
 - Escolha seu cluster, `codecatalyst-ecs-cluster`.
 - Escolha a guia Tasks (Tarefas).
 - Escolha qualquer uma das três tarefas.
 - No campo IP público, escolha endereço aberto.

Uma `Tutorial complete!` página é exibida.

- (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 AWS Management Console, limpe nesta ordem:

- No Amazon ECS, faça o seguinte:
 - Excluir `codecatalyst-ecs-service`.
 - Excluir `codecatalyst-ecs-cluster`.
 - Cancelar registro de `codecatalyst-ecs-task-definition`.
- No Amazon ECR, `codecatalyst-ecs-image-repo` exclua.
- No 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. Excluircodecatalyst-ecs-workflow.
2. Excluircodecatalyst-ecs-environment.
3. Excluircodecatalyst-ecs-source-repository.
4. Excluircodecatalyst-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.

Adicionando a ação “Implantar no Amazon ECS”

Use as instruções a seguir para adicionar a ação Deploy to 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy to Amazon ECS e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Implantar no Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da ação “Implantar no Amazon ECS”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy to Amazon ECS e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Implantar no Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da ação “Implantar no Amazon ECS”](#).
11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Variáveis produzidas pela ação “Deploy to Amazon ECS”

A ação Deploy to Amazon ECS produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

Chave	Valor
cluster	O nome do cluster Amazon ECS no qual foi implantado durante a execução do fluxo de trabalho. Exemplo: <code>codecatalyst-ecs-cluster</code>
plataforma de implantação	O nome da plataforma de implantação. Codificado para. <code>AWS : ECS</code>
serviço	O nome do serviço Amazon ECS que foi implantado durante a execução do fluxo de trabalho.

Chave	Valor
	Exemplo: <code>codecatalyst-ecs-service</code>
<code>task-definition-arn</code>	<p>O Amazon Resource Name (ARN) da definição da tarefa que foi registrada durante a execução do fluxo de trabalho.</p> <p>Exemplo: <code>arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</code></p> <p>O <code>:8</code> exemplo anterior indica a revisão que foi registrada.</p>
<code>url de implantação</code>	<p>Um link para a guia Eventos do console Amazon ECS, onde você pode ver detalhes da implantação do Amazon ECS associada à execução do fluxo de trabalho.</p> <p>Exemplo: <code>https://console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/codecatalyst-ecs-cluster/services/codecatalyst-ecs-service/events</code></p>
<code>região</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>

Definição YAML da ação “Implantar no Amazon ECS”

A seguir está a definição YAML da ação Deploy to Amazon ECS. Para saber como usar essa ação, consulte [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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: DeployToECS
  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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: ECSDeployAction_nn.

UI correspondente: guia de 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 ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: aws/ecs-deploy@v1.

UI correspondente: diagrama de fluxo de trabalho/ECS _nn/aws/ecs-deploy @v1 label DeployAction

DependsOn

(*ECSDeployAction*/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*ECSDeployAction*/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(*ECSDeployAction*/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)
Otimizado para flexibilidade durante as corridas 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: guia Configuração/Avançado - opcional/Tipo de computação

Fleet

(*ECSDeployAction*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se `Fleet` for omitido, o padrão será `Linux.x86-64.Large`

UI correspondente: guia Configuração/Avançado - 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Environment

(*ECSDeployAction*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(*ECSDeployAction*/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(*ECSDeployAction*/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Name

(*ECSDeployAction*/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Role

(*ECSDeployAction*/Environment/Connections/Role)

(Obrigatório)

Especifique o nome da função do IAM que a ação Deploy to Amazon ECS usa para acessar AWS. Certifique-se de que essa função inclua as seguintes políticas:

- A seguinte política de permissões:

⚠ Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "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 a função for usada, use o caractere curinga a seguir na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A seguinte política de confiança personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
```

Certifique-se de que essa função seja adicionada à conexão da sua conta. Para saber mais sobre como adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a CodeCatalystWorkflowDevelopmentRole-*spaceName* função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Função

Inputs

(*ECSDeployAction*/Inputs)

(Optional)

A Inputs seção define os dados *ECSDeployAction* necessários durante a execução de um fluxo de trabalho.

Note

Somente uma entrada (uma fonte ou um artefato) é permitida por ação do Deploy to Amazon ECS.

UI 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 desse repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seu arquivo de definição de tarefa 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 contido em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se o arquivo de definição de tarefas não estiver contido em um artefato, ele deverá residir no seu repositório de origem.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Configuration

(ECSDeployAction/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI correspondente: guia Configuração

region

(Configuration/region)

(Obrigatório)

Especifique a AWS região em que seu cluster e serviço do Amazon ECS residem. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no Referência geral da AWS.

UI correspondente: guia de configuração/região

cluster

(ECSDeployAction/Configuration/cluster)

(Obrigatório)

Especifique o nome de um cluster Amazon ECS existente. A ação Deploy to Amazon ECS implantará seu aplicativo em contêineres como uma tarefa nesse cluster. Para obter mais informações sobre clusters do Amazon ECS, consulte [Clusters](#) no Amazon Elastic Container Service Developer Guide.

UI correspondente: guia de configuração/cluster

service

(ECSDeployAction/Configuration/service)

(Obrigatório)

Especifique o nome de um serviço Amazon ECS existente que instanciará o arquivo de definição de tarefas. Esse serviço deve residir no cluster especificado no `cluster` campo. Para obter mais

informações sobre os serviços do Amazon ECS, consulte os serviços do [Amazon ECS](#) no Amazon Elastic Container Service Developer Guide.

UI correspondente: guia de configuração/serviço

task-definition

(ECSDeployAction/Configuration/task-definition)

(Obrigatório)

Especifique o caminho para um arquivo de definição de tarefa existente. Se o arquivo residir no 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 obter mais informações sobre arquivos de definição de tarefas, consulte [Definições de tarefas](#) no Amazon Elastic Container Service Developer Guide.

UI correspondente: guia de configuração/definição de tarefa

force-new-deployment

(ECSDeployAction/Configuration/force-new-deployment)

(Obrigatório)

Se habilitado, o serviço 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 obter mais informações sobre forçar novas implantações, consulte [Atualização de um serviço](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Padrão: false

UI correspondente: guia de configuração/ Forçar uma nova implantação do serviço

codedeploy-appspec

(ECSDeployAction/Configuration/codedeploy-appspec)

(Obrigatório se você configurou seu serviço Amazon ECS para usar implantações azul/verde, 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 implantações azul/verdes. Para implantações de atualizações contínuas (o padrão), CodeDeploy omite as informações. Para obter mais informações sobre as implantações do Amazon ECS, consulte os [tipos de implantação do Amazon ECS](#) no Amazon Elastic Container Service Developer Guide.

Note

Os CodeDeploy campos podem estar ocultos no editor visual. Para que eles apareçam, consulte [Por que os CodeDeploy campos estão ausentes no editor visual?](#)

UI correspondente: guia Configuração/ CodeDeploy AppSpec

codedeploy-application

(*ECSDeployAction*/Configuration/codedeploy-application)

(Obrigatório se codedeploy-appspec estiver 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 estiver 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

Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho

Tip

Para ver um tutorial que mostra como usar a ação de cluster Implantar no Kubernetes, consulte. [Tutorial: Implantar um aplicativo 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 de cluster Implantar no Kubernetes ao seu fluxo de trabalho. Essa ação implanta seu aplicativo em um cluster 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 um aplicativo no Amazon EKS](#).

[Para obter mais informações sobre o Kubernetes, consulte a documentação do Kubernetes.](#)

Para obter mais informações sobre o Amazon EKS, consulte [O que é o Amazon EKS?](#) no Guia do usuário do Amazon EKS.

Como funciona a ação “Implantar no cluster Kubernetes”

O cluster Deploy to 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 é configurada `kubectl` para

apontar para o cluster Amazon EKS que você forneceu ao configurar a ação. O `kubectl` utilitário é necessário para executar o `kubectl apply` comando a seguir.

2. A ação executa o `kubectl apply -f my-manifest.yaml` comando, que executa as instruções em `my-manifest.yaml` para implantar seu aplicativo como um conjunto de contêineres e pods no cluster configurado. Para obter mais informações sobre esse comando, consulte o tópico [kubectl apply](#) na documentação de referência do Kubernetes.

Tópicos

- [Tutorial: Implantar um aplicativo no Amazon EKS](#)
- [Adicionando a ação “Implantar no cluster Kubernetes”](#)
- [Variáveis produzidas pela ação “Implantar no cluster Kubernetes”](#)
- [Definição YAML da ação “Implantar no cluster Kubernetes”](#)

Tutorial: Implantar um aplicativo 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 O aplicativo implantado é um simples “Hello, World!” site construído em uma imagem Docker do servidor web Apache. O tutorial mostra o trabalho de preparação necessário, como configurar uma máquina de desenvolvimento e um cluster Amazon EKS, e depois descreve como criar um fluxo de trabalho para criar o aplicativo e implantá-lo no cluster.

Depois que a implantação inicial for concluída, o tutorial instruirá você a fazer uma alteração na fonte do aplicativo. Essa alteração faz com que uma nova imagem do Docker seja criada e enviada ao seu repositório de imagens do Docker com as novas informações de revisão. A nova revisão da imagem do Docker é então implantada no Amazon EKS.

Tip

Em vez de seguir este tutorial, você pode usar um blueprint que faz uma configuração completa do Amazon EKS para você. Você precisará usar o plano EKS App Deployment. Para ter mais informações, consulte [Criando um projeto com um blueprint](#).

Tópicos

- [Pré-requisitos](#)

- [Etapa 1: configurar sua máquina de desenvolvimento](#)
- [Etapa 2: criar um cluster Amazon EKS](#)
- [Etapa 3: Criar um repositório de imagens 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: faça 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 ter mais informações, consulte [Criando um espaço](#).
- Em seu espaço, você precisa de um CodeCatalyst projeto vazio, Start from scratch, chamado:

```
codecatalyst-eks-project
```

Para ter 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 ter 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 ter mais informações, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

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 `eksctl` utilitário — para criação de clusters
- a `kubectl` utilidade — 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, além de ser 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.
- [Instalando ou atualizando a versão mais recente do AWS CLI](#) no Guia AWS Command Line Interface do Usuário.

Para iniciar um ambiente de desenvolvimento

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

2. Navegue até seu projeto, `codecatalyst-eks-project`.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha o nome do seu repositório de origem, `codecatalyst-eks-source-repository`.
5. Na parte superior, escolha Create Dev Environment e, em seguida, escolha AWS Cloud9 (no navegador).
6. Certifique-se de que Trabalhar na ramificação e principal existentes estejam selecionadas e, em seguida, escolha Criar.

Seu 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 Dev Environment, digite:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Digite:

```
chmod +x ./kubectl
```

3. Digite:

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

4. Digite:

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

5. Digite:

```
kubectl version --short --client
```

6. Verifique se uma versão aparece.

Agora você instalou `kubectl`.

Para instalar e configurar o eksctl

Note

eksctl não é estritamente necessário porque você pode usar `kubectl` em vez disso. No entanto, `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 Dev Environment, digite:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

2. Digite:

```
sudo cp /tmp/eksctl /usr/bin
```

3. Digite:

```
eksctl version
```

4. Verifique se uma versão aparece.

Agora você instalou `eksctl`.

Para verificar se o AWS CLI está instalado

1. No terminal do Dev Environment, digite:

```
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 [Configurando o AWS CLI](#) no Guia do AWS Command Line Interface Usuário.

1. Crie um usuário do IAM Identity Center, da seguinte forma:
 - a. Faça login no AWS Management Console e abra o AWS IAM Identity Center console em <https://console.aws.amazon.com/singlesignon/>.

(Talvez seja necessário escolher Ativar se 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 ter mais informações, consulte [Criando 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, escolha 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 Primeiro nome, digite:

`codecatalyst-eks-user`

- g. Em Sobrenome, digite:

`codecatalyst-eks-user`

- h. Em Nome de exibição, mantenha:


```
codecatalyst-eks-user codecatalyst-eks-user
```

- i. Selecione Next (Próximo).
- j. Na página Adicionar usuário aos grupos, escolha Avançar.
- k. Na página Revisar e adicionar usuário, revise as informações e escolha Adicionar usuário.

Uma caixa de diálogo de senha de uso único é exibida.

- l. Escolha 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, em seguida, escolha Criar conjunto de permissões.
 - b. Escolha Conjunto de permissões predefinido e, em seguida, selecione AdministratorAccess. Essa política fornece permissões completas para todos Serviços da AWS.
 - c. Selecione Next (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 paracodecatalyst-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 `codecatalyst-eks-user` de acesso 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 `codecatalyst-eks-user` detalhes no IAM Identity Center, escolha Redefinir senha, Gerar uma senha de uso único [...] e redefina a 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 escolha Definir nova senha.

Uma caixa de Conta da AWS aparece na tela.

- f. Escolha e Conta da AWS, em seguida, escolha o nome do Conta da AWS ao qual você atribuiu o `codecatalyst-eks-user` usuário e o conjunto de permissões.
- g. Ao lado de `codecatalyst-eks-permission-set`, escolha Linha de comando ou acesso programático.
- h. Copie os comandos no meio da página. Eles são parecidos com os seguintes:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
```

```
export AWS_SECRET_ACCESS_KEY="wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... onde o *token de sessão* é uma longa sequência aleatória.

5. Adicione as chaves de acesso e o token de sessão ao AWS CLI, da seguinte forma:
 - a. Retorne ao seu ambiente de CodeCatalyst desenvolvimento.
 - b. 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 ver 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 `aws configure` comando. 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 Amazon EKS

Nesta seção, você cria um cluster no Amazon EKS. As instruções abaixo descrevem uma maneira rápida de criar o cluster usando `eksctl`, mas se você quiser instruções detalhadas, consulte:

- [Introdução ao eksctl](#) no Guia do usuário do Amazon EKS

ou

- [Introdução ao console e AWS CLI ao](#) Guia do usuário do Amazon EKS (este tópico fornece `kubectl` instruções para criar o cluster)

Note

Os [clusters privados](#) não são compatíveis com a CodeCatalyst integração com o Amazon EKS.

Antes de começar

Certifique-se de ter concluído as seguintes tarefas em sua máquina de desenvolvimento:

- Instalou o `eksctl` utilitário.
- Instalou o `kubectl` utilitário.
- Instalei AWS CLI e configurei com chaves de acesso e um token de sessão.

Para obter 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 `eksctl` utilitário, conforme descrito nas etapas a seguir.

1. Acesse seu 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 por 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 `waiting for CloudFormation stack` mensagens ao AWS criar seu cluster. Isso é esperado.

3. Verifique se seu cluster foi criado com sucesso:

```
kubectl cluster-info
```

Você verá uma mensagem semelhante à seguinte, indicando uma criação bem-sucedida do cluster:

```
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 um repositório de imagens do Amazon ECR

Nesta seção, você cria um repositório privado de imagens no Amazon Elastic Container Registry (Amazon ECR). Esse repositório armazena a imagem do Docker para o 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 seu 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 nomeou seu repositório `codecatalyst-eks-image-repo`.

3. Exiba os detalhes do repositório Amazon ECR:

```
aws ecr describe-repositories \
  --repository-names codecatalyst-eks-image-repo
```

4. Observe o “repositoryUri”: valor, 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 do aplicativo ao seu repositório de origem (`codecatalyst-eks-source-repository`). Eles consistem em:

- Um `index.html` arquivo — Exibe uma mensagem “Hello, World!” mensagem no navegador.
- Um `Dockerfile` — descreve a imagem base a ser usada para sua imagem do Docker e os comandos do Docker a serem aplicados a ela.
- Um `deployment.yaml` arquivo — O manifesto do Kubernetes que define o serviço e a implantação do Kubernetes.

A estrutura de pastas é a seguinte:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
    | |– index.html
  |– Dockerfile
```

Tópicos

- [index.html](#)
- [Dockerfile](#)
- [deployment.yaml](#)

index.html

O `index.html` arquivo exibe uma mensagem “Hello, World!” mensagem no navegador.

Para adicionar o arquivo index.html

1. Acesse seu 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 seguinte conteúdo:

```
<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, digite:

```
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 `public-html` pasta.

Dockerfile

O Dockerfile descreve a imagem base do Docker a ser usada e os comandos do Docker a serem aplicados a ela. Para obter mais informações sobre o Dockerfile, consulte a Referência do [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 `EXPOSE` instrução no Dockerfile diz ao Docker que o contêiner está escutando na porta 80.

Para adicionar o Dockerfile

1. Em `emcodecatalyst-eks-source-repository`, crie um arquivo chamado `Dockerfile` com o seguinte conteúdo:

```
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 seu repositório. O `Docker build` comando 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 seu repositório.

`deployment.yaml`

Nesta seção, você adiciona um `deployment.yaml` arquivo ao seu repositório. O `deployment.yaml` arquivo é um manifesto do Kubernetes que define dois tipos ou tipos de recursos do Kubernetes a serem executados: um 'serviço' e uma 'implantação'.

- O 'serviço' implanta um balanceador de carga no 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 o link "Hello, World!" aplicativo.

- A 'implantação' implanta três pods, e cada pod conterá um contêiner Docker com a mensagem '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 `emcodecatalyst-eks-source-repository`, crie uma pasta chamada `Kubernetes`.
2. Em `Kubernetes`, crie um arquivo chamado `deployment.yaml` com o seguinte conteúdo:

```
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 `deployment.yaml` arquivo 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 você tenha colocado todos os arquivos nas pastas corretas. A estrutura de pastas é a seguinte:

```
|– 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. Essas funções são:

- Função de criação — concede permissão à ação de CodeCatalyst criação (no fluxo de trabalho) para acessar sua AWS conta e gravar no Amazon ECR e no 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 obter mais informações sobre as funções do IAM, consulte [Funções do IAM](#) no Guia AWS Identity and Access Management do usuário.

Note

Para economizar tempo, você pode criar uma única função, chamada de `CodeCatalystWorkflowDevelopmentRole-spaceName` função, em vez das duas funções listadas anteriormente. Para ter mais informações, consulte [Criando a `CodeCatalystWorkflowDevelopmentRole-spaceName` função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando as duas funções listadas anteriormente.

Para criar as funções de criação e implantação, conclua a série de procedimentos a seguir.

1. Para criar uma política de confiança para ambas as funções
 1. Acesse seu ambiente de desenvolvimento.
 2. No Cloud9-*long-string* diretório, crie um arquivo chamado `codecatalyst-eks-trust-policy.json` com o seguinte conteúdo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Para criar a política de criação para a função de construção

- No Cloud9-*long-string* diretório, crie um arquivo chamado `codecatalyst-eks-build-policy.json` com o seguinte conteúdo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

3. Para criar a política de implantação para a função de implantação

- No Cloud9-*long-string* diretório, crie um arquivo chamado `codecatalyst-eks-deploy-policy.json` com o seguinte conteúdo:

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

```

```

        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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. Sua estrutura de diretórios agora tem a seguinte aparência:

```

|- 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 Dev Environment, digite:

```
cd /projects
```

2. Digite:

```
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 "arn": valor, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Você precisará desse ARN mais tarde.

5. Para adicionar a política de implantação ao AWS

1. Digite:

```
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 "arn": valor da política de implantação, por exemplo, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy`. Você precisará desse ARN mais tarde.

6. Para criar a função de criação

1. Digite:

```
aws iam create-role \  
  --role-name codecatalyst-eks-build-role \  
  --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Pressione Enter.

3. Digite:

```
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/* é substituído pelo ARN da política *codecatalyst-eks-build-policy* de compilação que você observou anteriormente.

4. Pressione Enter.
5. No prompt do terminal, digite:

```
aws iam get-role \  
    --role-name codecatalyst-eks-build-role
```

6. Pressione Enter.
7. Observe o "Arn": valor da função, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-eks-build-role`. Você precisará desse ARN mais tarde.

7. Para criar a função de implantação

1. Digite:

```
aws iam create-role \  
    --role-name codecatalyst-eks-deploy-role \  
    --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Pressione Enter.

3. Digite:

```
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/* é substituído pelo ARN da *política codecatalyst-eks-deploy-policy* de implantação que você observou anteriormente.

4. Pressione Enter.

5. Digite:

```
aws iam get-role \  
    --role-name codecatalyst-eks-deploy-role
```

6. Pressione Enter.

7. Observe o "Arn": valor da função, por exemplo, `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Você precisará desse ARN mais tarde.

Agora você criou funções de criação e implantação e anotou seus 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 as funções 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, escolha Configurações.
3. No painel de navegação, escolha AWS contas. 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.) Você precisará desse valor 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 acessar a página.

7. Selecione Adicionar uma função existente que você criou no IAM.

Uma lista suspensa é exibida. A lista exibe as funções de criação e implantação e quaisquer outras funções do IAM com uma política de confiança que inclua os diretores de `codecatalyst.amazonaws.com` serviço `codecatalyst-runner.amazonaws.com` e os principais.

8. Na lista suspensa, adicione:
 - `codecatalyst-eks-build-role`
 - `codecatalyst-eks-deploy-role`

Note

Se você ver `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.

As funções de criação e implantação agora devem aparecer em funções do IAM.

Essas funções agora estão disponíveis para uso em CodeCatalyst fluxos de trabalho.

Etapa 7: atualizar o ConfigMap

Você deve adicionar a função de implantação que você criou [Etapa 5: criar AWS funções](#) ao ConfigMap arquivo Kubernetes para dar à ação de cluster Implantar no Kubernetes (em seu fluxo de trabalho) a capacidade de acessar e interagir com seu cluster. Você pode usar `eksctl` ou `kubectl` para executar essa tarefa.

Para configurar o arquivo Kubernetes ConfigMap usando `eksctl`

- No terminal do Dev Environment, digite:

```
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 por sua região.

Para obter detalhes sobre esse comando, consulte [Gerenciar usuários e funções 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 Dev Environment, digite:

```
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 obter detalhes, consulte [Habilitar o acesso principal do IAM ao seu cluster](#) no Guia do usuário do Amazon EKS.

Agora você concedeu à função de implantação e, por extensão, à ação Implantar no Amazon EKS, `system:masters` permissões para seu cluster Kubernetes.

Etapa 8: criar e executar um fluxo de trabalho

Nesta etapa, você cria um fluxo de trabalho que pega seus arquivos de origem, os transforma em uma imagem do Docker e, em seguida, implanta a imagem em pods de árvores no seu cluster Amazon EKS.

O fluxo de trabalho consiste nos seguintes blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com 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 `$IMAGE_TAG` as variáveis `$REPOSITORY_URI` e no `deployment.yaml` arquivo com os valores corretos e, em seguida, cria um artefato de saída desse arquivo e de quaisquer outros na Kubernetes pasta. Neste tutorial, o único arquivo na Kubernetes pasta é `deployment.yaml`, mas você pode incluir mais arquivos. O artefato é usado como entrada para a ação de implantação, que será a próxima.

Para obter mais informações sobre a ação de criação, consulte [Construindo 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 construção (Manifests) e encontra o `deployment.yaml` arquivo dentro dela. Em seguida, a ação segue as instruções no `deployment.yaml` arquivo para executar três pods, cada um contendo um único “Hello, World!” Contêiner Docker — dentro do seu cluster Amazon EKS.

Para criar um fluxo de trabalho

1. Vá para o CodeCatalyst console.
2. Navegue até seu projeto (`codecatalyst-eks-project`).
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha Criar fluxo de trabalho.
5. Em Repositório de origem, escolha `codecatalyst-eks-source-repository`.
6. Para Branch, escolha `main`.
7. Selecione Create (Criar).
8. Exclua o código de amostra YAML.
9. Adicione o código YAML a seguir para criar um novo arquivo de definição de fluxo de trabalho:

Note

Para obter mais informações sobre o arquivo de definição do fluxo de trabalho, consulte [Definição de fluxo de trabalho YAML](#).

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 ter 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 propriedade `Value:`) com o URI do repositório Amazon ECR em que você criou. [Etapa 3: Criar um repositório de imagens do Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (no comando `Run: aws ecr`) 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 de *us-west-2* com AWS seu código de região. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no 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) Escolha Validar para garantir que o código YAML seja válido antes da confirmação.
11. Escolha Commit (Confirmar).
12. Na caixa de diálogo Confirmar fluxo de trabalho, insira o seguinte:

- a. Em Confirmar mensagem, remova o texto e digite:

Add first workflow

- b. Para Repositório, escolha *codecatalyst-eks-source-repository*.
- c. Em Nome da filial, escolha principal.
- d. Escolha Commit (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 `workflow.yaml` arquivo ao seu repositório de origem, o gatilho iniciou a execução do fluxo de trabalho.

Para ver o progresso 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 compilação.
4. Escolha DeployToEKS para ver o progresso da implantação.

Para obter mais informações sobre a visualização dos detalhes da execução, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

Para verificar a implantação

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. À esquerda, na parte inferior, escolha Load Balancers.
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 DNS na barra de endereço do seu navegador.

O 'Hello, World!' A página da web aparece em seu navegador, indicando que você implantou seu aplicativo com sucesso.

Etapa 9: faça uma alteração nos arquivos de origem

Nesta seção, você faz uma alteração no `index.html` arquivo no seu repositório de origem. Essa alteração faz com que o fluxo de trabalho crie uma nova imagem do Docker, marque-a com um ID de confirmação, envie-a para o Amazon ECR e a implemente no Amazon ECS.

Para alterar o `index.html`

1. Acesse seu 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 `Hello, World!` texto para `Tutorial complete!`.
6. Adicione, confirme e envie:

```
git add .  
git commit -m "update index.html title"  
git push
```

A execução do fluxo de trabalho é iniciada automaticamente.

7. (Opcional) Digite:

```
git show HEAD
```

Anote o ID de confirmação da `index.html` alteração. 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 progresso 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` ver a última execução.

- c. Escolha BuildBackende DeployToEKS para ver o progresso da execução do fluxo de trabalho.
9. Verifique se seu aplicativo foi atualizado, da seguinte forma:
- a. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
 - b. À esquerda, na parte inferior, escolha Load Balancers.
 - c. Selecione o balanceador de carga que foi criado como parte da sua implantação do Kubernetes.
 - d. Copie e cole o valor do nome DNS na barra de endereço do seu navegador.
- O 'Tutorial concluído!' A página da web aparece no seu navegador, indicando que você implantou com sucesso uma nova revisão do seu aplicativo.
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 ser cobrado desnecessariamente pelos recursos de armazenamento e computação usados neste tutorial.

Para limpar

1. Exclua seu cluster:
 - No terminal do Dev Environment, digite:

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Em que:

- *us-west-2* é substituído por sua região.
- *codecatalyst-eks-cluster* é substituído pelo nome do cluster que você criou.

Após 5 a 10 minutos, o cluster e os recursos associados são excluídos, incluindo, mas não se limitando a, AWS CloudFormation pilhas, grupos de nós (no 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 AWS credenciais, consulte [Como faço para corrigir os erros “Não é possível localizar credenciais” e ExpiredToken ""?](#) Para atualizar suas `kubectl` credenciais, consulte [Como faço para corrigir os erros “Não é possível conectar-se ao servidor”?](#)

2. No AWS console, limpe da seguinte forma:
 1. No Amazon ECR, `codecatalyst-eks-image-repo` exclua.
 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. Excluir `codecatalyst-eks-workflow`.
 2. Excluir `codecatalyst-eks-environment`.
 3. Excluir `codecatalyst-eks-source-repository`.
 4. Exclua seu ambiente de desenvolvimento.
 5. Excluir `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.

Adicionando a ação “Implantar no cluster Kubernetes”

Use as instruções a seguir para adicionar a ação de cluster Implantar no Kubernetes ao seu fluxo de trabalho.

Antes de começar

Antes de adicionar a ação de cluster Implantar no Kubernetes ao seu fluxo de trabalho, você deve preparar o seguinte:

Tip

Para configurar esses pré-requisitos rapidamente, siga as instruções em [Tutorial: Implantar um aplicativo no Amazon EKS](#)

- Um cluster Kubernetes no Amazon EKS. Para obter informações sobre clusters, consulte [os clusters do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- Pelo menos um Dockerfile que descreva como montar seu aplicativo em uma imagem do Docker. Para obter mais informações sobre Dockerfiles, consulte a referência do [Dockerfile](#).
- Pelo menos um arquivo de manifesto do Kubernetes, chamado de arquivo ou configuração de configuração na documentação do Kubernetes. Para obter mais informações, consulte [Gerenciamento de recursos](#) na documentação do Kubernetes.
- Uma função do IAM que dá à ação de cluster Deploy to Kubernetes a capacidade de acessar e interagir com seu cluster Amazon EKS. Para obter mais informações, consulte o tópico [Role](#) no [Definição YAML da ação “Implantar no cluster Kubernetes”](#).

Depois de criar essa função, você deve adicioná-la 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 funções 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 [Criando um espaço](#), [Criando um projeto vazio na Amazon CodeCatalyst](#) e [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

- Um repositório de origem suportado pelo CodeCatalyst. O repositório armazena os arquivos de origem do aplicativo, Dockerfiles e manifestos do Kubernetes. Para ter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).

Visual

Para adicionar a ação “Implantar no cluster 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de cluster Deploy to Kubernetes e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Implantar no cluster Kubernetes. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da ação “Implantar no cluster Kubernetes”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar a ação “Implantar no cluster 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de cluster Deploy to Kubernetes e faça o seguinte:

- Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Implantar no cluster Kubernetes. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da ação “Implantar no cluster Kubernetes”](#).
 11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

- Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Variáveis produzidas pela ação “Implantar no cluster Kubernetes”

A ação de cluster Deploy to Kubernetes produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

Chave	Valor
cluster	O nome do recurso da Amazon.com (ARN) do cluster Amazon EKS que foi implantado durante a execução do fluxo de trabalho. Exemplo: <code>arn:aws:eks:us-west-2:111122223333:cluster/codecatalyst-eks-cluster</code>
plataforma de implantação	O nome da plataforma de implantação. Codificado para. <code>AWS:EKS</code>
metadata	Reservado. Metadados formatados em JSON relacionados ao cluster implantado durante a execução do fluxo de trabalho.
namespace	O namespace Kubernetes no qual o cluster foi implantado. Exemplo: <code>default</code>
recursos	Reservado. Metadados formatados em JSON relacionados aos recursos implantados durante a execução do fluxo de trabalho.
servidor	O nome do endpoint do servidor de API que você pode usar para se comunicar com seu

Chave	Valor
	<p>cluster usando ferramentas de gerenciamento, como <code>kubectl</code>.</p> <p>Para obter mais informações sobre o endpoint do serviço de API, consulte Controle de acesso ao endpoint do cluster Amazon EKS no Guia do usuário do Amazon EKS.</p> <p>Exemplo: <code>https://random-st ring.gr7.us-west-2.eks.amazonaws.com</code></p>

Definição YAML da ação “Implantar no cluster Kubernetes”

Veja a seguir a definição YAML da ação de cluster Deploy to Kubernetes. Para saber como usar essa ação, consulte [Implantação de um aplicativo no Amazon Elastic Kubernetes Service 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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@v1DependsOn:

- *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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: DeployToKubernetesCluster_nn.

UI correspondente: guia de 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 ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: `aws/kubernetes-deploy@v1`.

UI correspondente: diagrama de fluxo de trabalho/ DeployToKubernetesCluster _nn/ aws/kubernetes-deploy @v1 label

DependsOn

(DeployToKubernetesCluster/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(DeployToKubernetesCluster/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(DeployToKubernetesCluster/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia Configuração/Avançado - opcional/Tipo de computação

Fleet

(DeployToKubernetesCluster/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`

UI correspondente: guia Configuração/Avançado - 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Environment

(DeployToKubernetesCluster/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(DeployToKubernetesCluster/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(DeployToKubernetesCluster/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Name

(DeployToKubernetesCluster/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta


Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Obrigatório)


Especifique o nome da função do IAM que a ação de cluster Deploy to Kubernetes usa para acessar. AWS Certifique-se de que essa função inclua as seguintes políticas:

- A seguinte política de permissões:

 Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

Na primeira vez em que a função for usada, use o caractere curinga a seguir na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Certifique-se de que essa função seja adicionada a:

- Conexão da sua conta. Para saber mais sobre como adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções 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 um aplicativo no Amazon EKS](#) para obter instruções sobre como adicionar uma função do IAM a uma conexão de conta ConfigMap e.

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante. Se você decidir usar a `CodeCatalystWorkflowDevelopmentRole-spaceName` função, certifique-se de adicioná-la ao seu ConfigMap arquivo, seguindo as instruções em [Gerenciar usuários e funções do IAM](#) na eksctl documentação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Função

Inputs

(DeployToKubernetesCluster/Inputs)

(Optional)

A Inputs seção define os dados `DeployToKubernetesCluster` necessários durante a execução de um fluxo de trabalho.

Note

Somente uma entrada (uma fonte ou um artefato) é permitida por ação Deploy to Amazon EKS.

UI correspondente: guia Entradas

Sources

(DeployToKubernetesCluster/Inputs/Sources)

(Obrigatório se o arquivo de manifesto estiver armazenado em um repositório de origem)

Se seu arquivo ou arquivos de manifesto do Kubernetes estiverem armazenados em um repositório de origem, especifique o rótulo desse repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seus arquivos de manifesto não estiverem contidos em um repositório de origem, eles deverão residir em um artefato gerado por outra ação.

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - opcional

Artifacts - input

(DeployToKubernetesCluster/Inputs/Artifacts)

(Obrigatório se o arquivo de manifesto estiver armazenado em um [artefato de saída](#) de uma ação anterior)

Se o arquivo ou arquivos de manifesto do Kubernetes estiverem contidos em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se seus arquivos de manifesto não estiverem contidos em um artefato, eles deverão residir no seu repositório de origem.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Configuration

(DeployToKubernetesCluster/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI correspondente: guia Configuração

Namespace

(DeployToKubernetesCluster/Configuration/namespace)

(Optional)

Especifique o namespace Kubernetes no qual seu aplicativo Kubernetes será implantado. Use `default` se você não estiver usando namespaces com seu cluster. Para obter mais informações sobre namespaces, consulte [Subdividir seu cluster usando namespaces do Kubernetes na documentação do Kubernetes](#).

Se você omitir o namespace, um valor de `default` será usado.

UI correspondente: guia de configuração/namespace

Region

(DeployToKubernetesCluster/Configuration/Region)

(Obrigatório)

Especifique a AWS região em que seu cluster e serviço do Amazon EKS residem. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no Referência geral da AWS.

UI correspondente: guia de configuração/região

Cluster

(DeployToKubernetesCluster/Configuration/Cluster)

(Obrigatório)

Especifique o nome de um cluster Amazon EKS existente. A ação de cluster Deploy to Kubernetes implantará seu aplicativo em contêineres nesse cluster. Para obter mais informações sobre os clusters do Amazon EKS, consulte [Clusters](#) no Guia do usuário do Amazon EKS.

UI correspondente: guia de 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, 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, certifique-se de prefixar os nomes dos arquivos com números ou letras crescentes para controlar a ordem de processamento. Por exemplo: .

```
00-namespace.yaml
```

```
01-deployment.yaml
```

Se seus arquivos de manifesto residirem no 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 será relativo à pasta raiz do artefato.


Exemplos:

```
Manifests/
```

```
deployment.yaml
```

```
my-deployment.yml
```

Não use curingas (*).

 Note

[Os gráficos do Helm](#) e os [arquivos de personalização não são suportados](#).

Para obter mais informações sobre arquivos de manifesto, consulte [Organização das configurações de recursos na documentação](#) do Kubernetes.

UI correspondente: guia de configuração/manifestos

Implantando uma AWS CloudFormation pilha com um fluxo de trabalho

Esta seção descreve como implantar uma AWS CloudFormation pilha usando um CodeCatalyst fluxo de trabalho. Para fazer isso, você deve adicionar a ação Deploy AWS 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:

- AWS CloudFormation modelo — Para obter mais informações, consulte Como [trabalhar com AWS 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 um aplicativo sem servidor usando AWS CloudFormation](#).

Se a pilha já existir, a ação executará a CloudFormation [CreateChangeSet](#) operação e, em seguida, a [ExecuteChangeSet](#) operação. A ação então espera que as alterações sejam implantadas e se marca como bem-sucedida ou falhada, dependendo dos resultados.

Use a ação Deploy AWS CloudFormation stack se você já tiver um AWS SAM modelo AWS 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 AWS CloudFormation stack.

Tip

Para ver um tutorial que mostra como implantar um aplicativo sem servidor usando a ação Deploy AWS CloudFormation stack, consulte. [Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation](#)

Tópicos

- [Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation](#)
- [Adicionando a ação “Deploy AWS CloudFormation stack”](#)
- [Configurando reversões](#)
- [Variáveis produzidas pela ação “Deploy AWS CloudFormation stack”](#)
- [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#)

Tutorial: implantar um aplicativo sem servidor usando AWS CloudFormation

Neste tutorial, você aprende a criar, testar e implantar um aplicativo sem servidor como uma CloudFormation pilha usando um fluxo de trabalho.

O aplicativo neste tutorial é um aplicativo 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 [AWS 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: faça uma alteração](#)
- [Limpeza](#)

Pré-requisitos

Antes de começar

- Você precisa de um CodeCatalyst espaço com uma AWS conta conectada. Para ter mais informações, consulte [Criando um espaço](#).
- Em seu espaço, você precisa de um CodeCatalyst projeto vazio, Start from scratch, chamado:

```
codecatalyst-cfn-project
```

Para ter 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 ter mais informações, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).

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 Lambda.

Para obter mais informações sobre repositórios de origem, consulte [Criando 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, digite:

```
codecatalyst-cfn-source-repository
```

4. Selecione Create (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 AWS CloudFormation stack para acessar sua AWS conta e CloudFormation serviço onde você implantará seu aplicativo sem servidor. A ação Deploy AWS 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 obter mais informações sobre as funções do IAM, consulte [Funções do IAM](#) no Guia AWS Identity and Access Management do usuário.

Note

Para economizar tempo, você pode criar uma única função, chamada de `CodeCatalystWorkflowDevelopmentRole-spaceName` função, em vez das três funções listadas anteriormente. Para ter mais informações, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco à segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante. Este tutorial pressupõe que você esteja criando as três funções listadas anteriormente.

Note

Uma [função de execução do Lambda](#) também é necessária, mas você não precisa criá-la agora porque o `sam-template.yml` arquivo a cria para você quando você executa o fluxo de trabalho na etapa 5.

Para criar uma função de implantação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Criar política.
 - e. Escolha a guia JSON.

- f. Exclua o código existente.
- g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-deploy-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-deploy-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

codecatalyst-deploy-role

- i. Em Descrição da função, insira:

CodeCatalyst deploy role

- j. Selecione Criar função.

Agora você criou uma função de implantação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN da função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (codecatalyst-deploy-role).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de implantação com as permissões apropriadas e obteve seu ARN.

Para criar uma função de criação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Criar política.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
```

```
}]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
}
```

```
        ],
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

codecatalyst-build-role

- i. Em Descrição da função, insira:

CodeCatalyst build role

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN da função de construção, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-build-role`).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de criação com as permissões apropriadas e obteve seu ARN.

Para criar uma função de pilha

1. Faça login AWS usando a conta na qual você deseja implantar sua pilha.
2. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
3. Crie a função de pilha da seguinte forma:
 - a. No painel de navegação, escolha Roles.
 - 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. Na parte inferior, escolha Avançar.
 - g. Usando a caixa de pesquisa, localize as políticas de permissões a seguir e marque suas respectivas caixas de seleção.

Note

Se você pesquisar uma política e ela não aparecer, escolha Limpar filtros e tente novamente.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- IAM FullAccess
- AWS Lambda_ FullAccess
- API da Amazon GatewayAdministrator
- Amazon S3 FullAccess
- Amazon EC2 ContainerRegistryFullAccess

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 obter mais informações, consulte [Permissões](#) no Guia do AWS Serverless Application Model desenvolvedor.

- h. Selecione Next (Próximo).
 - i. Em Nome da função, digite:

`codecatalyst-stack-role`
 - j. Selecione Criar função.
4. Obtenha o ARN da função de pilha, da seguinte forma:
- a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-stack-role`).
 - c. Escolha a função na lista.
 - d. Na seção Resumo, copie o valor do ARN. Você precisará disso mais tarde.

Agora você criou a função de pilha com as permissões apropriadas e obteve seu 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

Você não precisa adicionar a função 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 funções de criação e implantação à conexão da sua conta

1. Dentro CodeCatalyst, navegue até seu espaço.
2. Escolha AWS contas. 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 acessar a página.

5. Selecione Adicionar uma função existente que você criou no IAM.

Uma lista suspensa é exibida. A lista exibe todas as funções do IAM com uma política de confiança que inclui os diretores `codecatalyst-runner.amazonaws.com` e `codecatalyst.amazonaws.com` de serviço.

6. Na lista suspensa `codecatalyst-build-role`, escolha e escolha Adicionar função.
7. Escolha Adicionar função do IAM, escolha Adicionar uma função existente que você criou no IAM e, na lista suspensa, escolha `codecatalyst-deploy-role` Escolha Add role (adicionar função).

Agora você adicionou as funções de criação e implantação ao seu espaço.

8. Copie o valor do nome de CodeCatalyst exibição da Amazon. Você precisará desse valor posteriormente, ao criar seu fluxo de trabalho.

Etapa 4: criar um bucket do Amazon S3

Nesta etapa, você cria um bucket do Amazon S3 onde armazena o arquivo.zip do pacote de implantação do seu aplicativo sem servidor.

Para criar um bucket do Amazon S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel principal, escolha Criar bucket.
3. Em Nome do bucket, insira:

`codecatalyst-cfn-s3-bucket`

4. Em AWS Region (Região da AWS), escolha uma região. Este tutorial pressupõe que você escolheu US West (Oregon) us-west-2. Para obter informações sobre regiões suportadas pelo Amazon S3, consulte [endpoints e cotas do Amazon Simple Storage Service](#) no. Referência geral da AWS
5. Na parte inferior da página, escolha Criar bucket.

Agora você criou um bucket chamado **codecatalyst-cfn-s3-bucket** na região us-west-2 do Oeste dos EUA (Oregon).

Etapa 5: Adicionar arquivos de origem

Nesta etapa, você adiciona vários arquivos de origem do aplicativo ao seu repositório CodeCatalyst de origem. A `hello-world` pasta contém os arquivos do aplicativo que você implantará. A `tests` pasta contém testes de unidade. A estrutura de pastas é a seguinte:

```
.
|- hello-world
|  |- tests
|     |- unit
|         |- test-handler.js
|  |- app.js
|- .npmignore
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

arquivo.npmignore

O `.npmignore` arquivo indica quais arquivos e pastas o npm deve excluir do pacote do aplicativo. Neste tutorial, o npm exclui a `tests` pasta porque ela não faz parte do aplicativo.

Para adicionar o arquivo.npmignore

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Escolha seu projeto, `codecatalyst-cfn-project`
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Na lista de repositórios de origem, escolha seu repositório, `codecatalyst-cfn-source-repository`
5. Em Arquivos, escolha Criar arquivo.
6. Em Nome do arquivo, digite:

```
.npmignore
```

7. Na caixa de texto, insira o seguinte código:

```
tests/*
```

- Escolha Confirmar e, em seguida, escolha Confirmar novamente.

Agora você criou um arquivo chamado `.npmignore` na raiz do seu repositório.

arquivo `package.json`

O `package.json` arquivo 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 seu aplicativo.

`package.json` Neste tutorial, há uma lista de dependências e um `test script`. O script de teste faz o seguinte:

- Usando o [mocha](#), o script de teste executa os testes unitários especificados em `hello-world/tests/unit/` e grava os resultados em um `junit.xml` arquivo usando o repórter [xunit](#).
- Usando [Istanbul \(nyc\)](#), o script de teste gera um relatório de cobertura de código (`clover.xml`) usando o [clover reporter](#). Para obter mais informações, consulte [Usando repórteres alternativos](#) na documentação de Istanbul.

Para adicionar o arquivo `package.json`

- No seu repositório, em Arquivos, escolha Criar arquivo.
- Em Nome do arquivo, digite:

```
package.json
```

- Na caixa de texto, insira o seguinte código:

```
{
  "name": "hello_world",
  "version": "1.0.0",
  "description": "hello world sample for NodeJS",
  "main": "app.js",
  "repository": "https://github.com/aws-labs/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. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `package.json` à raiz do repositório.

arquivo `sam-template.yml`

O `sam-template.yml` arquivo contém as instruções para implantar a função Lambda e o API Gateway e configurá-los juntos. Ele segue a [especificação do AWS Serverless Application Model modelo](#), que estende a especificação do AWS CloudFormation modelo.

Você usa um AWS SAM modelo neste tutorial em vez de um AWS 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, `AWS::Serverless::Function` cria uma função Lambda, uma função de execução do Lambda e mapeamentos de origem de eventos 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 ter mais informações, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#).

Para adicionar o arquivo `sam-template.yml`

1. No seu repositório, em Arquivos, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

sam-template.yml

3. Na caixa de texto, insira o seguinte código:

```
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. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `sam-template.yml` na pasta raiz do seu 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 `hello-world` fonte.

Para adicionar o arquivo `setup-sam.sh`

1. No seu repositório, em Arquivos, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

```
setup-sam.sh
```

3. Na caixa de texto, insira o seguinte código:

```
#!/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, substitua `us-west-2` pela sua região. AWS

4. Escolha Confirmar e, em seguida, 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 Lambda. Neste tutorial, o código retorna o texto `hello world`.

Para adicionar o arquivo `app.js`

1. No seu repositório, em Arquivos, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

```
hello-world/app.js
```

3. Na caixa de texto, insira o seguinte código:

```
// 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. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

Agora você criou uma pasta chamada `hello-world` e um arquivo chamado `app.js`.

arquivo `test-handler.js`

O `test-handler.js` arquivo contém testes de unidade para a função Lambda.

Para adicionar o arquivo `test-handler.js`

1. No seu repositório, em Arquivos, escolha Criar arquivo.
2. Em Nome do arquivo, digite:

```
hello-world/tests/unit/test-handler.js
```

3. Na caixa de texto, insira o seguinte código:

```
'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. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

Agora você adicionou um arquivo chamado `test-handler.js` abaixo da `hello-world/` `tests/unit` pasta.

Agora você adicionou todos os seus arquivos de origem.

Reserve um momento para verificar seu trabalho e garantir que você tenha colocado todos os arquivos nas pastas corretas. A estrutura de pastas é a seguinte:

```
.
|- 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 seu código-fonte do Lambda e o implanta. O fluxo de trabalho consiste nos seguintes blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação de teste (Test) — No gatilho, essa ação instala o [gerenciador de pacotes Node \(npm\)](#) e, em seguida, executa o `npm run test` comando. Esse comando diz ao npm para executar o `test script` definido no `package.json` arquivo. O `test script`, por sua vez, executa os testes unitários e gera dois relatórios: um relatório de teste (`junit.xml`) e um relatório de cobertura de código (`clover.xml`). Para ter 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 obter mais informações sobre a ação de teste, consulte [Testando 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 obter mais informações sobre a ação de criação, consulte [Construindo 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.

Para criar um fluxo de trabalho

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha Criar fluxo de trabalho.
3. Em Repositório de origem, escolha `codecatalyst-cfn-source-repository`.
4. Para Branch, escolha `main`.
5. Selecione Create (Criar).
6. Exclua o código de amostra YAML.
7. Adicione o seguinte código YAML:

```
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 ter mais informações, consulte [Etapa 3: adicionar AWS funções a CodeCatalyst](#).
- *codecatalyst-build-role* com o nome da função de construção que você criou 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 de *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 ambas as regiões estejam definidas como *us-west-2*. Para obter detalhes sobre as regiões suportadas pelo Amazon S3 AWS CloudFormation, consulte [Endpoints e cotas de serviço](#) no. Referência geral da AWS

- *codecatalyst-deploy-role* com o nome da função de implantação que você criou 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/ com* o Amazon Resource Name (ARN) StackRole da função de pilha que você criou. [Etapa 2: criar AWS funções](#)

 Note

Se você decidiu não criar, implantar e empilhar funções, *codecatalyst-build-roles* substitua e *arn:aws:iam: StackRole :111122223333:role/* pelo nome ou ARN da função. *codecatalyst-deploy-role* CodeCatalystWorkflowDevelopmentRole-*spaceName* Para obter mais informações sobre essa função, consulte [Etapa 2: criar AWS funções](#).

Para obter informações sobre as propriedades no código mostrado anteriormente, consulte [Definição YAML da ação "Implantar AWS CloudFormation pilha"](#) o.

8. (Opcional) Escolha Validar para garantir que o código YAML seja válido antes de confirmar.
9. Escolha Commit (Confirmar).
10. Na caixa de diálogo Confirmar fluxo de trabalho, digite o seguinte:
 - a. Para o nome do arquivo do fluxo de trabalho, mantenha o padrão, `codecatalyst-cfn-workflow`.
 - b. Em Confirmar mensagem, digite:

```
add initial workflow file
```

- c. Para Repositório, escolha `codecatalyst-cfn-source-repository`.
- d. Em Nome da filial, escolha principal.
- e. Escolha Commit (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 `codecatalyst-cfn-workflow.yaml` arquivo ao seu 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, em seguida, escolha 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 a ID da execução.
5. Escolha Teste para ver o progresso dos testes.
6. Escolha BuildBackendver o progresso da construção.
7. Escolha DeployCloudFormationStackver o progresso da implantação.

Para obter mais informações sobre a visualização dos detalhes da execução, consulte [Visualizando o status e os detalhes da 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 AWS CloudFormation console AWS antes de executar novamente o fluxo de trabalho.

Para verificar a implantação

1. Depois de uma implantação bem-sucedida, escolha Variáveis (7) na barra de menu horizontal próxima à parte superior. (Não escolha Variáveis no painel à direita.)
2. Em seguida HelloWorldApi, cole o `https://` URL em um navegador.

Uma mensagem JSON hello world da função Lambda é exibida, indicando que o fluxo de trabalho implantou e configurou a função Lambda e o API Gateway com sucesso.

Tip

Você pode CodeCatalyst exibir esse URL no diagrama do fluxo de trabalho com algumas configurações pequenas. Para ter mais informações, consulte [Exibindo a URL do aplicativo implantado no diagrama do fluxo de trabalho](#).

Para verificar os resultados do teste unitário e a cobertura do código

1. No diagrama do fluxo de trabalho, escolha Teste e, em seguida, 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 AWS Management Console 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 `Configuration/name` valor 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. Escolha sua função Lambda, `codecatalyst-cfn-stack-HelloWorldFunction-string`
6. Você pode ver como o API Gateway é um gatilho para a função. Essa integração foi configurada automaticamente pelo tipo `AWS SAM AWS::Serverless::Function` de recurso.

Etapa 7: faça uma alteração

Nesta etapa, você faz uma alteração no código-fonte do Lambda e o confirma. Esse commit inicia a execução de um novo fluxo de trabalho. Essa execução implanta a nova função Lambda em um esquema azul esverdeado que usa a configuração padrão de mudança de tráfego especificada no console Lambda.

Para fazer uma alteração na sua fonte do Lambda

1. Em CodeCatalyst, navegue até seu projeto.
2. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
3. Escolha seu repositório `codecatalyst-cfn-source-repository` de origem.
4. Altere o arquivo do aplicativo:
 - a. Escolha a pasta `hello-world`.
 - b. Escolha o arquivo `app.js`.
 - c. Selecione a opção Editar.
 - d. Na linha 23, mude `hello world` para **Tutorial complete!**.
 - e. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

A confirmação faz com que a 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 unitários:
 - a. Selecione `hello-world\tests\unit\test-handler.js`.
 - b. Selecione a opção Editar.
 - c. Na linha 19, mude `hello world` para **Tutorial complete!**.
 - d. Escolha Confirmar e, em seguida, escolha Confirmar novamente.

A confirmação faz com que outra execução do fluxo de trabalho seja iniciada. Essa corrida será bem-sucedida.

6. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
7. Escolha `ecodecatalyst-cfn-workflow`, em seguida, escolha Executar.
8. Escolha o ID de execução da última execução. Ainda deve estar 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, escolha Variáveis (7) na parte superior.
11. Em seguida HelloWorldApi, cole o `https://` URL em um navegador.

Uma `Tutorial complete!` mensagem aparece no navegador, indicando que seu novo aplicativo foi implantado com sucesso.

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. Excluir `codecatalyst-cfn-workflow`.
3. Excluir `codecatalyst-cfn-environment`.
4. Excluir `codecatalyst-cfn-source-repository`.
5. Excluir `codecatalyst-cfn-project`.

Para limpar no AWS Management Console

1. Limpe da CloudFormation seguinte forma:
 - a. Abra o AWS 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 do API Gateway e dos serviços 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 o conteúdo do bucket.
 - d. Excluir o bucket.
3. Limpe no IAM, da seguinte forma:
 - a. Abra o console 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 AWS CloudFormation stack.

Adicionando a ação “Deploy AWS CloudFormation stack”

Use as instruções a seguir para adicionar a ação Deploy AWS CloudFormation stack ao seu fluxo de trabalho.

Visual

Para adicionar a ação “Implantar AWS 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy AWS CloudFormation stack e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Deploy AWS CloudFormation stack. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar a ação “Implantar AWS 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação Deploy AWS CloudFormation stack e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Deploy AWS CloudFormation stack. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#).

11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Configurando reversões

Por padrão, se a ação Implantar AWS CloudFormation pilha falhar, ela fará AWS 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 AWS CloudFormation stack. Para ter mais informações, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#).
2. Na função especificada no campo opcional Função da pilha - da ação Implantar AWS CloudFormation pilha, certifique-se de incluir a CloudWatchFullAccesspermissão. Para obter informações sobre como criar essa função com as permissões apropriadas, consulte [Etapa 2: criar AWS funções](#).


Para configurar alarmes de reversão para a ação “Implantar AWS CloudFormation pilha”

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

3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Deploy AWS CloudFormation stack. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. Escolha sua ação Deploy AWS CloudFormation stack.
8. No painel de detalhes, escolha Configuração.
9. Na parte inferior, expanda Avançado.
10. Em Monitorar ARNs de alarme, escolha Adicionar alarme.
11. Insira as informações nos campos a seguir.

- 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 ter mais informações, consulte [Configurando 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” AWS CloudFormation

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Deploy AWS CloudFormation stack. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Adicione as `monitor-timeout-in-minutes` propriedades `monitor-alarm-arns` e no código YAML para adicionar gatilhos de reversão. Para obter uma explicação de cada propriedade, consulte [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#).
8. Na função especificada na `role-arn` propriedade da ação Deploy AWS CloudFormation stack, certifique-se de incluir a `CloudWatchFullAccess` permissão. Para obter informações sobre como criar essa função com as permissões apropriadas, consulte [Etapa 2: criar AWS funções](#).

Visual

Para desativar as reversões da ação “Implantar AWS CloudFormation pilha”

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Deploy AWS CloudFormation stack. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. Escolha sua ação Deploy AWS CloudFormation stack.
8. No painel de detalhes, escolha Configuração.

9. Na parte inferior, expanda Avançado.
10. Ative a opção Desativar reversão.

YAML

Para desativar as reversões da ação “Implantar AWS CloudFormation pilha”

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome de um fluxo de trabalho que inclua a ação Deploy AWS CloudFormation stack. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Adicione a `disable-rollback: 1` propriedade no código YAML para interromper as reversões. Para obter uma explicação sobre essa propriedade, consulte [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#).

Variáveis produzidas pela ação “Deploy AWS CloudFormation stack”

A ação Deploy AWS CloudFormation stack produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

Chave	Valor
plataforma de implantação	O nome da plataforma de implantação. Codificado para. <code>AWS:CloudFormation</code>
região	O código da região em Região da AWS que foi implantado durante a execução do fluxo de trabalho.

Chave	Valor
	Exemplo: us-west-2
identificação da pilha	<p>O Amazon Resource Name (ARN) da pilha implantada.</p> <p>Exemplo: arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cfn-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</p>

Definição YAML da ação “Implantar AWS CloudFormation pilha”

A seguir está a definição YAML da ação Deploy AWS CloudFormation stack. Para saber como usar essa ação, consulte [Implantando uma AWS CloudFormation pilha 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: `DeployCloudFormationStack_nn`.

UI correspondente: guia de 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 ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: `aws/cfn-deploy@v1`.

UI correspondente: diagrama de fluxo de trabalho/ `DeployCloudFormationStack_nn/aws/cfn-deploy@v1` label

DependsOn

(DeployCloudFormationStack/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(DeployCloudFormationStack/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(DeployCloudFormationStack/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia Configuração/Avançado - opcional/Tipo de computação

Fleet

(DeployCloudFormationStack/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

UI correspondente: guia Configuração/Avançado - 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/Tempo limite em minutos - opcional

Environment

(DeployCloudFormationStack/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(DeployCloudFormationStack/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(DeployCloudFormationStack/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Name

(DeployCloudFormationStack/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Role


(DeployCloudFormationStack/Environment/Connections/Role)

(Obrigatório)

Especifique o nome da função do IAM que a ação Deploy AWS CloudFormation stack usa para acessar AWS e o AWS CloudFormation serviço.

Certifique-se de que essa função inclua as seguintes políticas:

- A seguinte política de permissões:

 Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
```

```

    "cloudformation:Describe*",
    "cloudformation:UpdateStack",
    "cloudformation:CreateChangeSet",
    "cloudformation>DeleteChangeSet",
    "cloudformation:ExecuteChangeSet",
    "cloudformation:SetStackPolicy",
    "cloudformation:ValidateTemplate",
    "cloudformation:List*",
    "iam:PassRole"
  ],
  "Resource": "*",
  "Effect": "Allow"
}]
}

```

Note

Na primeira vez em que a função for usada, use o caractere curinga a seguir na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A seguinte política de confiança personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Certifique-se de que essa função seja adicionada à conexão da conta em seu espaço. Para saber mais sobre como adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a `CodeCatalystWorkflowDevelopmentRole-spaceName` função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Função

Inputs

(*DeployCloudFormationStack*/Inputs)

(Optional)

A Inputs seção define os dados `DeployCloudFormationStack` necessários 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 AWS CloudFormation pilha.

Se você precisar se referir a arquivos que residem em entradas diferentes (digamos, uma fonte e um artefato), a entrada de origem é a entrada primária e o artefato é a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferenciá-las das primárias. Para obter detalhes, consulte [Exemplo: referenciando arquivos em vários artefatos](#).

UI 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Configuration

(DeployCloudFormationStack/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI correspondente: guia Configuração

name

(DeployCloudFormationStack/Configuration/name)

(Obrigatório)

Especifique um nome para a CloudFormation pilha que a ação Deploy AWS CloudFormation stack cria ou atualiza.

UI correspondente: guia de 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 da , consulte Regions and Endpoints.

UI correspondente: guia de 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 da 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 fonte encontrá-lo. Para obter mais informações, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#).

UI correspondente: guia de 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`.

Certifique-se de que a função da pilha inclua:

- Uma ou mais políticas de permissões. As políticas dependem dos recursos que você tem em sua 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 um aplicativo sem servidor usando AWS CloudFormation](#), ele inclui um procedimento intitulado, [Para criar uma função de pilha](#) que lista as permissões que a função de pilha precisa se você estiver implantando uma pilha típica de aplicativos sem servidor.

 Warning

Limite as permissões às exigidas pelo CloudFormation serviço para acessar os recursos em sua pilha. Usar uma função com permissões mais amplas pode representar um risco de segurança.

- A seguinte política de confiança:

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

```
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudformation.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
```

Opcionalmente, associe essa função à conexão da sua conta. Para saber mais sobre como associar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#). Se você não associar a função da pilha à conexão da conta, a função da pilha não aparecerá na lista suspensa da função da pilha no editor visual; no entanto, o ARN da função ainda poderá ser especificado no campo usando o editor YAML. `role-arn`

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia de configuração/função de pilha - opcional

capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Obrigatório)

Especifique uma lista dos recursos do IAM necessários para permitir AWS CloudFormation a criação de determinadas pilhas. Na maioria dos casos, você pode sair `capabilities` com o valor padrão `CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_AUTO_EXPAND`.

Se você ver os ##[error] requires capabilities: [*capability-name*] registros da ação do Deploy AWS CloudFormation stack, consulte [Como faço para corrigir erros de recursos do IAM?](#) para obter informações sobre como corrigir o problema.

Para obter mais informações sobre os recursos do IAM, consulte Como [reconhecer recursos do IAM em AWS CloudFormation modelos](#) no Guia do usuário do IAM.

UI correspondente: guia Configuração/Avançada/Capacidades

parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Optional)

Especifique parâmetros no seu AWS 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 parameter-overrides propriedade 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 seguintes formatos:

- Se seu 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 obter detalhes.

- Se seu arquivo JSON residir no seu repositório de origem, use:

```
file:///sources/WorkflowSource/path-to-json-file
```

Consulte o Exemplo 2 para obter 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 seu 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
```

Para usar uma lista de parâmetros separada por vírgula

- Adicione pares nome-valor do parâmetro na `parameter-overrides` propriedade usando o seguinte formato:

param-1=value-1,param-2=value-2

Por exemplo, assumindo o seguinte AWS CloudFormation modelo:

```
##My CloudFormation template

Description: My AWS 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 `parameter-overrides` propriedade 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.

UI correspondente:

- Guia de configuração/Avançada/Substituições de parâmetros
- Guia de configuração/Avançado/Substituições de parâmetros/Especificar substituições usando um arquivo
- Guia de configuração/Avançado/Substituições de parâmetros/Especificar 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 AWS CloudFormation [deploy](#) na Referência de AWS CLI Comandos. Para obter mais informações sobre a visualização de conjuntos de alterações, consulte [Visualizar um conjunto de alterações](#) no Guia AWS CloudFormation do usuário.

UI correspondente: guia Configuração/Avançado/Sem conjunto de alterações de execução

fail-on-empty-changeset

(DeployCloudFormationStack/Configuration/fail-on-empty-changeset)

(Optional)

Especifique se você deseja CodeCatalyst falhar na ação Deploy AWS CloudFormation stack se o conjunto de CloudFormation alterações estiver vazio. (Se um conjunto de alterações estiver vazio, 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 UPDATE_COMPLETE mensagem mesmo que a pilha não tenha sido atualizada.

Para obter mais informações sobre essa configuração, consulte o parâmetro AWS CloudFormation [deploy](#) na Referência de AWS CLI Comandos. Para obter mais informações sobre conjuntos de alterações, consulte [Atualização de pilhas usando conjuntos de alterações](#) no Guia do AWS CloudFormation usuário.

UI 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 AWS CloudFormation [deploy](#) na Referência de AWS CLI Comandos.

Para obter mais informações sobre como a ação Deploy AWS CloudFormation stack lida com reversões, consulte. [Configurando reversões](#)

Para obter mais informações sobre como reverter uma pilha, consulte [Opções de falha de pilha](#) no Guia do AWS CloudFormation usuário.

UI correspondente: guia Configuração/Avançado/Desativar reversão

termination-protection

(*DeployCloudFormationStack*/Configuration/termination-protection)

(Optional)

Especifique se você deseja que a AWS 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 rescisão. Para obter mais informações, consulte [Protegendo uma pilha de ser excluída](#) no Guia do AWS CloudFormation usuário.

UI correspondente: guia Configuração/Avançada/Proteção de terminação

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 não CloudFormation conseguir criar a pilha inteira no tempo alocado, ela falha na criação da pilha devido ao tempo limite e reverte a pilha.

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 AWS CloudFormation stack é executada, CodeCatalyst CloudFormation

coordena-se para enviar uma notificação por AWS CloudFormation evento que ocorre durante o processo de criação ou atualização da pilha. (Os eventos são visíveis na guia Eventos do AWS CloudFormation console para a pilha.) Você pode especificar até cinco tópicos. Para obter mais informações, consulte [O que é a Amazon SNS?](#)


UI correspondente: guia Configuração/Avançado/ARNs de notificação

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 ter mais informações, consulte [Configurando reversões](#).

UI correspondente: guia Configuração/Avançado/Monitorar ARNs de alarme

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.

UI 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 arbitrários de valores-chave que você pode usar para identificar sua pilha para fins como alocação de custos. Para obter mais informações sobre o que as tags são e como elas podem ser usadas, consulte [Marcar recursos](#) no Guia do usuário do Amazon EC2. Para obter mais informações sobre como marcar CloudFormation, consulte [Configuração das opções de AWS 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.

Você pode adicionar até 50 tags exclusivas para cada pilha.

UI correspondente: guia Configuração/Avançado/Tags

Implantando um AWS Cloud Development Kit (AWS CDK) aplicativo com um fluxo de trabalho

Esta seção descreve como implantar um AWS CDK aplicativo em sua AWS conta usando um fluxo de trabalho. Para fazer isso, você deve adicionar a ação de AWS CDK implantação 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.

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 AWS CDK implantação 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 AWS CDK a imagem de compilação. CodeCatalyst](#)

Se você especificou a versão 1.0.13 ou posterior, a ação vem junto com uma [versão específica](#) da CLI do CDK, 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 obter mais informações sobre esse comando, consulte o tópico [AWS CDK Kit de ferramentas \(comando cdk\)](#) no Guia do AWS Cloud Development Kit (AWS CDK) desenvolvedor.

Versões da CLI do CDK usadas pela ação AWS CDK “deploy”

A tabela a seguir mostra qual versão da CLI do CDK é usada por padrão por diferentes versões AWS CDK da ação de implantação.

Note

Talvez você consiga substituir o padrão. Para ter mais informações, consulte [CdkCliVersion](#) no [Definição YAML da ação “AWS CDK implantar”](#).

versão de ação “AWS CDK implantar”	AWS CDK Versão CLI
1.0.0 — 1.0.12	mais recente
1.0.13 ou posterior	2.99.1

Quantas pilhas a ação pode implantar?

A AWS CDK implantação 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.

Tópicos

- [Exemplo de fluxo de trabalho que implanta um aplicativo AWS CDK](#)
- [Adicionando a ação “AWS CDK implantar”](#)

- [Variáveis produzidas pela ação “AWS CDK implantar”](#)
- [Definição YAML da ação “AWS CDK implantar”](#)

Exemplo de fluxo de trabalho que implanta um aplicativo AWS CDK

O exemplo de fluxo de trabalho a seguir inclui a ação de AWS CDK implantação, junto com a ação de AWS CDK bootstrap. O fluxo de trabalho consiste nos seguintes blocos de construção que são executados sequencialmente:

Note

O exemplo de fluxo de trabalho a seguir serve para fins ilustrativos e não funcionará sem configuração adicional. O objetivo é dar um exemplo de como pode ser um fluxo de trabalho quando configurado com a ação de AWS CDK implantação.

- 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação de AWS CDK bootstrap (CDKBootstrap) — No gatilho, a ação implanta a pilha de CDKToolkit bootstrap em. AWS Se a CDKToolkit pilha 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 AWS CloudFormation modelo e implanta a pilha definida no modelo em. AWS

```
Name: codecatalyst-cdk-deploy-workflow
```

```
SchemaVersion: 1.0
```

```
Triggers:
```

```
- Type: PUSH
```

```
  Branches:
```

```
    - main
```

```
Actions:
```

```
  CDKBootstrap:
```

```
Identifier: aws/cdk-bootstrap@v1
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@v1
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 de AWS CDK implantação ao seu fluxo de trabalho.

Antes de começar

Antes de adicionar a ação de AWS CDK implantação 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 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

2. Inicialize seu AWS ambiente. Para inicializar, você pode:

- Use um dos métodos descritos em [Como inicializar no](#) Guia do AWS Cloud Development Kit (AWS CDK) desenvolvedor.
- Use a ação AWS CDK bootstrap. Você pode adicionar essa ação no mesmo fluxo de trabalho da sua AWS CDK implantação ou em um diferente. Apenas certifique-se de que a ação de bootstrap seja executada pelo menos uma vez antes de executar a ação de AWS CDK implantação 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 obter mais informações sobre bootstrapping, consulte [Bootstrapping no Guia do desenvolvedor](#).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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação de AWS CDK implantação e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha AWS CDK implantar. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:

- (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da ação “AWS CDK implantar”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
 12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se sua ação de AWS CDK implantação falhar com um `npm install` erro, consulte [Como faço para corrigir erros de “instalação do npm”?](#) para obter 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação de AWS CDK implantação e faça o seguinte:

- Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha AWS CDK implantar. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Baixar para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da ação “AWS CDK implantar”](#).
 11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
 12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se sua ação de AWS CDK implantação falhar com um `npm install` erro, consulte [Como faço para corrigir erros de “instalação do npm”?](#) para obter informações sobre como corrigir o erro.

Variáveis produzidas pela ação “AWS CDK implantar”

A ação de AWS CDK implantação produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

Chave	Valor
identificação da pilha	O Amazon Resource Name (ARN) da pilha de AWS CDK aplicativos que foi implantada durante a execução do fluxo de trabalho.

Chave	Valor
	Exemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-app-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>
plataforma de implantação	O nome da plataforma de implantação. Codificado para. <code>AWS:CloudFormation</code>
região	O código da região em Região da AWS que foi implantado durante a execução do fluxo de trabalho. Exemplo: <code>us-west-2</code>
IGNORAR A IMPLANTAÇÃO	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. Essa variável só é produzida se seu valor for <code>true</code> . Codificado para. <code>true</code>

Chave	Valor
AWS CloudFormation variables	Além de gerar as variáveis listadas anteriormente, a ação de AWS CDK implantação também expõe as variáveis de CloudFormation saída 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 de AWS CDK implantação uma vez e, em seguida, 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 <code>CfnOutputVariables</code> YAML. Para obter mais informações, consulte a descrição da CfnOutputVariables propriedade no Definição YAML da ação “AWS CDK implantar” .

Definição YAML da ação “AWS CDK implantar”

A seguir está a definição YAML da ação de AWS CDK implantação. Para saber como usar essa ação, consulte [Implantando um AWS Cloud Development Kit \(AWS CDK\) 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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@v1
  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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: `CDKDeploy_nn`.

UI correspondente: guia de configuração/nome da ação

Identifier

(*CDKDeploy*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: `aws/cdk-deploy@v1`.

UI correspondente: diagrama de fluxo de trabalho/ CDKDeploy _nn/ aws/cdk-deploy @v1 label

DependsOn

(*CDKDeploy*/DependsOn)

(Optional)

Especifique uma ação ou um grupo de ações que deve ser executado com êxito para que a ação de AWS CDK implantação seja executada. Recomendamos especificar a ação de AWS CDK bootstrap na DependsOn propriedade, assim:

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
```

Note

A [inicialização é um pré-requisito obrigatório para implantar um aplicativo](#). AWS CDK Se você não incluir a ação AWS CDK Bootstrap em seu fluxo de trabalho, deverá encontrar outra maneira de implantar a pilha de AWS CDK bootstrap antes de executar sua AWS CDK ação de implantação. Para obter mais informações, consulte [Adicionando a ação "AWS CDK implantar"](#) em [Implantando um AWS Cloud Development Kit \(AWS CDK\) aplicativo com um fluxo de trabalho](#).

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*CDKDeploy*/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(*CDKDeploy*/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia Configuração/Avançado - opcional/Tipo de computação

Fleet

(*CDKDeploy*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 é concluída. Exemplos de frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

UI correspondente: guia Configuração/Avançado - 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é o mesmo que o tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Inputs

(*CDKDeploy*/Inputs)

(Optional)

A Inputs seção define os dados `CDKDeploy` necessários durante a execução de um fluxo de trabalho.

Note

Somente uma entrada (uma fonte ou um artefato) é permitida para cada ação de AWS CDK implantação.

UI 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 de AWS CDK implantação sintetiza o aplicativo 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI 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.

UI 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos

Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique o nome do artefato que conterá o AWS 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. Considere salvar o modelo sintetizado em um artefato para preservar um registro dele para fins de teste ou solução de problemas.


UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Criar nome do artefato

Files

(*CDKDeploy*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `"cdk.out/**/*"` para incluir o modelo sintetizado AWS 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 do `cdk.out` seu `cdk.json` arquivo, especifique esse diretório aqui em vez de `cdk.out`.

UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Arquivos produzidos pela compilação

Environment

(*CDKDeploy*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(*CDKDeploy*/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(*CDKDeploy*/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Name

(*CDKDeploy*/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Role

(*CDKDeploy*/Environment/Connections/Role)

(Obrigatório)

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 que essa função inclua:

- A seguinte política de permissões:

Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "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::aws-account:role/cdk-*"
    }
  ]
}
```

- A seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",

```

```
        "codecatalyst.amazonaws.com"
      ],
    },
    "Action": "sts:AssumeRole"
  }
]
```

Certifique-se de que essa função seja adicionada à conexão da sua conta. Para saber mais sobre como adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia Configuração/'Ambiente/Conta/Função'/Função

Configuration

(*CDKDeploy*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI 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 0 exemplo a seguir mostra o conteúdo de um arquivo de TypeScript ponto de entrada, com o nome da pilha destacado em itálico vermelho.* Se o arquivo do ponto de entrada estiver em um idioma diferente, 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 só pode especificar 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.

UI correspondente: guia de configuração/nome da pilha

Region

(*CDKDeploy*/Configuration/Region)

(Obrigatório)

Especifique Região da AWS no qual a pilha de AWS CDK aplicativos será implantada. Para obter uma lista dos códigos das regiões da , consulte Regions and Endpoints.

UI correspondente: guia de 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 própria pilha, bem como aos recursos individuais na pilha. Para obter mais

informações sobre marcação, consulte [Marcação](#) no Guia do AWS Cloud Development Kit (AWS CDK) desenvolvedor.

UI correspondente: guia Configuração/Avançado - 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 obter mais informações sobre contextos, consulte [Contextos de tempo de execução](#) no Guia do AWS Cloud Development Kit (AWS CDK) desenvolvedor.

UI correspondente: guia Configuração/Avançado - opcional/Contexto

CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Optional)

Essa propriedade está disponível com a versão 1.0.13 ou posterior da ação de AWS CDK implantação e a versão 1.0.8 ou posterior da AWS CDK ação bootstrap.

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 seu aplicativo.

Ou

- latest. Considere 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 obter uma lista das versões compatíveis da CLI do CDK que você pode usar, AWS CDK consulte Versões.](#)

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 AWS CDK “deploy”](#)
- [Versões do CDK CLI usadas pela ação AWS CDK “bootstrap”](#)

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 de AWS CDK implantação 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.

UI correspondente: guia de 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. Em seguida, você pode referenciar as variáveis de saída do fluxo de trabalho em ações subsequentes em seu fluxo de trabalho. Para obter mais informações sobre variáveis em CodeCatalyst, consulte [Configurando e usando variáveis em um fluxo 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, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'bucketName', {
      value: bucket.bucketName,
      description: 'The name of the s3 bucket',
      exportName: 'myBucket',
    });
    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 sua `CfnOutputVariables` propriedade tem a seguinte aparência:

Configuration:

```
...
CfnOutputVariables: '["bucketName","tableName"]'
```

... então a ação gera as seguintes variáveis de saída do fluxo de trabalho:

Chave	Valor
bucketName	bucket.bucketName
tableName	table.tableName

Em seguida, você pode referenciar as `tableName` variáveis `bucketName` e em ações subsequentes. Para saber como referenciar variáveis de saída do fluxo de trabalho em ações subsequentes, consulte [Referenciando 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 ter mais informações, consulte [Variáveis produzidas pela ação “AWS CDK implantar”](#).

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 AWS 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 AWS 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 AWS 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 `--app` opção está presente, a `cdk synth` operação não ocorre.

Se você não especificar um diretório de montagem em nuvem, a ação de AWS CDK implantação executará o `cdk deploy` comando sem a `--app` opção. 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 conjunto 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 uma montagem em nuvem, é possível que a ação de AWS CDK implantação sintetize e implante arquivos diferentes, dependendo de quando ela for executada. Por exemplo, a ação de AWS CDK implantação pode sintetizar um conjunto 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 do diretório de montagem Path to cloud (editor visual) ou a `CloudAssemblyRootPath` propriedade (editor YAML) para especificar o conjunto de nuvem já sintetizado.

- Use gerenciadores de pacotes e ferramentas não padrão com o aplicativo AWS CDK

Durante uma `synth` operação, a ação de AWS CDK implantação tenta executar seu aplicativo usando ferramentas padrão, como `npm` ou `pip`. Se a ação não conseguir executar seu aplicativo com êxito 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 a montagem em nuvem for gerada, você poderá especificá-la no campo Caminho para o diretório de montagem em nuvem (editor visual) ou na `CloudAssemblyRootPath` propriedade (editor YAML) da ação de AWS CDK implantação.

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 obter informações sobre os `cdk synth` comandos `cdk deploy` e, bem como sobre a `--app` opção, consulte [Implantando pilhas, Sintetizando pilhas e Ignorando a síntese](#) no Guia do desenvolvedor.AWS Cloud Development Kit (AWS CDK)

Para obter informações sobre assemblies em nuvem, consulte [Cloud Assembly](#) na Referência da AWS Cloud Development Kit (AWS CDK) API.

UI correspondente: guia de configuração/caminho para o diretório de montagem em 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 AWS CDK bootstrap ao seu fluxo de trabalho.

[A ação AWS CDK bootstrap provisiona uma pilha de bootstrap em seu AWS ambiente usando o modelo moderno.](#) Se uma pilha de bootstrap já existir, a ação a atualiza, se necessário. Ter uma pilha de bootstrap presente AWS é um pré-requisito para implantar um aplicativo. AWS CDK

Para obter mais informações sobre bootstrapping, consulte [Bootstrapping no Guia do desenvolvedor](#).AWS Cloud Development Kit (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 bootstrap usando outro mecanismo e deseja mantê-la intacta (sem atualizações).
- Você deseja usar um [modelo de bootstrap personalizado](#), que não é compatível com a ação AWS CDK bootstrap.

Como funciona a ação "AWS CDK bootstrap"

O AWS CDK bootstrap 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, portanto, nenhum download ocorre.

- A ação usa a CLI do CDK para executar o comando `cdk bootstrap`. Esse comando executa as tarefas de inicialização descritas no tópico [Bootstrapping](#) no Guia do desenvolvedor.AWS Cloud Development Kit (AWS CDK)

Versões do CDK CLI usadas pela ação AWS CDK "bootstrap"

A tabela a seguir mostra qual versão da CLI do CDK é usada por padrão por diferentes versões AWS CDK da ação de bootstrap.

Note

Talvez você consiga substituir o padrão. Para ter mais informações, consulte [CdkCliVersion](#) no [definição YAML da ação "AWS CDK bootstrap"](#).

versão de ação "AWS CDK bootstrap"	AWS CDK Versão CLI
1.0.0 — 1.0.7	mais recente
1.0.8 ou posterior	2.99.1

Tópicos

- [Exemplo de fluxo de trabalho que inicializa um aplicativo AWS CDK](#)
- [Adicionando a ação "AWS CDK bootstrap"](#)
- [Variáveis produzidas pela ação "AWS CDK bootstrap"](#)
- [definição YAML da ação "AWS CDK bootstrap"](#)

Exemplo de fluxo de trabalho que inicializa um aplicativo AWS CDK

Consulte o [Exemplo de fluxo de trabalho que implanta um aplicativo AWS CDK](#) no para [Implantando um AWS Cloud Development Kit \(AWS CDK\) aplicativo com um fluxo de trabalho](#) obter um fluxo de trabalho que inclui a ação de AWS CDK bootstrap.

Adicionando a ação "AWS CDK bootstrap"

Use as instruções a seguir para adicionar a ação de AWS CDK bootstrap ao seu fluxo de trabalho.

Antes de começar

Antes de usar a ação AWS CDK bootstrap, verifique se você tem um AWS CDK aplicativo pronto. A ação bootstrap sintetizará o AWS CDK aplicativo antes da inicialização. Você pode escrever seu aplicativo em qualquer linguagem de programação compatível com AWS CDK o.

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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação de AWS CDK bootstrap e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha AWS CDK bootstrap. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [definição YAML da ação “AWS CDK bootstrap”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Note


Se sua ação de AWS CDK bootstrap falhar com um `npm install` erro, consulte [Como faço para corrigir erros de “instalação do npm”?](#) para obter 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Procure a ação de AWS CDK bootstrap e escolha + para adicioná-la ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
10. Modifique as propriedades no código YAML de acordo com suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [definição YAML da ação “AWS CDK bootstrap”](#).

11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

 Note

Se sua ação de AWS CDK bootstrap falhar com um `npm install` erro, consulte [Como faço para corrigir erros de “instalação do npm”?](#) para obter informações sobre como corrigir o erro.

Variáveis produzidas pela ação “AWS CDK bootstrap”

A ação AWS CDK bootstrap produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

Chave	Valor
plataforma de implantação	O nome da plataforma de implantação. Codificado para. <code>AWS:CloudFormation</code>
região	O código de região em Região da AWS que a pilha de AWS CDK bootstrap foi implantada durante a execução do fluxo de trabalho. Exemplo: <code>us-west-2</code>
identificação da pilha	O Amazon Resource Name (ARN) da pilha de bootstrap implantada. AWS CDK Exemplo: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>

Chave	Valor
IGNORAR A IMPLANTAÇÃO	<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 é produzida se seu valor for <code>true</code>.</p> <p>Codificado para: <code>true</code></p>

definição YAML da ação “AWS CDK bootstrap”

A seguir está a definição YAML da ação de AWS CDK bootstrap. 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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@v1
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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: CDKBootstrapAction_nn.

UI correspondente: guia de 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 ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: `aws/cdk-bootstrap@v1`.

UI correspondente: diagrama de fluxo de trabalho/ CDKBootstrapAction _nn/aws/cdk-bootstrap @v1 label

DependsOn

(CDKBootstrapAction/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(CDKBootstrapAction/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(CDKBootstrapAction/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia Configuração/Avançado - opcional/Tipo de computação

Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

UI correspondente: guia Configuração/Avançado - 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 5 minutos e o

máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Inputs

(*CDKBootstrapAction*/Inputs)

(Optional)

A Inputs seção define os dados que a ação de AWS CDK bootstrap precisa durante a execução de um fluxo de trabalho.

UI correspondente: guia Entradas

Note

Somente uma entrada (uma fonte ou um artefato) é permitida para cada ação de AWS CDK bootstrap.

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 AWS CDK bootstrap sintetiza o aplicativo nesse repositório antes de iniciar o processo de inicialização. Atualmente, o único rótulo de 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 seu AWS CDK aplicativo não estiver contido em um artefato, ele deverá residir no seu repositório de origem.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI 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.

UI 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos

Name

(CDKBootstrapAction/Outputs/Artifacts/Name)

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique o nome do artefato que conterá o AWS 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. Considere salvar o modelo sintetizado em um artefato para preservar um registro dele para fins de teste ou solução de problemas.

UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Criar nome do artefato

Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique os arquivos a serem incluídos no artefato. Você deve especificar `"cdk.out/**/*"` para incluir o modelo sintetizado AWS 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 do `cdk.out` seu `cdk.json` arquivo, especifique esse diretório aqui em vez de `cdk.out`.

UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Arquivos produzidos pela compilação

Environment

(*CDKBootstrapAction*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(*CDKBootstrapAction*/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(*CDKBootstrapAction*/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.


UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Obrigatório)

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 que essa função inclua as seguintes políticas:

 Note

As permissões mostradas na política de permissões a seguir são aquelas exigidas pelo `cdk bootstrap` comando para realizar sua inicialização. Essas permissões podem mudar se o comando `bootstrap for AWS CDK` alterado.

⚠ Warning

Use essa função somente com a ação AWS CDK bootstrap. É muito permissivo e usá-lo com outras ações pode representar um risco à segurança.

- A seguinte política de permissões:

⚠ Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "ssm:GetParameterHistory",
        "ecr:PutImageScanningConfiguration",
        "cloudformation:*",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ssm:GetParameters",
        "iam:PutRolePolicy",
        "ssm:GetParameter",
        "ssm>DeleteParameters",
        "ecr>DeleteRepository",
        "ssm:PutParameter",
        "ssm>DeleteParameter",
        "iam:PassRole",
        "ecr:SetRepositoryPolicy",
        "ssm:GetParametersByPath",
        "ecr:DescribeRepositories",
        "ecr:GetLifecyclePolicy"
      ],
      "Resource": [
```

```

        "arn:aws:ssm:aws-region:aws-account:parameter/cdk-bootstrap/*",
        "arn:aws:cloudformation:aws-region:aws-account:stack/CDKToolkit/*",
        "arn:aws:ecr:aws-region:aws-account:repository/cdk-*",
        "arn:aws:iam::aws-account:role/cdk-*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:RegisterType",
        "cloudformation:CreateUploadBucket",
        "cloudformation:ListExports",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:SetTypeDefaultVersion",
        "cloudformation:RegisterPublisher",
        "cloudformation:ActivateType",
        "cloudformation:ListTypes",
        "cloudformation:DeactivateType",
        "cloudformation:SetTypeConfiguration",
        "cloudformation:DeregisterType",
        "cloudformation:ListTypeRegistrations",
        "cloudformation:EstimateTemplateCost",
        "cloudformation:DescribeAccountLimits",
        "cloudformation:BatchDescribeTypeConfigurations",
        "cloudformation:CreateStackSet",
        "cloudformation:ListStacks",
        "cloudformation:DescribeType",
        "cloudformation:ListImports",
        "s3:*",
        "cloudformation:PublishType",
        "ecr:CreateRepository",
        "cloudformation:DescribePublisher",
        "cloudformation:DescribeTypeRegistration",
        "cloudformation:TestType",
        "cloudformation:ValidateTemplate",
        "cloudformation:ListTypeVersions"
    ],
    "Resource": "*"
}
]
}

```

Note

Na primeira vez em que a função for usada, use o caractere curinga a seguir na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- A seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Certifique-se de que essa função seja adicionada à conexão da sua conta. Para saber mais sobre como adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a CodeCatalystWorkflowDevelopmentRole-*spaceName* função tem

permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Função

Configuration

(*CDKBootstrapAction*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI 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 da , consulte Regions and Endpoints.

UI correspondente: guia de configuração/região

CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

(Optional)

Essa propriedade está disponível com a versão 1.0.13 ou posterior da ação de AWS CDK implantação e a versão 1.0.8 ou posterior da AWS CDK ação bootstrap.

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 seu aplicativo.

Ou

- `latest`. Considere 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 obter uma lista das versões compatíveis da CLI do CDK que você pode usar, AWS CDK consulte Versões.](#)

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 AWS CDK “deploy”](#)
- [Versões do CDK CLI usadas pela ação AWS CDK “bootstrap”](#)

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 de publicação do Amazon S3 ao seu fluxo de trabalho. A ação de publicação 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

Quando usar a ação “Amazon S3 publish”

Use essa ação se:

- Você tem 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 você deseja hospedar no Amazon S3. Nesse caso, seu fluxo de trabalho incluiria uma [ação de criação](#) para

criar os arquivos HTML e de suporte do site e uma ação de publicação do Amazon S3 para copiar os arquivos para o Amazon S3.

- Você tem um repositório de origem que contém arquivos que você deseja armazenar no Amazon S3.

Por exemplo, você pode ter um repositório de origem com arquivos de origem do aplicativo que deseja arquivar todas as noites no Amazon S3.

Tópicos

- [Exemplo de fluxo de trabalho que publica arquivos no Amazon S3](#)
- [Adicionando a ação “Amazon S3 publish”](#)
- [Definição YAML da ação “Amazon S3 publish”](#)

Exemplo de fluxo de trabalho que publica arquivos no Amazon S3

O exemplo de fluxo de trabalho a seguir inclui a ação de publicação do Amazon S3, junto com uma ação de criação. O fluxo de trabalho cria um site de documentação estática e depois o publica no Amazon S3, onde está hospedado. O fluxo de trabalho consiste nos seguintes blocos de construção que são executados sequencialmente:

Note

O exemplo de fluxo de trabalho a seguir serve para fins ilustrativos e só funcionará com configurações adicionais.

- Um gatilho — Esse gatilho inicia a execução automática do fluxo de trabalho quando você envia uma alteração ao seu repositório de origem. Para obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação de construção (BuildDocs) — No gatilho, a ação cria um site de documentação estática (`mkdocs build`) e adiciona os arquivos HTML associados e os metadados de suporte a um artefato chamado `MyDocsSite`. Para obter mais informações sobre a ação de criação, consulte [Construindo com fluxos de trabalho](#).
- Uma ação de publicação do Amazon S3 (`PublishToS3`) — Ao concluir a ação de criação, essa ação copia o site no `MyDocsSite` artefato para o Amazon S3 para hospedagem.

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: my-bucket

SourcePath: /artifacts/PublishToS3/MyDocSite/site

TargetPath: my/docs/site

Adicionando a ação “Amazon S3 publish”

Use as instruções a seguir para adicionar a ação de publicação do Amazon S3 ao seu fluxo de trabalho.

Visual

Para adicionar a ação “Amazon S3 publish” 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, em seguida, escolha Fluxos de trabalho.
 4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
 5. Selecione a opção Editar.
 6. Escolha Visual.
 7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
 8. Na lista suspensa, escolha Amazon. CodeCatalyst
 9. Pesquise a ação de publicação do Amazon S3 e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.
- Ou
- Escolha Amazon S3 Publish. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da ação “Amazon S3 publish”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar a ação “Amazon S3 publish” 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de publicação do Amazon S3 e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Amazon S3 Publish. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da ação “Amazon S3 publish”](#).
 11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Definição YAML da ação “Amazon S3 publish”

A seguir está a definição YAML da ação de publicação do 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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: s3-bucket-name
    TargetPath: my/target

```

S3Publish

(Obrigatório)

Especifique o nome da ação. Todos os nomes das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: S3Publish_nn.

UI correspondente: guia de configuração/nome da ação

Identifier

(*S3Publish*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: aws/s3-publish@v1.

UI correspondente: diagrama de fluxo de trabalho/ S3Publish_nn/ aws/s3-publish @v1 label

DependsOn

(*S3Publish*/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*S3Publish*/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(*S3Publish*/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia de configuração/tipo de computação

Fleet

(*S3Publish*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 é concluída. Exemplos de frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`

UI correspondente: guia de 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Inputs

(*S3Publish*/Inputs)

(Optional)

A `Inputs` seção define os dados `S3Publish` necessários durante a execução de um fluxo de trabalho.

Note

São permitidas no máximo quatro entradas (uma fonte e três artefatos) para cada ação de AWS CDK implantação. As variáveis não contam para esse total.

Se você precisar se referir a arquivos que residem em entradas diferentes (digamos, uma fonte e um artefato), a entrada de origem é a entrada primária e o artefato é a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferenciá-las das primárias. Para obter detalhes, consulte [Exemplo: referenciando arquivos em vários artefatos](#).

UI 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 contidos em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se seus arquivos não estiverem contidos em um artefato, eles devem residir no seu repositório de origem.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Variables - input

(*S3Publish*/Inputs/Variables)

(Optional)

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para habilitar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de entradas/variáveis - opcional

Environment

(*S3Publish*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#) [Criar um ambiente](#) e.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Ambiente

Name

(*S3Publish*/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Ambiente

Connections

(*S3Publish*/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Conexão

Name

(*S3Publish*/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Conexão

Role

(*S3Publish*/Environment/Connections/Role)

(Obrigatório)

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 que essa função inclua:

- A seguinte política de permissões:

⚠ Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "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 seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

Verifique se essa função está associada à conexão da sua conta. Para saber mais sobre como associar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Função

Configuration

(*S3Publish*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI 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 do `/myFolder` para o Amazon S3 e preserva a estrutura de diretórios subjacente.

Especificando `./myFolder/myfile.txt` cópias somente `myfile.txt` para o Amazon S3. (A estrutura de diretórios foi removida.)

Você não pode usar curingas.

Note

Talvez seja necessário adicionar um prefixo ao diretório ou ao caminho do arquivo para indicar em qual artefato ou fonte encontrá-lo. Para obter mais informações, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#).

UI correspondente: guia de configuração/caminho de origem

DestinationBucketName

(*S3Publish*/Configuration/DestinationBucketName)

(Obrigatório)

Especifique o nome do bucket do Amazon S3 em que você deseja publicar arquivos.

UI correspondente: guia de configuração/bucket de destino - opcional

TargetPath

(*S3Publish*/Configuration/TargetPath)

(Optional)

Especifique o nome e o caminho do diretório no Amazon S3 onde você deseja publicar seus arquivos. Se o diretório não existir, ele será criado. O caminho do diretório não deve incluir o nome do bucket.

Exemplos:

`myS3Folder`


```
./myS3Folder/myS3Subfolder
```

UI correspondente: guia de configuração/diretório de destino - opcional

Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst

Você pode fazer com que uma ação CodeCatalyst de fluxo de trabalho implante seu aplicativo e outros recursos em sua VPC Conta da AWS ou na Amazon VPC. Para fazer isso, você deve configurar um CodeCatalyst ambiente.

Um ambiente, que não deve ser confundido com um [ambiente de desenvolvimento](#), é onde o código é implantado. Geralmente, ele contém uma instância de um aplicativo em execução junto com sua infraestrutura associada. Você pode dar um nome ao seu ambiente, como desenvolvimento, teste, preparação ou produção. Todas as implantações CodeCatalyst geradas por um ambiente aparecerão na página Ambientes. Para configurar um ambiente, você dá a ele um nome, como `my-production-environment`, e depois o associa ao seu Conta da AWS.

Além de exibir informações de implantação, um ambiente também serve como mecanismo para atribuir funções AWS do IAM às [ações](#) do fluxo de trabalho.

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 `my-staging-environment` ambiente e outra a um `my-production-environment` ambiente.

Quais ações apoiam os ambientes?

As ações a seguir permitem que suas informações de implantação sejam exibidas na página Ambientes:

- Deploy AWS CloudFormation stack — Para obter mais informações, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#)
- Implante no Amazon ECS — Para obter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) com um fluxo de trabalho](#)
- Implemente no cluster Kubernetes — Para obter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#)
- AWS CDK implantar — Para obter mais informações, consulte [Implantando um AWS Cloud Development Kit \(AWS CDK\) aplicativo com um fluxo de trabalho](#)

Note

Se quiser permitir que uma ação acesse e realize operações em sua AWS conta, você precisará associá-la a um ambiente. Muitas ações apoiam uma associação ambiental, incluindo, mas não se limitando às ações listadas anteriormente. Você pode saber quais ações oferecem suporte a uma associação de ambiente porque elas incluirão uma lista suspensa Ambiente na guia Configuração no editor [visual](#).

Regiões compatíveis

A página Ambientes pode exibir recursos em qualquer AWS região.

Os ambientes são obrigatórios?

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).

Tópicos

- [Criar um ambiente](#)
- [Associando um ambiente, uma conexão de conta e uma função do IAM a uma ação de fluxo de trabalho](#)
- [Associando uma conexão VPC a um ambiente](#)
- [Associando um Conta da AWS a um ambiente](#)

Criar um ambiente


Use as instruções a seguir para criar um ambiente vazio que você possa associar posteriormente a uma ação de fluxo de trabalho.

Antes de começar

Você precisa do seguinte:

- Um CodeCatalyst espaço. Para ter mais informações, consulte [Configure e faça login em CodeCatalyst](#).

- Um CodeCatalyst projeto. Para ter mais informações, consulte [Criando um projeto com um blueprint](#).
- Uma conexão de AWS conta que inclui as funções do IAM que sua ação de fluxo de trabalho precisará acessar AWS. Você pode usar no máximo uma conexão de conta por ambiente. Para ter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

 Note

Você pode criar um ambiente sem uma conexão de conta; no entanto, você precisará voltar e adicionar a conexão posteriormente.

Para criar um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha 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 seu aplicativo para garantir que ele esteja funcionando conforme o esperado antes de colocá-lo em produção.
 - Produção — Um ambiente “ativo” que está disponível publicamente e hospeda seu aplicativo finalizado.

Se você escolher Produção, um selo de produção aparecerá na interface do 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 crachá, não há diferenças entre ambientes de produção e não produção.

6. (Opcional) Em Conexão VPC, escolha uma conexão VPC que você deseja associar a esse ambiente. Para obter mais informações sobre a criação dessa conexão VPC, consulte Gerenciando [Amazon Virtual Private Clouds no Guia](#) do CodeCatalyst Administrador.
7. (Opcional) Em Descrição, insira uma descrição como **Production environment for the hello-world app**.
8. Em Conexão - opcional, escolha a conexão de AWS conta que você deseja associar a esse ambiente. Certifique-se de que a conexão da conta inclua as funções do IAM que você

deseja associar ao ambiente. Para obter mais informações sobre como criar essa conexão, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

9. Escolha Criar ambiente. CodeCatalyst cria um ambiente vazio.

Próximas etapas

- Agora que você criou um ambiente, está pronto para associá-lo a uma ação de fluxo de trabalho. Para ter mais informações, consulte [Associando um ambiente, uma conexão de conta e uma função do IAM a uma ação de fluxo de trabalho](#).

Associando um ambiente, uma conexão de conta e uma função do IAM a uma ação de fluxo de trabalho

Quando você associa um ambiente, uma conexão de conta e uma função do IAM a uma [ação de fluxo de trabalho compatível](#), a função do IAM fica disponível para uso pela ação. Além de obter acesso a uma função do IAM, a ação também pode ter suas informações de implantação importadas para a página Ambientes. Para ter mais informações, consulte [Quais ações apoiam os ambientes?](#).

Use as instruções a seguir para associar um ambiente, uma conexão de conta e uma função do IAM a uma ação.

Etapa 1: associar o ambiente, a conexão da conta e a função a uma ação de fluxo de trabalho

Use o procedimento a seguir para associar um ambiente, uma conexão de conta e uma função a uma ação de fluxo de trabalho.

Visual

Para associar um ambiente, uma conexão de conta e uma função 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.

5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha uma ação compatível com ambientes. Para ter mais informações, consulte [Quais ações apoiam os ambientes?](#).
8. Escolha a guia Configuração e especifique as informações nos campos, da seguinte maneira.

Ambiente

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

Conexão de conta ou Conexão - opcional (o que estiver disponível)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em Environment.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

Função

Especifique o nome da função do IAM que essa ação usa para acessar e operar em AWS serviços como Amazon S3 e Amazon ECR. Certifique-se de que essa função seja adicionada à conexão da sua conta. Para adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função aqui, desde que ela tenha permissões suficientes. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a CodeCatalystWorkflowDevelopmentRole-*spaceName* função tem permissões muito amplas que podem representar um risco de segurança.

Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

Se você não vê a função na lista, é porque você não a associou à conexão da conta. Para ter mais informações, consulte [Adicionar funções do IAM às conexões da conta](#).

9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para associar um ambiente, uma conexão de conta e uma função 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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
Connections:
  - Name: account-connection-name
    Role: iam-role-name
```

Para obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Etapa 2: preencher o ambiente

Depois de associar um ambiente, uma conexão de conta e uma função à ação do fluxo de trabalho, você pode preencher a página Ambientes com informações de implantação. Use as instruções a seguir para preencher a página Ambientes.

Note

A página Ambientes é suportada somente por um subconjunto de ações do fluxo de trabalho. Para ter mais informações, consulte [Quais ações apoiam os ambientes?](#).

Para povoar o meio ambiente

1. Se a execução de um fluxo de trabalho não foi iniciada automaticamente quando você confirmou suas alterações [Etapa 1: associar o ambiente, a conexão da conta e a função a uma ação de fluxo de trabalho](#), inicie manualmente a execução da seguinte forma:
 - a. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
 - b. Escolha o nome do fluxo de trabalho em que você deseja iniciar uma execução. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
 - c. Escolha Executar.

A execução do fluxo de trabalho inicia uma nova implantação, o que faz com que CodeCatalyst as informações de recursos do aplicativo sejam adicionadas em Ambientes.

2. Verifique se os recursos do seu aplicativo aparecem em seu ambiente:
 - a. No painel de navegação, escolha CI/CD e, em seguida, escolha 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 com êxito os recursos do seu aplicativo.
 - d. Escolha a guia Alvos de implantação e verifique se os recursos do seu aplicativo aparecem.

Associando uma conexão 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, escolha CI/CD e, em seguida, escolha Ambientes.
4. Escolha seu ambiente (por exemplo, Production).
5. Escolha a guia Propriedades do ambiente.
6. Escolha Gerenciar conexão VPC, escolha a conexão VPC desejada e escolha Confirmar. Isso associa sua conexão VPC selecionada a esse ambiente.

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.

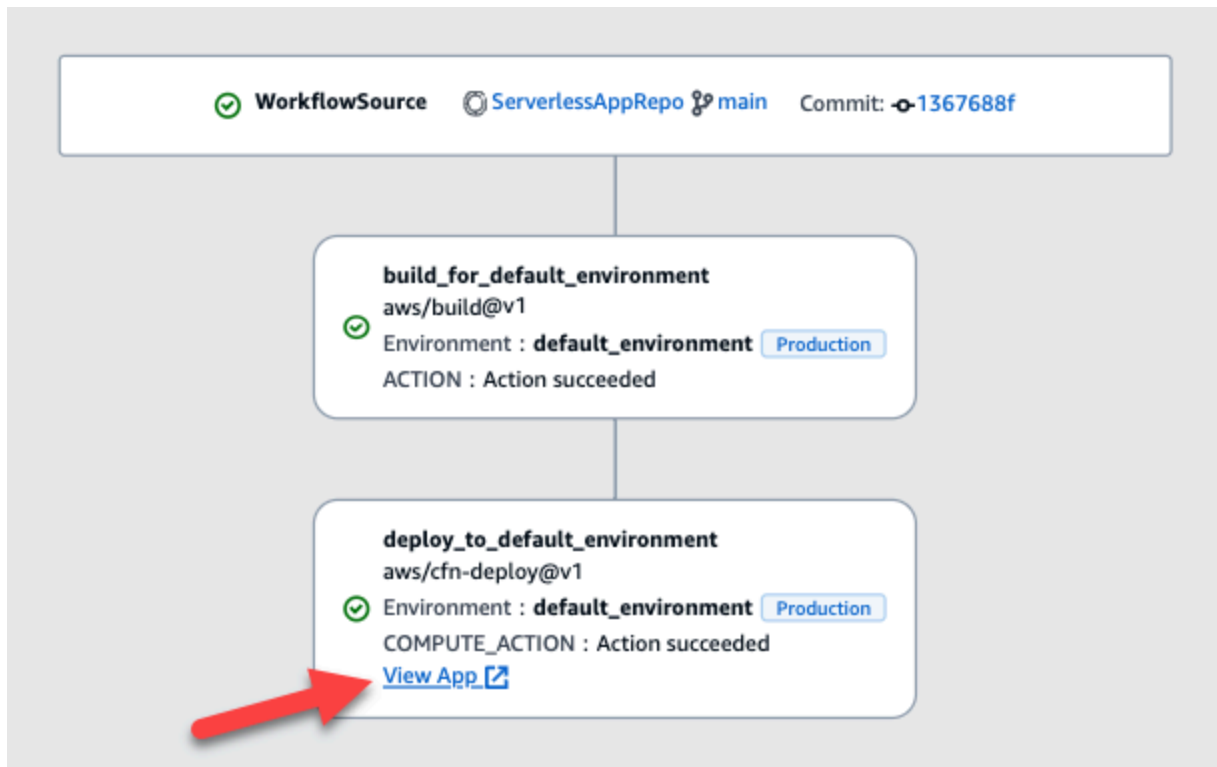
Para associar um Conta da AWS a um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Ambientes.
4. Escolha seu ambiente (por exemplo, Production).
5. Escolha a guia Propriedades do ambiente.
6. Escolha Associar Conta da AWS, escolha o desejado Conta da AWS e escolha Associar. Isso associa seu selecionado Conta da AWS a esse ambiente.

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

Exibindo a URL do aplicativo implantado 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 a URL do aplicativo Exibir 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 do aplicativo não é compatível com a ação Deploy to Amazon ECS.

Para ativar 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 não diferencia maiúsculas de minúsculas. Defina o valor da variável como `http` ou `https` URL do seu aplicativo implantado.

Note

Se você estiver atualizando uma variável de saída existente para incluir a `endpoint url` string ou `app url`, atualize todas as referências a essa variável para usar o novo nome da variável.

Para obter etapas detalhadas, consulte um dos procedimentos a seguir:

- [Para exibir o URL do aplicativo na ação “AWS CDK implantar”](#)
- [Para exibir o URL do aplicativo na ação “Implantar AWS CloudFormation pilha”](#)
- [Para exibir o URL do aplicativo em todas as outras ações](#)

Ao terminar de configurar o URL, verifique se ele aparece conforme o esperado seguindo estas instruções:

- [Para verificar se o URL do aplicativo 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`, ou `endpointurl`, com ou sem um traço (-), sublinhado (_) ou espaço (). A string não diferencia maiúsculas de minúsculas.
 - O valor deve ser o `https` URL `http` ou do seu aplicativo implantado.

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, 'my-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 [Para verificar se o URL do aplicativo foi adicionado](#).

Para exibir o URL do aplicativo na ação “Implantar AWS CloudFormation pilha”

1. Se você estiver usando a ação Deploy AWS 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 lógica) deve conter `appurl`, ou `endpointurl`, com ou sem um traço (-), sublinhado (_) ou espaço (). A string não diferencia maiúsculas de minúsculas.
 - O valor deve ser o https URL http ou do seu aplicativo implantado.

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 [Para verificar se o URL do aplicativo foi adicionado](#).

Para exibir o URL do aplicativo 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 Steps seção Inputs ou da ação no arquivo de definição do fluxo de trabalho. A variável deve ter as seguintes características:
 - Eles name devem conter `appurl`, ou `endpointurl`, com ou sem um traço (-), sublinhado (_) ou espaço (). A string não diferencia maiúsculas de minúsculas.
 - O valor deve ser o https URL http ou do seu aplicativo implantado.

Por exemplo, uma ação de criação pode ter a seguinte aparência:

```
Build-action:  
  Identifier: aws/build@v1  
  Inputs:  
  Variables:  
    - Name: APP-URL  
      Value: https://mycompany.myapp.com
```

... ou isso:

```
Actions:  
  Build:  
    Identifier: aws/build@v1  
    Configuration:  
      Steps:  
        - Run: APP-URL=https://mycompany.myapp.com
```

Para obter mais informações sobre a definição de variáveis de ambiente, consulte [Definindo uma variável](#).

2. Exporte a variável.

Por exemplo, sua ação de criação pode ter a seguinte aparência:

```
Build-action:
  ...
  Outputs:
    Variables:
      - APP-URL
```

Para obter informações sobre a exportação de variáveis, consulte [Exportar uma variável para que outras ações possam usá-la](#).

3. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
4. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.
5. Vá para [Para verificar se o URL do aplicativo foi adicionado](#).

Para verificar se o URL do aplicativo foi adicionado

- Inicie a execução de um fluxo de trabalho, caso ele não tenha sido iniciado automaticamente. A nova execução deve ter a URL do aplicativo exibida como um link clicável em seu diagrama de fluxo de trabalho. Para obter mais informações sobre como iniciar corridas, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

Removendo um alvo de implantação

Você pode remover um destino de implantação, como um cluster ou AWS CloudFormation pilha do Amazon ECS, da página Ambientes 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 alvos podem ficar obsoletos se:

- Você excluiu o fluxo de trabalho implantado no destino.

- Você alterou a pilha ou o cluster em que está implantando.
- Você excluiu a pilha ou o cluster do serviço Amazon ECS CloudFormation ou do Amazon ECS no AWS console.

Para remover um alvo de implantação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Ambientes.
4. Escolha o nome do ambiente que contém o destino de implantação que você deseja remover. Para obter informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).
5. Escolha a guia Alvos de implantação.
6. Escolha o botão de rádio ao lado do destino de implantação que você deseja remover.
7. Escolha Remover.

O alvo é removido da página.

Rastreando o 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. Considere os 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 de 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 commits.

- O status das implantações que incluem os commits.
- Os ambientes nos quais os commits são implantados com sucesso.
- O status de qualquer teste executado em relação aos commits em seu fluxo de trabalho de CI/CD.

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 [repositório do Bitbucket](#), [GitHub](#) ou do [Bitbucket](#).

Para rastrear o status da implantação por meio de confirmação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Controle de alterações.
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. Escolha Exibir alterações.

Uma lista de confirmações é exibida.

Para cada confirmação, você pode ver o seguinte:

- Confirme informações como ID, autor, mensagem e quando foram confirmadas. Para ter mais informações, consulte [Armazene e colabore no código com repositórios de origem no CodeCatalyst](#).
- O status das implantações em cada ambiente. Para ter mais informações, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst](#).
- Resultados do teste e da cobertura do código. Para ter mais informações, consulte [Testando com fluxos de trabalho](#).

Note

Os resultados da Análise de Composição de Software (SCA) não são exibidos.

6. (Opcional) Para ver 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, escolha Exibir detalhes dessa confirmação.

Visualizando os registros de implantação

Você pode visualizar registros relacionados a ações de implantação específicas para solucionar problemas na Amazon CodeCatalyst.

Você pode visualizar os registros a partir de um [fluxo de trabalho](#) ou de um [ambiente](#).

Para visualizar os registros de uma ação de implantação a partir 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Executações.
6. Escolha a execução do fluxo de trabalho que implantou seu aplicativo.
7. No diagrama do fluxo de trabalho, escolha a ação cujos registros você deseja visualizar.
8. Escolha a guia Registros e expanda as seções para revelar as mensagens de registro.
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 em AWS.

Para visualizar os registros de uma ação de implantação a partir de um ambiente

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

3. No painel de navegação, escolha CI/CD e, em seguida, escolha Ambientes.
4. Escolha o ambiente no qual seu aplicativo foi implantado.
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 registros você deseja visualizar.
7. Escolha a guia Registros e expanda as seções para revelar as mensagens de registro.
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 em AWS.

Visualização do status da implantação, confirmações e pull requests

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 pull requests.
- Informações específicas da ação, por exemplo, CloudFormation eventos e resultados.

Você pode visualizar as informações de implantação a partir de um [fluxo de trabalho](#), um [ambiente](#) ou uma [ação](#) de fluxo de trabalho.

Para visualizar as informações de implantação a partir de um fluxo de trabalho

- Acesse a execução do fluxo de trabalho que implantou seu aplicativo. Para obter instruções, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

Para visualizar as informações de implantação a partir de um ambiente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Ambientes.
4. Escolha o ambiente em que sua pilha foi implantada, por exemplo, Production

5. Escolha Atividade de implantação para visualizar o histórico de implantação de suas pilhas, o status das implantações (por exemplo, BEM-SUCEDIDA ou FALHA) e outras informações relacionadas à implantação.
6. Escolha 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 a partir de uma ação

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. No diagrama do fluxo de trabalho, escolha a ação do fluxo de trabalho que implantou seu aplicativo. Por exemplo, você pode escolher DeployCloudFormationStack.
6. Revise o conteúdo no painel direito para obter informações de implantação específicas da ação.

Criação de um fluxo de trabalho

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Use os procedimentos a seguir para criar um fluxo de trabalho em CodeCatalyst.

Para obter mais informações sobre fluxos de trabalho, consulte [Crie, teste e implante com fluxos de trabalho em CodeCatalyst](#).

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha Criar fluxo de trabalho.

A caixa de diálogo Criar fluxo de trabalho é exibida.


5. No campo Repositório de origem, escolha um repositório de origem no qual o arquivo de definição do fluxo de trabalho residirá. O arquivo será armazenado na `~/.codecatalyst/workflows/` pasta do repositório escolhido. Se não existir nenhum repositório de origem, [crie um](#).
6. No campo Ramificação, escolha uma ramificação na qual o arquivo de definição do fluxo de trabalho residirá.
7. Selecione Create (Criar).

A Amazon CodeCatalyst salva as informações do repositório e da filial na memória, mas o fluxo de trabalho ainda não está comprometido.

8. Escolha Visual.
9. Crie o fluxo de trabalho:
 - a. (Opcional) No diagrama do fluxo de trabalho, escolha a caixa Fonte e Acionadores. Um painel Acionadores é exibido. Escolha Adicionar gatilho para adicionar um gatilho. Para ter mais informações, consulte [Adicionar um gatilho push, pull ou scheduler](#).
 - b. Escolha + Ações (canto superior esquerdo). O catálogo de ações é exibido.
 - c. Escolha 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 ter mais informações, consulte [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#).
 - d. (Opcional) Escolha 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

[Configurando o comportamento de enfileiramento das execuções](#) e [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#).

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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 insira seu próprio nome.
 - b. Em Confirmar mensagem, deixe a mensagem padrão ou insira a sua.
 - c. Para Repositório e Filial, escolha 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 que você especificou anteriormente na caixa de diálogo Criar fluxo de trabalho. Você pode alterar o repositório e a ramificação agora, se quiser.

 Note

Depois de confirmar seu arquivo de definição de fluxo de trabalho, ele não pode ser associado a outro repositório ou ramificação, portanto, certifique-se de escolhê-los com cuidado.

- d. Escolha Confirmar para confirmar o arquivo de definição do fluxo de trabalho.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha Criar fluxo de trabalho.

A caixa de diálogo Criar fluxo de trabalho é exibida.

5. No campo Repositório de origem, escolha um repositório de origem no qual o arquivo de definição do fluxo de trabalho residirá. O arquivo será armazenado na `~/codecatalyst/`

workflows/ pasta do repositório escolhido. Se não existir nenhum repositório de origem, [crie um](#).

6. No campo Ramificação, escolha uma ramificação na qual o arquivo de definição do fluxo de trabalho residirá.
7. Selecione Create (Criar).

A Amazon CodeCatalyst salva as informações do repositório e da filial na memória, mas o fluxo de trabalho ainda não está comprometido.

8. Escolha YAML.
9. Crie o fluxo de trabalho:
 - a. (Opcional) Adicione um gatilho ao código YAML. Para ter mais informações, consulte [Adicionar um gatilho push, pull ou scheduler](#).
 - b. Escolha + Ações (canto superior esquerdo). O catálogo de ações é exibido.
 - c. Escolha 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 ter mais informações, consulte [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#).
 - d. (Opcional) Escolha 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 [Configurando o comportamento de enfileiramento das execuções](#) e [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#).
10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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 insira seu próprio nome.
 - b. Em Confirmar mensagem, deixe a mensagem padrão ou insira a sua.
 - c. Para Repositório e Filial, escolha 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 que você especificou anteriormente na caixa de diálogo Criar fluxo de trabalho. Você pode alterar o repositório e a ramificação agora, se quiser.

Note

Depois de confirmar seu arquivo de definição de fluxo de trabalho, ele não pode ser associado a outro repositório ou ramificação, portanto, certifique-se de escolhê-los com cuidado.

- d. Escolha Confirmar para confirmar o arquivo de definição do fluxo de trabalho.

Executar um fluxo de trabalho

Uma 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 manualmente ou iniciar uma 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 um commit para sua filial principal.

Você também pode interromper manualmente a execução de um fluxo de trabalho no meio do processamento se você o iniciou por engano.

Se várias execuções de fluxo de trabalho forem iniciadas ao mesmo tempo, você poderá configurar como deseja que essas execuções sejam enfileiradas. Você pode usar 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 foi iniciado, por quem foi iniciado e se ainda está em execução.

Tópicos

- [Iniciando um fluxo de trabalho executado manualmente](#)
- [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#)
- [Interromper uma execução de fluxo de trabalho](#)

- [Bloqueando a execução de um fluxo de trabalho](#)
- [Exigindo aprovações em execuções de fluxo de trabalho](#)
- [Configurando o comportamento de enfileiramento das execuções](#)
- [Armazenando arquivos em cache entre execuções de fluxo de trabalho](#)
- [Visualizando o status e os detalhes da execução do fluxo de trabalho](#)

Iniciando um fluxo de trabalho executado manualmente

Use o procedimento a seguir para iniciar a execução manual de um fluxo de trabalho.

Note

Você também pode iniciar a execução automática de um fluxo de trabalho [configurando um gatilho](#).

Para iniciar um fluxo de trabalho, execute manualmente

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do fluxo de trabalho que você deseja executar. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Executar.

Iniciando um fluxo de trabalho executado automaticamente com gatilhos

Um acionador 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.

Você pode usar três tipos de gatilho:

- Push — Um gatilho de envio de código faz com que a execução de um fluxo de trabalho seja iniciada sempre que um commit é enviado.
- Pull request — Um gatilho de pull request faz com que a execução de um fluxo de trabalho seja iniciada sempre que uma pull request é criada, revisada ou fechada.
- Agendamento — Um gatilho de agendamento faz com que a execução de um fluxo de trabalho comece em um cronograma definido por você. Considere usar um gatilho de agendamento para executar compilações noturnas de seu software para que a versão mais recente esteja pronta para seus desenvolvedores de software trabalharem na manhã seguinte.

Você pode usar gatilhos push, pull request e agendar sozinhos ou em combinação no mesmo fluxo de trabalho.

Tip

Para ver um gatilho em ação, inicie um projeto com um blueprint. A maioria dos blueprints contém um fluxo de trabalho com um gatilho. Procure a `Trigger` propriedade no arquivo de definição do fluxo de trabalho do blueprint. Para obter mais informações sobre esquemas, consulte [Criando um projeto com um blueprint](#).

Tópicos

- [Uma configuração de gatilho comum](#)
- [Acione considerações ao ramificar](#)
- [Adicionar um gatilho push, pull ou scheduler](#)
- [Exemplos de gatilhos](#)

Uma configuração de gatilho comum

Esta seção descreve como configurar acionadores para uma versão de software comum e uma estratégia de ramificação.

Estratégia de lançamento e ramificação de software:

- Você tem o código do aplicativo em um repositório de origem.
- Sua main ramificação contém um código finalizado que está sempre pronto para lançamento.

- Seus desenvolvedores de software fazem suas alterações em ramificações de recursos de forma `main` independente.
- Seus desenvolvedores de software [criam uma pull request](#) solicitando a fusão da ramificação de recursos `main` quando o recurso estiver pronto.

Você quer que essa pull request inicie automaticamente um fluxo de trabalho que cria e testa, mas não implanta, o aplicativo usando os arquivos na ramificação de recursos do desenvolvedor de software.

- Seus desenvolvedores de software verificam a compilação e os testes para garantir que tudo esteja bem. Em seguida, eles [mesclam o pull request](#) na `main` ramificação.

Você deseja que a mesclagem inicie automaticamente um fluxo de trabalho que cria e implanta o código do seu aplicativo.

Configuração proposta de fluxo de trabalho/gatilho:

Dada a estratégia de ramificação de software descrita anteriormente, talvez você queira usar dois fluxos de trabalho:

- O fluxo de trabalho 1 cria e testa seu aplicativo quando uma pull request é criada ou revisada.
- O Workflow 2 cria e implanta seu aplicativo quando uma pull request é mesclada.

O fluxo de trabalho 1 ficaria assim:

```
Triggers:
  - Type: PULLREQUEST
  Branches:
    - main
  Events:
    - OPEN
    - REVISION
Actions:
  BuildAction:
    instructions-for-building-the-app
  TestAction:
    instructions-for-test-the-app
```

O código de gatilho anterior inicia automaticamente a execução de um fluxo de trabalho sempre que um desenvolvedor de software cria uma pull request (ou [modifica uma](#)) solicitando a mesclagem

de sua ramificação de recursos com a `main` ramificação. CodeCatalyst inicia a execução de um fluxo de trabalho usando o código na ramificação de origem (ou seja, a ramificação de recursos do desenvolvedor). O fluxo de trabalho cria e implanta o aplicativo.

O fluxo de trabalho 2 ficaria assim:

```
Triggers:
  - Type: PUSH
  Branches:
    - main
Actions:
  BuildAction:
    instructions-for-building-the-app
  DeployAction:
    instructions-for-deploying-the-app
```

No código de gatilho anterior, quando `main` ocorre uma mesclagem, o `PUSH` gatilho é ativado. CodeCatalyst inicia a execução de um fluxo de trabalho usando o código na `main` ramificação (que agora inclui o código da pull request). O fluxo de trabalho cria e implanta o aplicativo.

Para obter instruções sobre como adicionar acionadores a um arquivo de definição de fluxo de trabalho, consulte [Adicionar um gatilho push, pull ou scheduler](#)

Para obter mais exemplos de acionadores e explicações adicionais, consulte [Exemplos de gatilhos](#)

Acione considerações ao ramificar

Esta seção descreve algumas das principais considerações ao configurar gatilhos que incluem ramificações.

- Consideração 1: Para acionadores de solicitação push e pull, se você for especificar uma ramificação, deverá especificar a ramificação de destino (ou 'para') na configuração do acionador. Nunca especifique a ramificação de origem (ou 'de').

No exemplo a seguir, um push de qualquer ramificação para `main` ativar o fluxo de trabalho.

```
Triggers:
  - Type: PUSH
  Branches:
    - main
```

No exemplo a seguir, uma pull request de qualquer ramificação `main` ativa o fluxo de trabalho.

Triggers:

- Type: PULLREQUEST
- Branches:
 - main
- Events:
 - OPEN
 - REVISION

- Consideração 2: Para acionadores push, depois que o fluxo de trabalho for ativado, o fluxo de trabalho será executado usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação de destino.
- Consideração 3: Para acionadores de pull request, depois que o fluxo de trabalho for ativado, o fluxo de trabalho será executado usando o arquivo de definição do fluxo de trabalho e os arquivos de origem na ramificação de origem (mesmo que você tenha especificado a ramificação de destino na configuração do acionador).
- Consideração 4: O mesmo gatilho em uma ramificação pode não ser executado em outra ramificação.

Considere o seguinte gatilho de pressão:

Triggers:

- Type: PUSH
- Branches:
 - main

Se o arquivo de definição do fluxo de trabalho que contém esse gatilho existir `main` e for `clonadotest`, o fluxo de trabalho nunca começará a usar automaticamente os arquivos em `test` (embora você possa iniciar o fluxo de trabalho manualmente se usar os arquivos em `test`). Analise as considerações 1 e 2 para entender por que o fluxo de trabalho nunca será executado automaticamente usando os arquivos em `test`.

Considere também o seguinte gatilho de pull request:

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 `test` ramificação a partir de `main`, o fluxo de trabalho será executado usando os arquivos em `test`.) Analise as considerações 1 e 3 para entender o porquê.

Adicionar um gatilho push, pull ou scheduler

Use as instruções a seguir para adicionar um gatilho push, pull ou agendamento ao seu fluxo de trabalho.

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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a caixa Fonte e Acionadores.
8. No painel de configuração, escolha 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. Você pode usar um dos seguintes valores:

- Push (editor visual) ou PUSH (editor YAML)

Um gatilho push 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).

- Pull request (editor visual) ou PULLREQUEST (editor YAML)

Um gatilho de pull request inicia a execução de um fluxo de trabalho quando uma pull request é aberta, atualizada ou fechada no seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação da qual você está extraíndo (ou seja, a ramificação de origem).

- Agenda (editor visual) ou SCHEDULE (editor YAML)

Um gatilho de agendamento inicia a execução do fluxo de trabalho em um cronograma definido 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 `Branches` propriedade (editor YAML).)

Ao configurar um gatilho de agendamento, siga estas diretrizes:

- Use apenas um gatilho de agendamento 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 ter mais informações, consulte [Expression](#).

Para ver exemplos, consulte [Exemplos de gatilhos](#).

Eventos para pull request

Esse campo só aparece se você selecionou o tipo de gatilho de pull request.

Especifique o tipo de eventos de pull request que iniciarão a execução de um fluxo de trabalho. A seguir estão os valores válidos:

- A solicitação pull é criada (editor visual) ou OPEN (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma pull request é criada.

- A solicitação pull está fechada (editor visual) ou CLOSED (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma pull request é fechada. O comportamento do CLOSED evento é complicado e é melhor compreendido por meio de um exemplo. Consulte [Exemplo: um gatilho com um puxão, galhos e um evento “FECHADO”](#) Para mais informações.

- Uma nova revisão é feita para pull request (editor visual) ou REVISION (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma revisão de uma pull request é criada. A primeira revisão é criada quando a pull request é criada. Depois disso, uma nova revisão é criada toda vez que alguém envia um novo commit para a ramificação de origem especificada na pull request. Se você incluir o REVISION evento em seu gatilho de pull request, poderá omitir o OPEN evento, pois REVISION é um superconjunto de. OPEN

Você pode especificar vários eventos no mesmo gatilho de pull request.

Para ver exemplos, consulte [Exemplos de gatilhos](#).

Schedule (Programação)

Esse campo só aparece se você selecionou o tipo de gatilho de agendamento.

Especifique a expressão cron que descreve quando você deseja que suas execuções de fluxo de trabalho agendadas ocorram.

As expressões Cron CodeCatalyst usam a seguinte sintaxe de seis campos, em que cada campo é separado por um espaço:

minutos horas days-of-monthmês days-of-weekano

Exemplos de expressões cron

Minutos	Horas	Dias do mês	Mês	Dias da semana	Ano	Significado
0	0	?	*	SEG-SEX	*	Executa um fluxo de trabalho à meia-noite (UTC +0) de segunda a sexta-feira.
0	2	*	*	?	*	Executa um fluxo de trabalho às 2:00 da manhã (UTC+0) todos os dias.
15	22	*	*	?	*	Executa 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 no dia inicial e 2h no 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 SCHEDULE gatilho.

- Coloque a expressão cron entre aspas duplas (") no editor YAML.
- Especifique a hora 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 um 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 obter exemplos de acionadores de agendamento, consulte. [Exemplos de gatilhos](#)

Galhos 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 acionador:

- Para um gatilho de pressão, 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 pull request, 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 agendamento, especifique as ramificações que contêm os arquivos que você deseja que sua execução programada use. 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:

- O galho para o qual você está empurrando (para gatilhos de pressão). Para ter mais informações, consulte [Exemplo: um simples gatilho de envio de código](#).
- A ramificação da qual você está retirando (para acionadores de pull request). Para ter mais informações, consulte [Exemplo: um simples gatilho de pull request](#).
- Todas as filiais (para acionadores de agendamento). Uma execução de fluxo de trabalho será iniciada por ramificação em seu repositório de origem. Para ter mais informações, consulte [Exemplo: um simples gatilho de agendamento](#).

Para obter mais informações sobre ramificações e gatilhos, consulte [Ação de considerações ao ramificar](#)

Para obter mais exemplos, consulte [Exemplos de gatilhos](#).

Arquivos alterados

Esse campo só aparece se você selecionou o tipo de gatilho de solicitação push ou 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 combinar nomes ou caminhos de arquivos.

Para ver exemplos, consulte [Exemplos de gatilhos](#).

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Adicione uma Triggers seção e propriedades subjacentes usando o exemplo a seguir como guia. Para obter mais informações, consulte o [Triggers](#) no [Definição de fluxo de trabalho YAML](#).

Um gatilho de envio de código pode ter a seguinte aparência:

```
Triggers:  
  - Type: PUSH  
    Branches:  
      - main
```

Um gatilho de pull request pode ter a seguinte aparência:

```
Triggers:  
  - Type: PULLREQUEST  
    Branches:  
      - main.*  
    Events:  
      - OPEN  
      - REVISION
```

- CLOSED

Um gatilho de agendamento 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 obter mais exemplos de expressões cron que você pode usar na Expression propriedade, consulte [Expression](#).

Para obter mais exemplos de acionadores push, pull request e agendamento, consulte [Exemplos de gatilhos](#)

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Exemplos de gatilhos

Os exemplos a seguir mostram como adicionar diferentes tipos de acionadores no arquivo de definição do fluxo de trabalho.

Tópicos

- [Exemplo: um simples gatilho de envio de código](#)
- [Exemplo: um simples gatilho “push to main”](#)
- [Exemplo: um simples gatilho de pull request](#)
- [Exemplo: um simples gatilho de agendamento](#)
- [Exemplo: um gatilho com um cronograma e ramificações](#)
- [Exemplo: um gatilho com um cronograma, um push e ramificações](#)
- [Exemplo: um gatilho com um puxão e galhos](#)
- [Exemplo: um gatilho com um puxão, galhos e um evento “FECHADO”](#)
- [Exemplo: um gatilho com um push, ramificações e arquivos](#)

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, 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 um commit para `main`, CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem ativados.

Como outro exemplo, se você enviar um commit para `feature-branch-123`, CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem ativados.

Triggers:

- Type: PUSH

Note

Se você quiser que a execução de um fluxo de trabalho inicie somente quando você enviar para `main`, consulte [Exemplo: um simples gatilho “push to main”](#).

Exemplo: um simples gatilho “push to main”

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que o código é enviado para a `main` ramificação, e somente para a ramificação, em seu `main` repositório de origem.

Triggers:

- Type: PUSH
- Branches:
- `main`

Exemplo: um simples gatilho de pull request

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma pull request é criada ou revisada no seu repositório de origem.

Quando esse gatilho é ativado, CodeCatalyst inicia a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem na ramificação da qual você está extraindo (ou seja, a ramificação de origem).

Por exemplo, se você criar uma pull request com uma ramificação de origem chamada `feature-123` e uma ramificação de destino chamada `main`, CodeCatalyst iniciará a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem ativados. `feature-123`

Triggers:

- Type: PULLREQUEST

Events:

- OPEN
- REVISION

Exemplo: um simples gatilho de agendamento

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-feira.

Quando esse gatilho é ativado, 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, 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 obter mais exemplos de expressões cron que você pode usar na `Expression` propriedade, consulte [Expression](#).

Exemplo: um gatilho com um cronograma e ramificações

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho às 18h15 (UTC +0) todos os dias.

Quando esse gatilho é ativado, CodeCatalyst inicia a execução de um fluxo de trabalho usando os arquivos na `main` ramificação e inicia execuções adicionais para cada ramificação que começa com `release-`.

Por exemplo, se você tiver ramificações chamadas `main`, `release-v1bugfix-1`, e `bugfix-2` no seu repositório de origem, 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 `bugfix-1` as `bugfix-1` filiais e.

Triggers:

- Type: SCHEDULE
- Expression: "15 18 * * ? *"
- Branches:
 - main
 - release\-.*

Para obter mais exemplos de expressões cron que você pode usar na `Expression` propriedade, consulte [Expression](#).

Exemplo: um gatilho com um cronograma, um push 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 `main` ramificação.
- A execução do fluxo de trabalho também é iniciada sempre que você envia uma confirmação para a `main` ramificação. 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 obter mais exemplos de expressões cron que você pode usar na `Expression` propriedade, consulte [Expression](#).

Exemplo: um gatilho com um puxão e galhos

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que alguém abre ou modifica uma pull request com uma ramificação de destino chamada `main`. Embora a ramificação especificada na `Triggers` configuração 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 extraíndo).

```
Triggers:  
- Type: PULLREQUEST  
  Branches:  
    - main  
  Events:  
    - OPEN  
    - REVISION
```

Exemplo: um gatilho com um puxão, galhos e um evento “FECHADO”

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma pull request é fechada em uma ramificação que começa com `main`.

Neste exemplo:

- Quando você fecha uma pull request 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 pull request, essas ramificações nunca terão a chance de entrar no `CLOSED` estado. Isso significa que as ramificações mescladas não ativarão o `CLOSED` gatilho do pull request. A única maneira de ativar o `CLOSED` gatilho nesse cenário é fechar a pull request sem mesclá-la.

```
Triggers:  
- Type: PULLREQUEST
```



```
Branches:  
- main.*  
Events:  
- CLOSED
```

Exemplo: um gatilho com um push, 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 `filename.txt` arquivo, ou em qualquer arquivo no `src` diretório, na `main` ramificação.

Quando esse gatilho é ativado, CodeCatalyst inicia a execução de um fluxo de trabalho usando o arquivo de definição do fluxo de trabalho e outros arquivos de origem na `main` ramificação.

```
Triggers:  
- Type: PUSH  
Branches:  
- main  
FilesChanged:  
- filename.txt  
- src\*.*
```

Interromper uma execução de fluxo de trabalho

Use o procedimento a seguir para interromper a execução de um fluxo de trabalho em andamento. Talvez você queira interromper uma corrida se ela foi 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 interrompida imediatamente.

Para interromper a execução 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, em seguida, escolha Fluxos de trabalho.
4. Em Fluxos de trabalho, escolha Execuções e escolha a execução em andamento na lista.
5. Escolha Parar.

Bloqueando a execução de um fluxo de trabalho

Um portão é um componente do fluxo de trabalho que você pode usar 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.

Você pode adicionar portas entre sequências de ações em um fluxo de trabalho ou antes da primeira ação (que é executada imediatamente após o download do código-fonte). Você também pode adicionar portas após a última ação, se precisar fazer isso.

Tipos de portão

Atualmente, a Amazon CodeCatalyst oferece suporte a um tipo de portão: o portão de aprovação. Para ter mais informações, consulte [Exigindo aprovações em execuções de fluxo de trabalho](#).

Posso configurar um portão para funcionar paralelamente a outra ação?

Não. Os portões só podem funcionar antes ou depois de uma ação. Para ter mais informações, consulte [Configurando dependências entre portas e ações](#).

Posso usar um portão para impedir que a execução de um fluxo de trabalho seja iniciada?

Sim, com qualificações.

Você pode impedir 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 iniciam e depois são bloqueados por um portão ainda contam para o número máximo de execuções simultâneas de fluxo de trabalho por cota de espaço e outras cotas. Para garantir que você não exceda as cotas de fluxo de trabalho, considere usar um gatilho de fluxo de trabalho para iniciar condicionalmente um fluxo de trabalho em vez de usar um portão. Considere também usar uma regra de aprovação de pull request em vez de um portão. Para obter mais informações sobre cotas, gatilhos e regras de aprovação de pull request, consulte [Cotas para fluxos de trabalho](#), e [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#) [Gerenciando requisitos para mesclar uma pull request com regras de aprovação](#)

Limitações dos portões

Os portões têm as seguintes limitações:

- Os portões não podem ser usados em conjunto com o recurso de compartilhamento de computação. Para obter mais informações sobre esse recurso, consulte [Compartilhamento de computação entre ações](#).
- Os portões não podem ser usados em grupos de ação. Para obter mais informações sobre grupos de ação, consulte [Agrupando ações em grupos de ação](#).

Tópicos

- [Adicionar um portão a um fluxo de trabalho](#)
- [Configurando dependências entre portas e ações](#)
- [Especificar a versão principal, secundária ou de patch de um portão](#)

Adicionar um portão a um fluxo de trabalho

Use as instruções a seguir para adicionar um portão a um fluxo de trabalho e depois configurá-lo.

Para adicionar e configurar um portão

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

3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. À esquerda, escolha Portões.
8. No catálogo de portões, pesquise um portão e escolha o sinal de adição (+) para adicionar o portão ao seu fluxo de trabalho.
9. Configure o portão. Escolha Visual para usar o editor visual ou YAML para usar o editor YAML. Para obter instruções detalhadas, consulte:
 - [Adicionar o portão de “Aprovação” a um fluxo de trabalho](#)
10. (Opcional) Escolha Validar para garantir que o código YAML seja válido.
11. Escolha Confirmar para confirmar suas alterações.

Configurando dependências entre portas e ações

Você pode configurar um portão para correr antes ou depois de uma ação, grupo de ação ou portão. Por exemplo, você pode configurar um Approval portão para correr antes de uma Deploy ação. Nesse caso, diz-se que a Deploy ação depende do Approval portão.

Para configurar dependências entre portas e ações, configure a propriedade Dependence da porta ou ação. Para obter instruções, consulte [Configurando dependências entre ações](#). As instruções referenciadas se referem às ações do fluxo de trabalho, mas se aplicam igualmente às portas.

Para obter um exemplo de como configurar a propriedade Depends on com um portão, consulte [Exemplo: configuração de um portão de “aprovação”](#).

Especificar a versão principal, secundária ou de patch de um portão

Por padrão, quando você adiciona um portão a um fluxo de trabalho, 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 obter instruções, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#). O tópico referenciado se refere às ações do fluxo de trabalho, mas se aplica igualmente às portas.

Exigindo aprovações em execuções de fluxo de trabalho

Você pode configurar a execução de um fluxo de trabalho para exigir uma aprovação antes de continuar. Para fazer isso, você deve adicionar um [portão](#) de aprovação ao fluxo de trabalho. Um portão de aprovação impede que um fluxo de trabalho continue até que um usuário ou conjunto de usuários envie uma ou mais aprovações no CodeCatalyst console. Depois que todas as aprovações são concedidas, o portão é “destrancado” e a execução do fluxo de trabalho pode ser retomada.

Use um portal de aprovação em seu fluxo de trabalho para dar às equipes de desenvolvimento, operações e liderança a chance de revisar suas alterações antes que elas sejam implantadas em um público mais amplo.

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: (Só se aplica se você estiver usando o modo de execução substituído.) A execução não deve ser substituída por uma execução posterior. Para ter mais informações, consulte [Como as aprovações de fluxo de trabalho funcionam com os modos de execução em fila, substituídos e paralelos?](#).

Se alguma das condições não for atendida, CodeCatalyst interrompe o fluxo de trabalho e define o status de execução como Falha (no caso das Condições 1 a 3) ou Substituído (no caso da Condição 4).

Quando usar o portão de “Aprovação”

Normalmente, você usaria uma porta de aprovação em um fluxo de trabalho que implanta aplicativos e outros recursos em um servidor de produção ou em qualquer ambiente em que os padrões de qualidade devam ser validados. Ao colocar a porta antes da implantação para produção, você dá 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 a função de colaborador ou administrador do projeto pode fornecer uma aprovação. Os usuários com a função de administrador do Space que pertencem ao espaço do seu projeto também podem fornecer uma aprovação.

Note

Usuários com a função de 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:

- CodeCatalyst Envie-lhes uma notificação do Slack. Para ter mais informações, consulte [Configurando notificações de aprovação](#).
- Acesse a página no CodeCatalyst console em que estão os botões Aprovar e Rejeitar e cole o URL dessa página em um aplicativo de e-mail ou mensagem endereçado aos aprovadores. Para obter mais informações sobre como navegar até essa página, consulte [Aprovando ou rejeitando a 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 ter mais informações, consulte [Posso usar um portão para impedir que a execução de um fluxo de trabalho seja iniciada?](#)

Como as aprovações de fluxo de trabalho funcionam com os modos de execução em fila, substituídos e paralelos?

[Ao usar o modo de execução em fila, substituído ou paralelo, o portão de aprovação funciona de forma semelhante às ações.](#) Sugerimos a leitura das [Sobre o modo de execução paralela](#) seções [Sobre o modo de execução em fila](#) [Sobre o modo de execução substituído](#), para se familiarizar com esses modos de execução. Depois de ter uma compreensão básica deles, retorne a esta seção para descobrir como esses modos de execução funcionam quando a porta 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 corridas ficarão na fila atrás da execução que está aguardando aprovação no portão. Quando esse portão é destrancado (ou seja, todas as aprovações foram concedidas), a próxima corrida na fila avança até o portão e aguarda as aprovações. Esse processo continua com as execuções em fila sendo processadas pelo portão. one-by-one [Figure 1](#) ilustra esse processo.
- Se você estiver usando o [modo de execução substituído](#), o comportamento é o mesmo do modo de execução em fila, exceto que, em vez de as corridas se acumularem na fila do portão, as execuções mais recentes substituem (substituem) as anteriores. Não há filas, e qualquer corrida que esteja aguardando aprovação no portão será cancelada e substituída por uma corrida mais recente. [Figure 2](#) ilustra esse processo.
- Se você estiver usando o [modo de execução paralela](#), as execuções começam em paralelo e não formam filas. Cada corrida é processada pelo portão imediatamente, pois não há nenhuma corrida à sua frente. [Figure 3](#) ilustra esse processo.

Figura 1: “Modo de execução em fila” e uma porta de aprovação

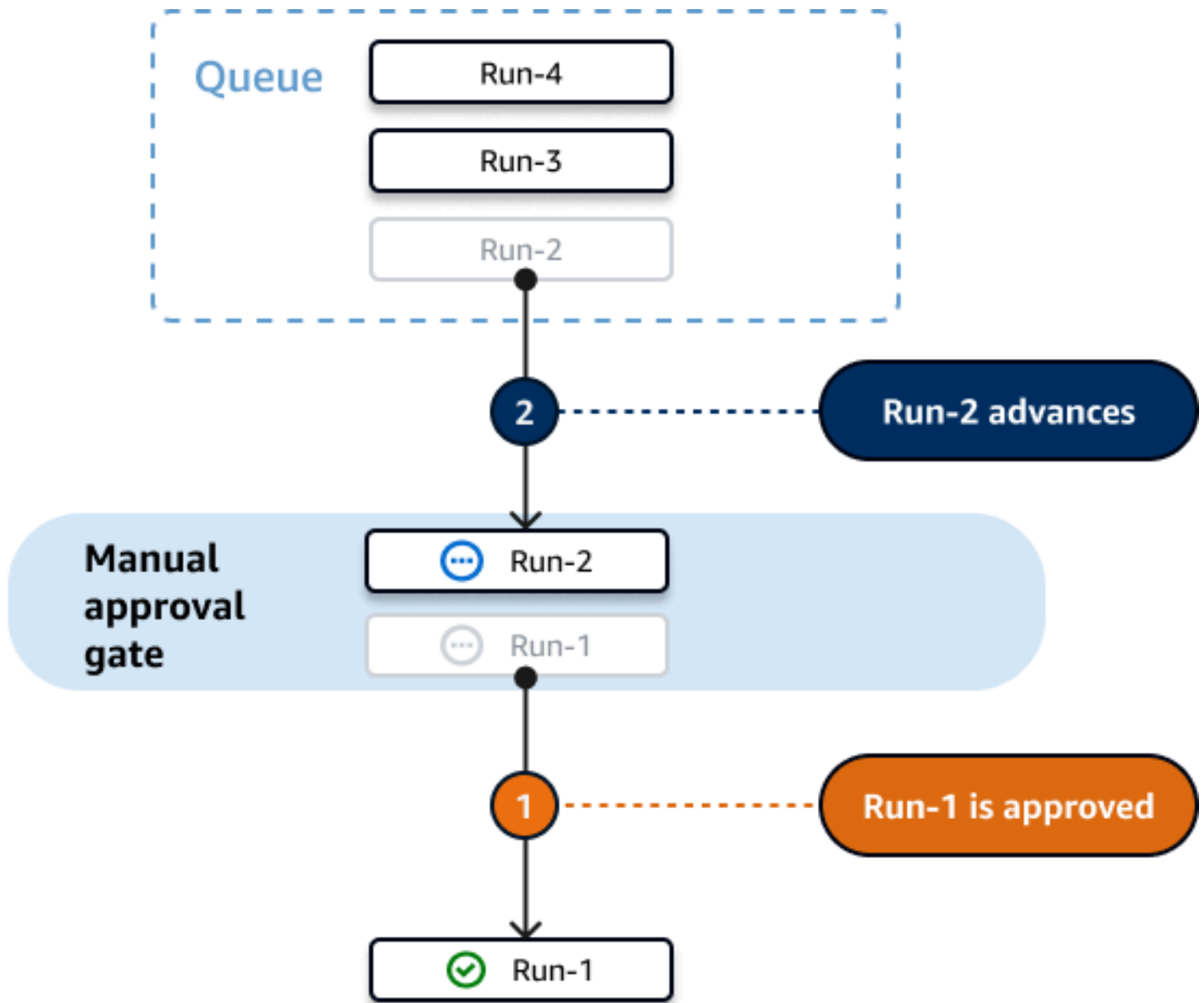


Figura 2: “Modo de execução substituído” e uma porta de aprovação

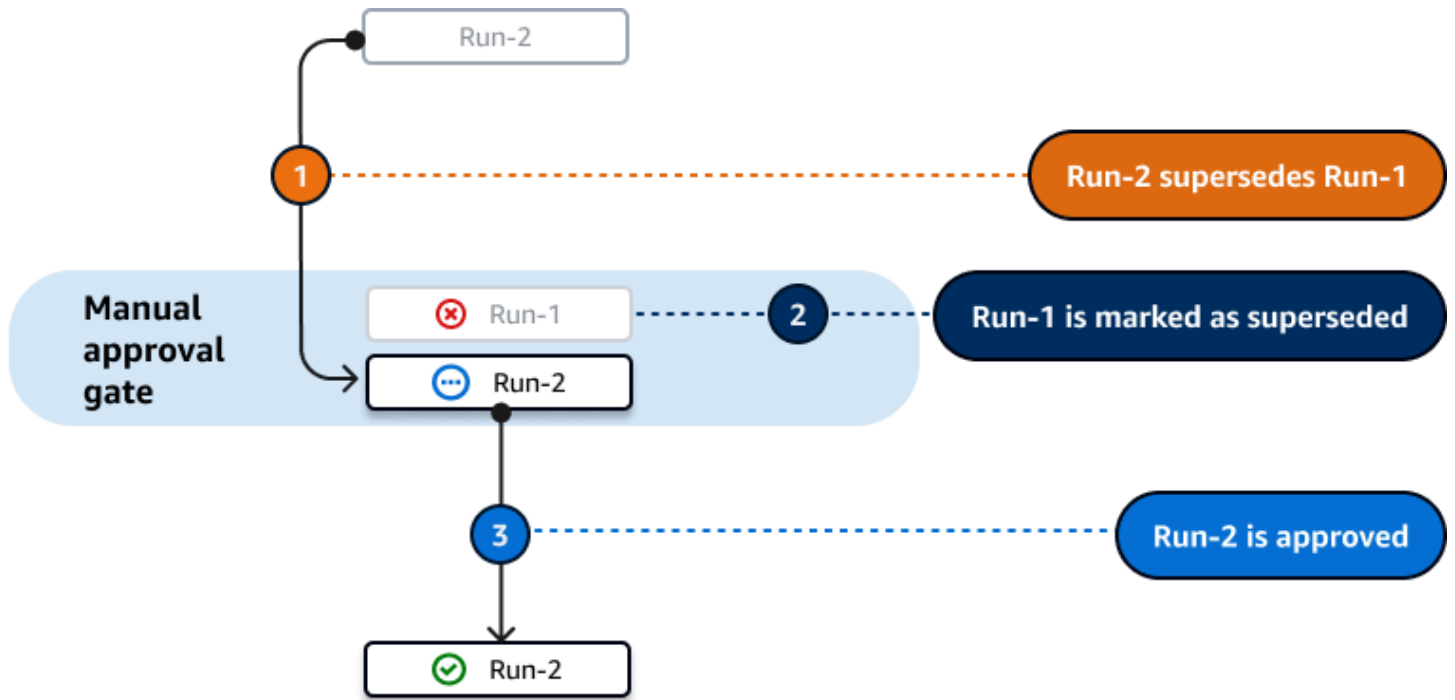
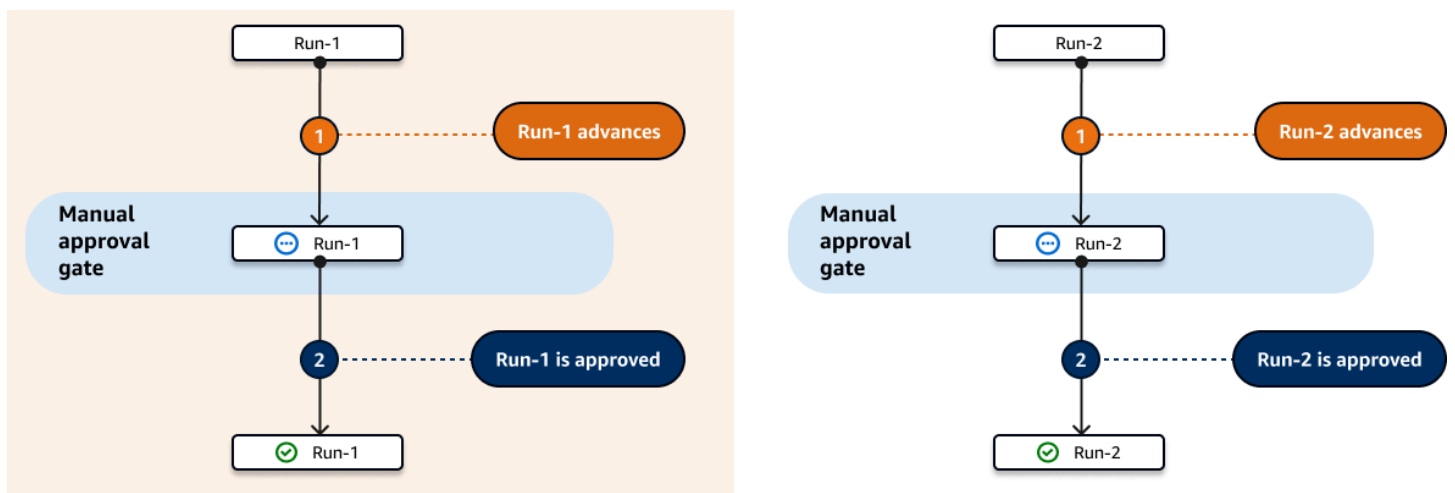


Figura 3: “Modo de execução paralela” e uma porta de aprovação



Tópicos

- [Exemplo: configuração de um portão de “aprovação”](#)
- [Adicionar o portão de “Aprovação” a um fluxo de trabalho](#)
- [Configurando notificações de aprovação](#)
- [Aprovando ou rejeitando a execução de um fluxo de trabalho](#)
- [Definição de YAML da porta de “aprovação”](#)

Exemplo: configuração de um portão de “aprovação”

O exemplo a seguir mostra como adicionar um portão de aprovação chamado `Approval_01` entre duas ações chamadas `Staging Production` e. A `Staging` ação acontece primeiro, o `Approval_01` portão depois e a `Production` ação por último. A `Production` ação só é executada se o `Approval_01` portão estiver destrancado. A `DependsOn` propriedade garante que as `Production` fases `StagingApproval_01`, e sejam executadas em ordem sequencial.

Para obter mais informações sobre o portão de aprovação, consulte [Exigindo 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:
      ...
```

Adicionar o portão de “Aprovação” a um fluxo de trabalho

Para configurar seu fluxo de trabalho para exigir uma aprovação, você deve adicionar 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 obter mais informações sobre esse portão, consulte [Exigindo aprovações em execuções de fluxo de trabalho](#).

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. No canto superior esquerdo, escolha Portões.
7. No catálogo Gates, em Aprovação, escolha o sinal de adição (+).
8. Escolha Entradas e, no campo Depende de, faça o seguinte.

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito 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 em seu 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, grupo de ação ou portão. Ele não pode ser configurado para ser executado em paralelo com outras ações, grupos de ação e portões.

Para obter mais informações sobre a funcionalidade Depende da funcionalidade, consulte [Configurando dependências entre portas 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 portões devem ser exclusivos no fluxo de trabalho. Os nomes dos portões são limitados a caracteres

alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes dos portões.

11. (Opcional) No campo Número de aprovações, faça o seguinte.

Especifique o número mínimo de aprovações necessárias para destravar o portão de aprovação. O mínimo é 1. O máximo é 2. Se omitido, o padrão é 1.

 Note

Se você quiser omitir a `ApprovalsRequired` propriedade, remova a `Configuration` seção do portão do arquivo de definição do fluxo de trabalho.

12. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
13. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Adicione uma `Approval` seção e propriedades subjacentes usando o exemplo a seguir como guia. Para obter mais informações, consulte o [Definição de YAML da porta de “aprovação”](#) no [Definição de fluxo de trabalho YAML](#).

```
Actions:
  MyApproval_01:
    Identifier: aws/approval@v1
    DependsOn:
```

```
- PreviousAction  
Configuration:  
ApprovalsRequired: 2
```

Para obter outro exemplo, consulte [Exemplo: configuração de um portão de “aprovação”](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Configurando notificações de aprovação

Você pode CodeCatalyst enviar 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 CodeCatalyst aprovações na qual 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 as notificações do Slack.

Antes de começar

Certifique-se de ter adicionado um portal de aprovação ao seu fluxo de trabalho. Para ter mais informações, consulte [Adicionar o portão de “Aprovação” a um fluxo de trabalho](#).

Para enviar notificações de aprovação do fluxo de trabalho para um canal do Slack

1. Configure CodeCatalyst com o Slack. Para ter mais informações, consulte [Começando a usar as notificações do Slack](#).
2. No CodeCatalyst projeto que contém o fluxo de trabalho que requer aprovação, ative as notificações, se elas ainda não estiverem habilitadas. Para ativar as notificações:
 - a. Navegue até seu projeto e, no painel de navegação, escolha Configurações do projeto.
 - b. Na parte superior, escolha Notificações.
 - c. Em Eventos de notificação, escolha Editar notificações.
 - d. Ative a aprovação do fluxo de trabalho pendente e escolha um canal do Slack para o qual 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, a execução do fluxo de trabalho rejeitada, a aprovação do fluxo de trabalho substituída e o tempo limite da aprovação do fluxo de trabalho. Ao lado de cada notificação, escolha o canal do Slack para o qual CodeCatalyst enviará a notificação.
- f. Escolha Salvar.

Aprovando ou rejeitando a 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 a partir de:

- o CodeCatalyst console
- um link fornecido por um membro da equipe
- uma 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 ter mais informações, consulte [Quem pode fornecer uma aprovação?](#).

Antes de começar


Certifique-se de ter adicionado um portal de aprovação ao seu fluxo de trabalho. Para ter mais informações, consulte [Adicionar o portão de “Aprovação” a um fluxo de trabalho](#).

Para aprovar ou rejeitar um fluxo de trabalho, execute a partir do console CodeCatalyst

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

5. No diagrama do fluxo de trabalho, escolha a caixa que representa a porta de aprovação.

Um painel lateral é exibido.

 Note

Nesse ponto, você pode enviar o URL dessa página para outros aprovadores, se quiser.

6. Em Revisar decisão, escolha 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 um fluxo de trabalho, execute a partir de um link fornecido por um membro da equipe

1. Escolha 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 em CodeCatalyst, se solicitado.

Você é redirecionado para a página de aprovação da execução do fluxo de trabalho.

3. Em Revisar decisão, escolha 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 um fluxo de trabalho, execute a partir de uma notificação automática do Slack

1. Certifique-se de que as notificações do Slack estejam configuradas. Consulte [Configurando notificações de aprovação](#).
2. No Slack, no canal para o qual a notificação de aprovação foi enviada, escolha o link na notificação de aprovação.
3. Faça login em CodeCatalyst, se solicitado.

Você é redirecionado para a página de execução do fluxo de trabalho.

4. No diagrama do fluxo de trabalho, escolha a porta de aprovação.
5. Em Revisar decisão, escolha 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.

Definição de YAML da porta de “aprovação”

A seguir está a definição YAML do portão de aprovação. Para saber como usar esse portão, consulte [Exigindo 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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 portões devem ser exclusivos no fluxo de trabalho. Os nomes dos portões são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes dos portões.

Padrão: Approval_nn.

UI correspondente: guia de configuração/nome do portão

Identifier

(Approval/Identifier)

(Obrigatório)

Identifica o portão. O portão de aprovação suporta a versão 1.0.0. Não altere essa propriedade, a menos que você queira encurtar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: aws/approval@v1.

UI correspondente: diagrama de fluxo de trabalho/ Approval_nn/aws/approval @v1 label

DependsOn

(Approval/DependsOn)

(Optional)

Especifique uma ação, um grupo de ação ou um portão que deve ser executado com êxito 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 em seu 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, grupo de ação ou portão. Ele não pode ser configurado para ser executado em paralelo com outras ações, grupos de ação e portões.

Para obter mais informações sobre a funcionalidade Dependência da funcionalidade, consulte [Configurando dependências entre portas e ações](#).

UI correspondente: guia Entradas/Depende de

Configuration

(*Approval*/Configuration)

(Optional)

Uma seção na qual você pode definir as propriedades de configuração do portão.

UI correspondente: guia Configuração

ApprovalsRequired

(*Approval*/Configuration/ApprovalsRequired)

(Optional)

Especifique o número mínimo de aprovações necessárias para destravar o portão de aprovação. O mínimo é 1. O máximo é 2. Se omitido, o padrão é 1.

Note

Se você quiser omitir a `ApprovalsRequired` propriedade, remova a `Configuration` seção do portão do arquivo de definição do fluxo de trabalho.

UI correspondente: guia de configuração/ Número de aprovações

Configurando o comportamento de enfileiramento das execuções

Por padrão, quando várias execuções de fluxo de trabalho ocorrem ao mesmo tempo, elas são colocadas em CodeCatalyst fila e as processam uma a uma, na ordem em que foram iniciadas. Você pode alterar esse comportamento padrão especificando um modo de execução. Existem alguns modos de execução:

- (Padrão) Modo de execução em fila — CodeCatalyst os processos são executados um por um

- Modo de execução substituído — CodeCatalyst os processos são executados um por um, com as execuções mais recentes ultrapassando as mais antigas
- Modo de execução paralela — CodeCatalyst os processos são executados em paralelo

Tópicos

- [Sobre o modo de execução em fila](#)
- [Sobre o modo de execução substituído](#)
- [Sobre o modo de execução paralela](#)
- [Configurando 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 entra em uma ação, a ação é bloqueada e nenhuma outra execução pode entrar. Quando a execução termina e sai da ação, a ação fica desbloqueada e pronta para a próxima execução.

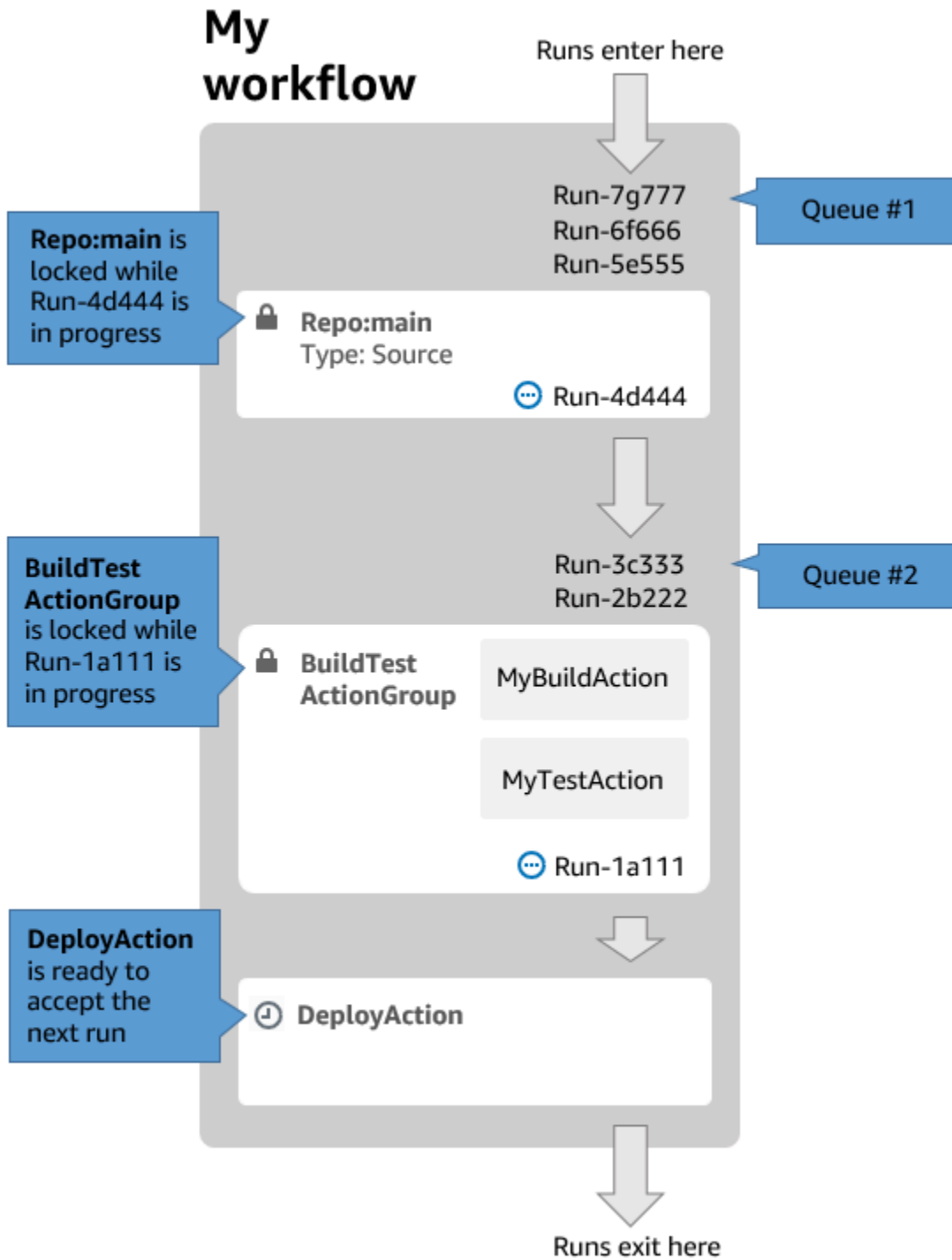
[Figure 1](#) ilustra um fluxo de trabalho configurado no modo de execução em fila. Mostra:

- 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 fonte de entrada (repo:main) e a. BuildTestActionGroup

Veja como as coisas acontecerão à medida que o fluxo de trabalho for executado e finalizado o processamento:

- Quando o Run-4d444 terminar de clonar o repositório de origem, ele sairá da fonte de entrada e entrará na fila atrás do Run-3c333. Em seguida, o Run-5e555 entrará na fonte de entrada.
- Quando o RUN-1A111 terminar de construir e testar, ele sairá da ação e entrará na BuildTestActionGroupação. DeployAction Em seguida, o Run-2b222 entrará na ação. BuildTestActionGroup

Figura 1: Um fluxo de trabalho configurado no “modo de execução em fila”



Use o modo de execução em fila se:

- Você deseja manter uma one-to-one relação entre recursos e execuções — esses recursos podem ser agrupados ao usar o modo substituído. 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 substituído em vez do modo em fila, seus recursos (e confirmações) serão agrupados na execução que substitui os outros.

- Você quer evitar condições de corrida e problemas inesperados que possam ocorrer ao usar o 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 implanta um novo código de aplicativo no cluster, fazendo com que os testes de Wang falhem ou testem o código errado. Como outro exemplo, você pode ter um destino que não tenha um mecanismo de bloqueio. Nesse caso, as duas execuções podem substituir as alterações uma da outra de maneiras inesperadas.
- Você quer limitar a carga nos recursos computacionais CodeCatalyst usados para processar suas execuções. Por exemplo, se você tiver três ações em seu fluxo de trabalho, poderá ter no máximo três execuções ocorrendo ao mesmo tempo. A imposição de um limite no número de execuções que podem ocorrer ao mesmo tempo torna a taxa de transferência de execução mais previsível.
- Você deseja restringir o número de solicitações feitas a serviços de terceiros pelo fluxo de trabalho. Por exemplo, seu 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 pull requests](#) que você pode fazer a um determinado número por hora por conta, e você será bloqueado se ultrapassar o limite. Usar o modo de execução em fila para diminuir a taxa de transferência de execução terá o efeito de gerar menos solicitações para o Docker Hub por hora, limitando assim o potencial de bloqueios e resultantes falhas de construção e execução.

Tamanho máximo da fila: 50

Notas sobre o tamanho máximo da fila:

- O tamanho máximo da fila se refere ao número máximo de execuções permitidas em todas as filas no fluxo de trabalho.
- Se uma fila tiver mais de 50 execuções, a 51ª e CodeCatalyst as subseqüentes serão descartadas.

Comportamento de falha:

Se uma execução deixar de responder enquanto estiver sendo processada por uma ação, as execuções por trás 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 substituído

O modo de execução substituído é o mesmo que o modo de execução em fila, exceto que:

- Se uma execução na fila alcançar outra execução na fila, a execução posterior substituirá (substituirá) a execução anterior, e a execução anterior será cancelada e marcada como 'substituída'.
- Como resultado do comportamento descrito no primeiro bullet, uma fila só pode incluir uma execução quando o modo de execução substituído é usado.

Usando o fluxo de trabalho [Figure 1](#) como guia, a aplicação do modo de execução substituído a esse fluxo de trabalho resultaria no seguinte:

- O Run-7G777 substituiria as outras duas execuções em sua fila e seria a única execução restante na Fila #1. A execução 6f666 e a execução -5e555 seriam canceladas.
- O Run-3C333 substituiria o Run-2B222 e seria a única execução restante na fila #2. O run-2b222 seria cancelado.

Use o modo de execução substituído se quiser:

- melhor taxa de transferência do que no modo em fila
- ainda menos solicitações em serviços de terceiros do que no modo em fila; isso é 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 corridas são independentes umas das outras e não espere que as outras sejam concluídas antes de começar. Não há filas e a taxa de transferência da execução é limitada 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 sua própria ramificação de recursos e implanta em destinos que não são compartilhados por outros usuários.

Important

Se você tiver um destino compartilhado no qual vários usuários possam implantar, como uma função 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 de fluxo de trabalho tentam alterar um recurso compartilhado ao mesmo tempo, levando a resultados imprevisíveis.

Número máximo de ensaios paralelos: 1000 por CodeCatalyst espaço

Configurando o modo de execução

Você pode definir o modo de execução como enfileirado, substituído ou paralelo. O padrão é em fila.

Quando você altera o modo de execução de enfileirado ou substituído para paralelo CodeCatalyst , cancela as execuções que estão na fila e permite que as execuções que estão sendo processadas atualmente por uma ação sejam concluídas antes de cancelá-las.

Quando você altera o modo de execução de paralelo para enfileirado ou substituído, permite que CodeCatalyst todas as execuções paralelas em execução no momento sejam concluídas. Todas as execuções que você iniciar após alterar o modo de execução para enfileirado ou substituído usam o novo modo.

Visual

Para alterar o modo de execução 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. No canto superior direito, escolha Propriedades do fluxo de trabalho.
7. Expanda Avançado e, em Modo de execução, escolha uma das seguintes opções:

- a. Em fila — consulte [Sobre o modo de execução em fila](#)
 - b. Substituído — consulte [Sobre o modo de execução substituído](#)
 - c. Paralelo — veja [Sobre o modo de execução paralela](#)
8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
 9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para alterar o modo de execução 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Adicione a RunMode propriedade, assim:

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED|SUPERSEDED|PARALLEL
```

Para obter mais informações, consulte a descrição da RunMode propriedade na [Propriedades de nível superior](#) seção do [Definição de fluxo de trabalho YAML](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Armazenando arquivos em cache entre execuções de fluxo de trabalho

Quando o armazenamento em cache de arquivos está ativado, as ações de criação 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 [computação EC2](#).

Tópicos

- [Sobre o armazenamento em cache de arquivos](#)
- [Criando um cache](#)
- [Restrições de armazenamento em cache de arquivos](#)

Sobre o armazenamento em cache de arquivos

O armazenamento em cache de arquivos permite que você organize seus dados em vários caches, cada um referenciado na propriedade. `FileCaching` Cada cache salva um diretório especificado por um determinado caminho. O diretório especificado será restaurado em futuras execuções do fluxo de trabalho. A seguir está um exemplo de trecho YAML para armazenamento em cache com vários caches chamados `e`, `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 frotas provisionadas ou máquinas sob demanda encontram uma perda de cache em um cache local, as dependências serão 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 afetam construções 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 do que `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 nova ramificação é criada ou quando um novo cache é referenciado 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 alternativos são caches alternativos e 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 abaixo `FileCaching` do nome da propriedade e, se não for encontrada, `RestoreKeys` avaliada. Se houver uma falha de cache tanto no nome da propriedade quanto em tudo `RestoreKeys`, o fluxo de trabalho continuará sendo executado, pois o armazenamento em cache é o melhor esforço e não é garantido.

Criando um cache

Você pode usar as instruções a seguir para adicionar um cache ao seu fluxo de trabalho.

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação em que você deseja adicionar seu cache.
8. Escolher configuração.
9. Em Cache de arquivos - opcional, escolha Adicionar cache e insira as informações nos campos, da seguinte forma:

Chave

Especifique o nome da sua propriedade de cache principal. Os nomes das propriedades de cache devem ser exclusivos em seu fluxo de trabalho. Cada ação pode ter até cinco entradas `FileCaching`.

Path

Especifique o caminho associado para seu cache.

Chaves de restauração - opcional

Especifique a chave de restauração a ser usada como alternativa quando a propriedade primária do cache não puder ser encontrada. Os nomes das chaves de restauração devem ser exclusivos em seu fluxo de trabalho. Cada cache pode ter até cinco entradas `RestoreKeys`.

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Restrições de armazenamento em cache de arquivos

A seguir estão as restrições para o nome da propriedade e: RestoreKeys

- Os nomes devem ser exclusivos em um fluxo de trabalho.
- Os nomes são limitados a caracteres alfanuméricos (A-Z, a-z, 0-9), hífen (-) e sublinhados (_).
- Os nomes podem ter até 180 caracteres.
- Cada ação pode ter até cinco caches. FileCaching
- Cada cache pode ter até cinco entradasRestoreKeys.

A seguir estão as restrições dos caminhos:

- Asteriscos (*) não são permitidos.
- Os caminhos podem ter até 255 caracteres.

Visualizando o status e os detalhes da execução do fluxo de trabalho

Você pode visualizar o status e os detalhes de uma única execução do fluxo de trabalho ou de várias execuções ao mesmo tempo.

Para obter uma lista dos 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 do fluxo de trabalho. Para ter mais informações, consulte [Visualizando o status do fluxo de trabalho](#).

Tópicos

- [Visualizando o status e os detalhes de uma única execução](#)
- [Visualizando o status e os detalhes de todas as execuções em seu projeto](#)
- [Visualizando o status e os detalhes de todas as execuções de um fluxo de trabalho específico](#)
- [Visualizando execuções de um fluxo de trabalho no diagrama do fluxo de trabalho](#)

Visualizando o status e os detalhes de uma única execução

Talvez você queira ver o status e os detalhes de uma única execução de fluxo de trabalho para verificar se ela foi bem-sucedida, para ver em que momento foi concluída ou para ver quem ou o que a iniciou.


Para ver o status e os detalhes de uma única execução

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Abaixo do nome do fluxo de trabalho, escolha Executar.
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 exibição também mostra o repositório de origem e a ramificação usados durante a execução.

No diagrama do fluxo de trabalho, escolha uma ação para ver detalhes como registros, relatórios e saídas geradas pela ação durante a execução. As informações mostradas dependem do tipo de ação selecionado. Para obter mais informações sobre a visualização de registros de criação ou implantação, consulte [Visualizando os resultados de uma ação de construção](#) ou [Visualizando os registros 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 [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).
- 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 obter mais informações sobre variáveis, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

 Note

Se o fluxo de trabalho principal da execução foi excluído, uma mensagem indicando esse fato aparecerá na parte superior da página de detalhes da execução.

Visualizando o status e os detalhes de todas as execuções em seu projeto

Talvez você queira ver o status e os 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 ver o status e os detalhes de todas as execuções em seu projeto


1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Em Fluxos de trabalho, escolha 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 da execução — O identificador exclusivo da execução. Escolha o link da ID da execução para ver informações detalhadas sobre a execução.
- Status — O status de processamento da execução do fluxo de trabalho. Para obter mais informações sobre estados de execução, consulte [Estados de execução do fluxo de trabalho](#).
- Acionador — A pessoa, o commit, o pull request (PR) ou o agendamento que iniciou a execução do fluxo de trabalho. Para ter mais informações, consulte [Iniciando um fluxo de trabalho executado automaticamente com 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 — A 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 — A hora em que a execução do fluxo de trabalho terminou.

Visualizando o status e os detalhes de todas as execuções de um fluxo de trabalho específico

Talvez você queira ver 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 os detalhes de todas as execuções de um fluxo de trabalho específico

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Abaixo do nome do fluxo de trabalho, escolha Executar.

As execuções associadas ao fluxo de trabalho escolhido são exibidas.

A página está dividida em duas seções:

- Execuções ativas — Exibe as execuções que estão em andamento. Essas corridas 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 obter mais informações sobre estados de execução, consulte [Estados de execução do fluxo de trabalho](#).

Visualizando 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 à medida que 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 esperando em uma fila.

Para ver o status de várias execuções à medida que elas progridem juntas em um fluxo de trabalho

Note

Esse procedimento só se aplica se o fluxo de trabalho estiver usando o modo de execução em fila ou substituído. Para ter mais informações, consulte [Configurando o comportamento de enfileiramento das execuções](#).

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do fluxo de trabalho que contém as execuções que você deseja visualizar. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.

Note

Verifique se você está vendo uma página de fluxo de trabalho e não uma página de execução.

5. Escolha 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 atualmente em andamento no fluxo de trabalho e as execuções mais recentes que foram concluídas. Mais especificamente:
- As execuções que aparecem na parte superior, antes de Sources, estão na fila e aguardam o início.
 - As execuções que aparecem entre as ações são colocadas em fila e aguardam para serem processadas pela próxima ação.
 - As execuções que aparecem em uma ação são 1. 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).

Configurando as ações que um fluxo de trabalho executa

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

Tópicos

- [Tipos de ação](#)
- [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#)
- [Removendo uma ação de um fluxo de trabalho](#)
- [Desenvolvendo uma ação personalizada](#)
- [Agrupando ações em grupos de ação](#)
- [Configurando ações para depender de outras ações](#)
- [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#)
- [Especificar a versão principal, secundária ou de patch de uma ação](#)
- [Determinar quais versões de uma ação estão disponíveis](#)
- [Visualizando o código-fonte de uma ação](#)
- [Integrando GitHub ações em um fluxo de trabalho](#)

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 ter mais informações, consulte [Adicionando a ação de construção](#).

- Teste

Essa ação executa testes de integração e sistema em relação ao seu aplicativo ou artefatos. Para ter mais informações, consulte [Adicionando a ação de teste](#).

- Publicação do Amazon S3

Essa ação copia os artefatos do seu aplicativo para um bucket do Amazon S3. Para ter 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 ter 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 ter mais informações, consulte [Implantando um AWS Cloud Development Kit \(AWS CDK\) aplicativo com um fluxo de trabalho](#).

- AWS Lambda invocar

Essa ação invoca uma AWS Lambda função. Para ter mais informações, consulte [Invocando uma AWS Lambda função usando um fluxo de trabalho](#).

- GitHub Ações

Essa ação é uma CodeCatalystação que permite executar GitHub ações em um CodeCatalyst fluxo de trabalho. Para ter mais informações, consulte [Invocando uma AWS Lambda função usando um fluxo de trabalho](#).

- Implante a AWS CloudFormation pilha

Essa ação implanta AWS CloudFormation pilhas. Para ter mais informações, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#).

- Implemente no Amazon ECS

Essa ação registra uma definição de tarefa do Amazon ECS e a implanta em um serviço do Amazon ECS. Para ter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) com um fluxo de trabalho](#).

- Implemente no cluster Kubernetes

Essa ação implanta um aplicativo em um cluster Kubernetes. Para ter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#).

- Renderizar a definição da 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 ter mais informações, consulte [Modificar um arquivo de definição de tarefas do Amazon ECS usando um fluxo de trabalho](#).

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 [Adicionar uma ação a um CodeCatalyst 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

Essa ação implanta um aplicativo 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 seu aplicativo 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.

- Publique 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.

- Empurre para o Amazon ECR

Essa ação cria 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 verificação de código.

- Edição Comunitária do Terraform

Essa ação executa a Terraform Community Edition plan e apply as operações.

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 [Adicionar uma ação a um CodeCatalyst 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 ter mais informações, consulte [Integrando GitHub ações em um fluxo de trabalho](#).

Ações de terceiros

Uma ação de terceiros é uma ação criada por 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 para ações de terceiros está disponível no readme de cada ação. Documentação adicional também pode ser fornecida pelo fornecedor terceirizado.

Para obter informações sobre como adicionar uma ação de terceiros a um fluxo de trabalho e visualizar seu readme, consulte [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#).

Adicionar uma ação a um CodeCatalyst 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. No canto superior esquerdo, escolha + Ações. O catálogo de ações é exibido.
7. Na lista suspensa, faça o seguinte:
 - 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 rótulo por **fornecedor**, em que *fornecedor é o nome do fornecedor* terceirizado.
 - Escolha GitHubver uma [lista organizada de GitHub ações](#).
8. No catálogo de ações, pesquise uma ação e, em seguida, faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao seu fluxo de trabalho.
 - Escolha o nome da ação para ver seu readme.
9. Configure a ação. Escolha Visual para usar o editor visual ou YAML para usar o editor YAML. Para obter instruções detalhadas, consulte os links a seguir.

Para obter instruções sobre como adicionar [CodeCatalystações](#), consulte:

- [Adicionando a ação de construção](#)

- [Adicionando a ação de teste](#)
- [Adicionando a ação “Implantar no Amazon ECS”](#)
- [Adicionando a ação “Implantar no cluster Kubernetes”](#)
- [Adicionando a ação “Deploy AWS CloudFormation stack”](#)
- [Adicionando a ação “AWS CDK implantar”](#)
- [Adicionando a ação “AWS CDK bootstrap”](#)
- [Adicionando a ação “Amazon S3 publish”](#)
- [Adicionando a AWS Lambda ação “invocar”](#)
- [Adicionando a ação “Render Amazon ECS task definition”](#)

Para obter instruções sobre como adicionar [ações do CodeCatalyst Labs](#), consulte:

- A ação é readme. Você pode encontrar o readme escolhendo o nome da ação no catálogo de ações.

Para obter instruções sobre como adicionar [GitHub ações](#), consulte:

- [Integrando GitHub ações em um fluxo de trabalho](#)

Para obter instruções sobre como adicionar [ações de terceiros](#), consulte:

- A ação é readme. Você pode encontrar o readme escolhendo o nome da ação no catálogo de ações.

10. (Opcional) Escolha Validar para garantir que o código YAML seja válido.

11. Escolha Confirmar para confirmar suas alterações.

Removendo uma ação de um fluxo de trabalho

Use as instruções a seguir para remover uma ação de um fluxo de trabalho.

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, na ação que você deseja remover, escolha o ícone de elipse vertical e escolha Remover.
8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Desenvolvendo 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.

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, incluindo um arquivo de definição de ação (`action.yml`) que você pode atualizar com seu próprio código.
4. Inicialize o código de ação para obter as ferramentas e bibliotecas necessárias para criar, 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](#).

Agrupando ações em grupos de ação

Um grupo de ações contém uma ou mais ações. O agrupamento de ações em grupos de ações ajuda a manter seu fluxo de trabalho organizado e também permite configurar dependências entre grupos diferentes.

Note

Você não pode agrupar grupos de ação em outros grupos de ação ou ações.

Tópicos

-
- [Exemplo: definição de dois grupos de ação](#)

Use as instruções a seguir para definir um grupo de ação.

Visual

Não disponível. Escolha YAML para ver as instruções do 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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: definição de dois grupos de ação](#). Para obter mais informações, consulte a descrição [Ações](#) da `action-group-name` propriedade no [Definição de fluxo de trabalho YAML](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Exemplo: definição de dois grupos de ação

O exemplo a seguir mostra como definir dois grupos de ação: BuildAndTest Deploy e. O BuildAndTest grupo inclui duas ações (BuildTest), e o Deploy grupo também inclui duas ações (DeployCloudFormationStackeDeployToECS).

```
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:
            ...
```

Configurando ações para depender de outras ações

Por padrão, quando você adiciona ações a um fluxo de trabalho, elas são adicionadas lado a lado no [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), você deve configurar dependências entre elas. Por exemplo, você pode configurar uma Test ação para depender da ação para que a Build ação de teste seja executada após a ação de compilação.

Você pode 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 o [Diretrizes para configurar dependências](#) para garantir que sua configuração de dependência esteja em conformidade com a sintaxe YAML do fluxo de trabalho.

Tópicos

- [Configurando dependências entre ações](#)
- [Diretrizes para configurar dependências](#)
- [Exemplos de como configurar dependências entre ações](#)

Configurando dependências entre ações

Use as instruções a seguir para configurar dependências entre ações em um fluxo de trabalho.

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação que dependerá de outra ação.
8. Escolha a guia Entradas.
9. Em Dependente de - opcional, faça o seguinte:

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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 obter diretrizes gerais, consulte [Diretrizes para configurar dependências](#). Para obter mais informações, consulte a descrição da DependsOn propriedade em [Definição de fluxo de trabalho YAML](#) Para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Diretrizes para configurar dependências

Ao configurar dependências, siga estas diretrizes:

- Se uma ação estiver dentro de um grupo, essa ação 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 do YAML, mas não em um nível diferente.

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: configuração de uma dependência simples](#)
- [Exemplo: configuração de um grupo de ações para depender de uma ação](#)
- [Exemplo: configuração de um grupo de ação para depender de outro grupo de ação](#)
- [Exemplo: configuração de um grupo de ações para depender de várias ações](#)

Exemplo: configuração de uma dependência simples

O exemplo a seguir mostra como configurar uma Test ação para depender da Build ação usando a DependsOn propriedade.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      ...
  Test:
    DependsOn:
      - Build
    Identifier: aws/managed-test@v1
    Configuration:
      ...
```

Exemplo: configuração de um grupo de ações para depender de uma ação

O exemplo a seguir mostra como configurar um grupo de DeployGroup ações para depender da FirstAction ação. Observe que a ação e o grupo de ação 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: configuração de um grupo de ação para depender de outro grupo de ação

O exemplo a seguir mostra como configurar um grupo de DeployGroup ação para depender do grupo de BuildAndTestGroup ação. Observe que os grupos de ação estão no mesmo nível.

```

Actions:
  BuildAndTestGroup: # Action group 1
    Actions:
      BuildAction:
      ...
      TestAction:
      ...
  DeployGroup: #Action group 2
DependsOn:
  - BuildAndTestGroup
Actions:
  DeployAction1:
  ...
  DeployAction2:
  ...

```

Exemplo: configuração de um grupo de ações para depender de várias ações

O exemplo a seguir mostra como configurar um grupo de DeployGroup ações para depender da SecondAction ação, da ação e do grupo de BuildAndTestGroup ações. FirstAction Observe que DeployGroup está no mesmo nível de FirstActionSecondAction, BuildAndTestGroup e.

```

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:
      ...

```

Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos

Um artefato é a saída de uma ação de fluxo de trabalho e normalmente consiste em uma pasta ou arquivo 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 `sam-template.yml` arquivo, mas deseja que uma ação de implantação a use. Nesse cenário, você usaria um artefato para permitir que a ação de criação compartilhasse o `sam-template.yml` arquivo com a ação de implantação. O código pode ser mais ou menos assim:

```

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 build (BuildAction) gera um `sam-template.yml` arquivo e o adiciona a um artefato de saída chamado MYARTIFACT. Uma ação de implantação subsequente (DeployAction) é especificada MYARTIFACT como uma entrada, dando a ela acesso ao `sam-template.yml` arquivo.

Posso compartilhar artefatos sem especificá-los como saídas e entradas?

Sim, você pode compartilhar artefatos entre ações sem especificá-los nas `Inputs` seções `Outputs` e do código YAML de suas ações. Para fazer isso, você deve ativar o compartilhamento de computação. Para obter 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 `Inputs` das seções `Outputs` e, o recurso tem limitações que você deve conhecer antes de ativá-lo. Para obter informações sobre essas limitações, consulte [Considerações sobre o compartilhamento de computação](#).

Posso compartilhar artefatos entre fluxos de trabalho?

Não, você não pode compartilhar artefatos entre fluxos de trabalho diferentes; no entanto, você pode compartilhar artefatos entre ações dentro do mesmo fluxo de trabalho.

Tópicos

- [Definindo um artefato de saída](#)
- [Definindo um artefato de entrada](#)
- [Fazendo referência a arquivos em um artefato](#)
- [Baixando artefatos](#)

- [Exemplos de artefatos](#)

Definindo um artefato de saída

Use as instruções a seguir para definir um artefato que você deseja que uma ação produza. Esse artefato então fica disponível para uso de outras ações.

Note

Nem todas as ações oferecem suporte a artefatos de saída. Para determinar se sua ação os suporta, siga as instruções do editor visual a seguir e veja se a ação inclui um botão de artefatos de saída na guia Saídas. Se sim, há suporte para artefatos de saída.

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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação que produzirá o artefato.
8. Escolha a guia Outputs.
9. Em Artefatos, escolha Adicionar artefato.
10. Escolha Adicionar artefato e insira as informações nos campos, da seguinte maneira.

Nome do artefato de construção

Especifique o nome de um artefato gerado pela ação. Os nomes dos artefatos devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (_). Espaços, hífen (-) e outros caracteres especiais não são permitidos.

Você não pode usar aspas para ativar espaços, hífen e outros caracteres especiais nos nomes dos artefatos de saída.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Arquivos produzidos por build

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 seu 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. Você pode usar padrões globais 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 obter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou fonte encontrá-lo. Para obter mais informações, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#).

11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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 obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Definindo um artefato de entrada

Se quiser usar um artefato gerado por outra ação, você deve especificá-lo como uma entrada para a ação atual. Talvez você possa especificar vários artefatos como entrada — isso depende da ação. Para obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) 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, verifique se você gerou o artefato da outra ação. Para ter mais informações, consulte [Definindo 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação em que você deseja especificar um artefato como entrada.

- Escolha Entradas.
- 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

 Note

Se a lista suspensa Artefacts - opcional não estiver disponível (editor visual) ou se você receber erros ao validar seu YAML (editor YAML), talvez seja porque a ação suporta apenas uma entrada. Nesse caso, tente remover a entrada da fonte.

- (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para especificar um artefato como entrada para uma ação (editor YAML)

- Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
- Selecione o projeto.
- No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
- Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
- Selecione a opção Editar.
- Escolha YAML.
- 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](#).

- (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Fazendo referência a arquivos em um artefato

Se você tiver um arquivo que reside em um artefato e precisar se referir a esse arquivo em uma das ações do fluxo de trabalho, conclua o procedimento a seguir.

Note

Consulte também [Fazendo referência a arquivos em um repositório de origem](#).


Para referenciar arquivos em um artefato

- Na ação em que você deseja referenciar um arquivo, adicione um código semelhante ao seguinte:

```
Actions:  
  My-action:  
    Inputs:  
      Sources:  
        - WorkflowSource  
      Artifacts:  
        - artifact-name  
    Configuration:  
      Steps:  
        - run: cd $CATALYST_SOURCE_DIR_artifact-name/build-output && cat file.txt
```

No código anterior, a ação procura no build-output diretório do artefato *nome-do-artefato* para encontrar e exibir o arquivo. `file.txt`

Para obter mais exemplos, consulte [Exemplos de artefatos](#).

 Note

Você pode omitir o `$(CATALYST_SOURCE_DIR)_artifact-name`/ prefixo dependendo de como você configurou sua ação. Para obter mais informações, consulte as diretrizes a seguir.

Orientação sobre como se referir às variáveis:

- Se sua ação incluir somente um item abaixo Inputs (por exemplo, inclui um artefato de entrada e nenhuma fonte), você poderá omitir o prefixo e especificar apenas o caminho do arquivo em relação à raiz do artefato.
- Você também pode omitir o prefixo se o arquivo residir na entrada primária. A entrada primária é `aWorkflowSource`, ou o primeiro artefato de entrada listado, se não `WorkflowSource` houver.
- O prefixo pode ser diferente dependendo da ação que você está usando. Para obter mais informações, consulte a tabela a seguir.

Tipo de ação	Prefixo do caminho do arquivo a ser usado	Exemplo
Crie ação , teste a ação	<code>\$(CATALYST_SOURCE_DIR)_<i>artifact-name</i></code> /	<code>\$(CATALYST_SOURCE_DIR)_MyArtifact/folder1/file.txt</code>
Todas as outras ações	<code>\$(CATALYST_SOURCE_DIR)_<i>artifact-name</i></code> / ou <code>/artifacts/ <i>current-action-name</i> /<i>artifact-name</i></code> / <i>(esse caminho é um link simbólico para</i>	<code>\$(CATALYST_SOURCE_DIR)_MyArtifact/folder1/file.txt</code> ou <code>/artifacts/MyCurrentAction/MyArtifact/folder1/file.txt</code>

Tipo de ação	Prefixo do caminho do arquivo a ser usado	Exemplo
	<i><code>\$CATALYST_SOURCE_D IR_ artifact-name/)</code></i>	

Baixando artefatos

Você pode baixar e inspecionar artefatos gerados pelas ações do seu fluxo de trabalho para fins de solução de problemas. Há dois tipos de artefatos que você pode baixar:

- Artefatos de origem — Um artefato que contém um instantâneo do conteúdo do repositório de origem tal como ele existia quando a execução foi iniciada.
- Artefatos do fluxo de trabalho — Um artefato definido na `Outputs` propriedade do arquivo de configuração do seu fluxo de trabalho.

Para baixar artefatos produzidos 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Sob o nome do fluxo de trabalho, escolha Executar.
6. Em Histórico de execução, na coluna ID da execução, escolha uma execução. Por exemplo, Run-95a4d.
7. Abaixo do nome da corrida, escolha Artefatos.
8. Ao lado de um artefato, escolha Baixar. Um arquivo de arquivamento 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.

Exemplos de artefatos

Os exemplos a seguir mostram como gerar, inserir e referenciar artefatos no arquivo de definição do fluxo de trabalho.

Tópicos

- [Exemplo: saída de um artefato](#)
- [Exemplo: inserindo um artefato gerado por outra ação](#)
- [Exemplo: referenciando arquivos em vários artefatos](#)
- [Exemplo: referenciando um arquivo em um único artefato](#)
- [Exemplo: referenciar um arquivo em um artefato quando um está presente WorkflowSource](#)

Exemplo: saída 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: inserindo um artefato gerado por outra ação

O exemplo a seguir mostra como gerar um artefato chamado ARTIFACT4 in BuildActionA e BuildActionB inseri-lo nele.

```
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: referenciando arquivos em vários artefatos

O exemplo a seguir mostra como gerar dois artefatos chamados ART5 e ART6 em eBuildActionC, em seguida, referenciar dois arquivos chamados file5.txt (no artefatoART5) e file6.txt (no artefatoART6) em BuildActionD (abaixo). Steps

Note

Para obter mais informações sobre como referenciar arquivos, consulte [Fazendo referência a arquivos em um artefato](#).

Note

Embora o exemplo mostre o \$CATALYST_SOURCE_DIR_ART5 prefixo que está sendo usado, você pode omiti-lo. Isso ocorre porque ART5 é a entrada principal. Para saber mais sobre a entrada principal, consulte [Fazendo referência a 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: referenciando um arquivo em um único artefato

O exemplo a seguir mostra como gerar um artefato chamado ART7 em `eBuildActionE`, em seguida, referenciar `file7.txt` (no artefato ART7) em `BuildActionF` (abaixoSteps).

Observe como a referência não exige o prefixo do `$CATALYST_SOURCE_DIR_ nome do artefato` na frente do `build-output` diretório, como acontecia em. [Exemplo: referenciando arquivos em vários artefatos](#) Isso ocorre porque há apenas um item especificado em `Inputs`.

Note

Para obter mais informações sobre como referenciar arquivos, consulte [Fazendo referência a 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: referenciar um arquivo em um artefato quando um está presente WorkflowSource

O exemplo a seguir mostra como gerar um artefato chamado ART8 em eBuildActionG, em seguida, referenciar file8.txt (no artefatoART8) em BuildActionH (abaixoSteps).

Observe como a referência exige o prefixo do \$CATALYST_SOURCE_DIR_ *nome do artefato*, como aconteceu em. [Exemplo: referenciando arquivos em vários artefatos](#) Isso ocorre porque há vários itens especificados em Inputs (uma fonte e um artefato), então você precisa do prefixo para indicar onde procurar o arquivo.

Note

Para obter mais informações sobre como referenciar arquivos, consulte [Fazendo referência a 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
```

Especificar a versão principal, secundária ou de patch de uma ação

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 `Identifier` propriedade 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 é `1.0.4`, então a ação será usada `1.0.4`. Se uma versão posterior for lançada, digamos `1.0.5`, a ação será usada `1.0.5`. Se uma versão secundária for lançada `1.1.0`, digamos, a ação continuará sendo usada `1.0.5`.

Para obter instruções detalhadas sobre como especificar versões, consulte um dos tópicos a seguir.

Use as instruções a seguir para indicar qual versão de uma ação você deseja que seu fluxo de trabalho use. Você pode especificar a versão principal ou secundária mais recente ou uma versão de patch específica.

Recomendamos usar a versão secundária ou de patch mais recente de uma ação.

Visual


Não disponível. Escolha YAML para ver as instruções do 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.

5. Selecione a opção Editar.
6. Escolha YAML.
7. Encontre a ação cuja versão você deseja editar.
8. Encontre a `Identifier` propriedade 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 específica do patch.

 Note

Se você não tiver certeza de quais versões estão disponíveis, consulte [Determinar quais versões de uma ação estão disponíveis](#).

 Note

Você não pode omitir a versão principal.

9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Determinar quais versões de uma ação estão 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, em seguida, escolha Fluxos de trabalho.
 - b. Escolha o nome de qualquer fluxo de trabalho ou crie um. Para obter informações sobre como criar um fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).
 - c. Selecione a opção Editar.
 - d. No canto superior esquerdo, escolha + 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 escolha seu nome. Não escolha o sinal de mais (+).

Os detalhes sobre a ação são exibidos.

4. Na caixa de diálogo de detalhes da ação, no canto superior direito, escolha a lista suspensa Versões para ver uma lista das versões disponíveis da ação.

YAML

Não disponível. Escolha 'visual' para ver as instruções do editor visual.

Visualizando o código-fonte de uma ação

Você pode visualizar 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.

Para ver 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, em seguida, escolha Fluxos de trabalho.
 - b. Escolha o nome de qualquer fluxo de trabalho ou crie um. Para obter informações sobre como criar um fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).
 - c. Selecione a opção Editar.
 - d. No canto superior esquerdo, escolha + 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 escolha seu nome. Não escolha o sinal de mais (+).

Os detalhes sobre a ação são exibidos.

4. Na caixa de diálogo de detalhes da ação, na parte inferior, escolha Baixar.

Uma página aparece mostrando o bucket do Amazon S3 onde reside o código-fonte da ação. Para obter 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.

Integrando GitHub ações em um fluxo de trabalho

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 ter 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 ter mais informações, consulte [Adicionando a ação "GitHub 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 que a ação funcione corretamente.

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 [Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo](#) e [Fazendo referência a um parâmetro GitHub de saída](#).

Quais GitHub ações posso usar?

Você pode usar qualquer GitHub ação disponível no CodeCatalyst console e qualquer GitHub ação disponível no [GitHub Marketplace](#). 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 obter mais informações sobre a imagem e as ferramentas, consulte [Especificando imagens do Docker do ambiente de execução](#).

- 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 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. O fluxo de trabalho é onde você define como criar, testar e implantar seu aplicativo. Para ter mais informações, consulte [Introdução aos fluxos de trabalho](#).
2. No fluxo de trabalho, você adiciona uma GitHub ação com curadoria ou adiciona a ação GitHub Ações.
3. Você faz um dos seguintes:
 - Se você optar por adicionar uma ação com curadoria, configure-a. Para ter 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 ter mais informações, consulte [Adicionando a ação "GitHub Ações"](#).
4. (Opcional) No fluxo de trabalho, você adiciona outras ações, como as ações de criação e teste. Para ter mais informações, consulte [Crie, teste e implante com fluxos de trabalho em CodeCatalyst](#).

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 ter mais informações, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

Para obter etapas detalhadas, consulte:

- [Adicionando uma ação com curadoria GitHub](#) .
- [Adicionando a ação "GitHub Ações"](#).

A GitHub ação é executada GitHub?

Não. A GitHub ação é CodeCatalyst executada usando [máquinas CodeCatalyst de construção](#).

Também posso usar GitHub fluxos de trabalho?

Não.

Tópicos

- [Tutorial: código Lint usando uma GitHub ação em um fluxo de trabalho](#)
- [Adicionando a ação "GitHub Ações"](#)
- [Adicionando uma ação com curadoria GitHub](#)
- [Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo](#)
- [Fazendo referência a um parâmetro GitHub de saída](#)
- [Definição YAML da GitHub ação "Ações"](#)

Tutorial: código Lint usando uma GitHub ação em um fluxo de trabalho

Neste tutorial, você adiciona a [GitHub ação Super-Linter](#) a um fluxo de trabalho da Amazon CodeCatalyst . A ação Super-Linter inspeciona o código, encontra áreas em que o código tem erros, problemas de formatação e construções suspeitas e, em seguida, envia os resultados para o console). CodeCatalyst Depois de adicionar o linter ao seu fluxo de trabalho, você executa o fluxo de trabalho para lintar um aplicativo Node.js de amostra ()app.js. Em seguida, você corrige os problemas relatados e executa o fluxo de trabalho novamente para ver se as correções funcionaram.

Tip

[Considere usar o Super-Linter para lintar arquivos YAML, como modelos.AWS 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 precisará:

- Um CodeCatalyst espaço com um conectado Conta da AWS. Para ter mais informações, consulte [Criando um espaço](#).
- Um projeto vazio em seu CodeCatalyst espaço chamado `codecatalyst-linter-project`. Escolha a opção Começar do zero para criar esse projeto.

Para ter 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 do aplicativo de amostra, `app.js`, para este tutorial.

Para obter mais informações sobre repositórios de origem, consulte [Criando um repositório de origem](#).

Como criar um repositório de origem

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

2. Navegue até seu 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, digite:

```
codecatalyst-linter-source-repository
```

6. Selecione Create (Criar).

Etapa 2: Adicionar um arquivo `app.js`

Nesta etapa, você adiciona um `app.js` arquivo ao seu 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 CodeCatalyst console, escolha seu projeto, `codecatalyst-linter-project`.
2. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
3. Na lista de repositórios de origem, escolha seu repositório, `codecatalyst-linter-source-repository`
4. Em Arquivos, escolha Criar arquivo.
5. Na caixa de texto, insira o seguinte código:

```
// 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. Escolha Commit (Confirmar).

Agora você criou um arquivo chamado `app.js`.

Etapa 3: criar um fluxo de trabalho que execute a ação Super-Linter

Nesta etapa, você cria um fluxo de trabalho que executa a ação Super-Linter ao enviar o código para o repositório de origem. O fluxo de trabalho consiste nos seguintes blocos de construção, 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação “GitHub Ações” — No gatilho, a ação GitHub Ações executa a ação Super-Linter, que por sua vez inspeciona todos os arquivos em seu 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 CodeCatalyst console, escolha seu projeto, `codecatalyst-linter-project`.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
3. Escolha Criar fluxo de trabalho.
4. Em Repositório de origem, escolha `codecatalyst-linter-source-repository`.
5. Para Branch, escolha `main`.
6. Selecione Create (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:
  SuperLinterAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        github-action-code
```

No código anterior, *github-action-code* substitua pelo código de ação Super-Linter, conforme instruído nas etapas a seguir deste procedimento.

9. Acesse a [página Super-Linter](#) no Marketplace GitHub .
10. Em `steps:` (minúsculas), localize o código e cole-o no CodeCatalyst fluxo de trabalho em `Steps:` (maiúsculas).

Ajuste o código de GitHub ação para se adequar aos CodeCatalyst padrões, conforme mostrado no código a seguir.

Seu CodeCatalyst fluxo de trabalho agora tem a seguinte aparência:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
```

```
- Type: PUSH
  Branches:
    - main
  Actions:
    SuperLintAction:
      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) Escolha Validar para garantir que o código YAML seja válido antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação, selecione seu `codecatalyst-linter-source-repository` repositório e escolha 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, em seguida, escolha Fluxos de trabalho.
2. Escolha o fluxo de trabalho que você acabou de criar: `codecatalyst-linter-workflow`.
3. No diagrama do fluxo de trabalho, escolha SuperLintAction.
4. Aguarde até que a ação falhe. Essa falha é esperada porque o linter encontrou problemas no código.
5. Deixe o CodeCatalyst console 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 `app.js` código, bem como no `README.md` arquivo incluído no seu repositório de origem.

Para corrigir os problemas encontrados pelo linter

1. No CodeCatalyst console, escolha a guia Logs e, em seguida, escolha Lint Code Base.

Os registros gerados pela ação Super-Linter são exibidos.

2. Nos registros do Super-Linter, role para baixo até a linha 90, onde você encontra o início dos problemas. Eles são parecidos com os seguintes:

```
/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` em seu repositório de origem e confirme suas 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 até que o fluxo de trabalho termine. Se você corrigiu todos os problemas, o fluxo de trabalho deverá ser bem-sucedido.

Limpeza

Faça uma limpeza CodeCatalyst para remover vestígios deste tutorial do seu ambiente.

Para limpar CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Excluir `codecatalyst-linter-source-repository`.
3. Excluir `codecatalyst-linter-workflow`.

Neste tutorial, você aprendeu como adicionar a GitHub ação Super-Linter a um CodeCatalyst fluxo de trabalho para codificar alguns códigos.

Adicionando a 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.

Para obter mais informações, consulte [Integrando GitHub ações em um fluxo de trabalho](#).

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 GitHub ação em um fluxo de trabalho](#).

Visual

Para adicionar a ação "GitHub 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Pesquise a ação GitHub Ações e faça o seguinte:
 - Escolha 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) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da GitHub ação "Ações"](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar a ação "GitHub 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Pesquise a ação GitHub Ações e faça o seguinte:
 - Escolha 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) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Definição YAML da GitHub ação "Ações"](#).
- 11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- 12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Definição de GitHub ação "Ações"

A ação GitHub Ações é definida como um conjunto de propriedades YAML dentro do arquivo de definição do fluxo de trabalho. Para obter informações sobre essas propriedades, consulte [Definição YAML da GitHub ação "Ações"](#) no [Definição de fluxo de trabalho YAML](#).

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 abaixo `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 trufflehog 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 [Integrando GitHub ações em um fluxo de trabalho](#).

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 ação "GitHub 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Procure ou pesquise uma GitHub ação e faça o seguinte:
 - Escolha 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) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Definição YAML da GitHub ação "Ações"](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Escolha Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha. GitHub
9. Procure ou pesquise uma GitHub ação e faça o seguinte:
 - Escolha 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) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível para a ação GitHub Ações é fornecida no [Definição YAML da GitHub ação "Ações"](#).

Para obter informações sobre as opções de configuração disponíveis para a GitHub Ação selecionada, consulte sua documentação.

11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo

Você pode usar [parâmetros GitHub de saída](#) em seus CodeCatalyst fluxos de trabalho.

 Note

Outra palavra para parâmetro de saída é variável. Como GitHub usa o termo parâmetro de saída em sua documentação, também usaremos esse termo.

Use as instruções a seguir para exportar um parâmetro de GitHub saída de uma GitHub Ação para que fique disponível para uso por outras ações do CodeCatalyst fluxo de trabalho.

Para exportar um parâmetro GitHub de saída

1. Abra um fluxo de trabalho e escolha Editar. Para ter mais informações, consulte [Criação de um fluxo de trabalho](#).
2. Na ação GitHub Ações que gera o parâmetro de saída que você deseja exportar, adicione uma Outputs seção com uma Variables propriedade subjacente semelhante a esta:

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'step-id_output-name'
```

Substitua:

- *step-id* com o valor da id: propriedade na seção da GitHub steps ação.
- *output-name com o nome* do parâmetro de saída. GitHub

Exemplo

O exemplo a seguir mostra como exportar um parâmetro GitHub de saída chamadoSELECTEDCOLOR.

```
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
```

Fazendo referência a um parâmetro GitHub de saída

Use as instruções a seguir para referenciar um parâmetro GitHub de saída.

Para referenciar um parâmetro GitHub de saída

1. Siga as etapas em [Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo](#).

O parâmetro GitHub de saída agora está disponível para uso em outras ações.

2. Observe o `Variables` valor do parâmetro de saída. Inclui um sublinhado (`_`).
3. Consulte o parâmetro de saída usando a seguinte sintaxe:

```
${action-name.output-name}
```

Substitua:

- ***action-name*** com o nome da CodeCatalyst GitHub Ação que produz o parâmetro de saída (não use o nome ou id da GitHub ação).
- ***output-name*** com o Variables valor 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 SELECTEDCOLOR variávelGitHubActionA, exibi-la e, em seguida, referenciá-la emBuildActionB.

```
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}
```

Definição YAML da GitHub ação “Ações”

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 de fluxo de trabalho YAML](#).

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 do usuário, use Ctrl+F. O elemento será listado com sua 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*

Configuration

Steps:

- *github-actions-code*

nome da ação

(Obrigatório)

Especifique o nome da ação. Todos os nomes das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

UI correspondente: guia de configuração/nome da ação

Identifier

(nome da ação/) Identifier

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Use `aws/github-actions-runner@v1` para GitHubações.

UI correspondente: diagrama de fluxo de *trabalho/nome da ação/ rótulo aws/*@v1 github-actions-runner

DependsOn

(nome da ação/) DependsOn

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(nome da ação) Compute

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Fleet

(nome da ação) /Compute/ Fleet

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 é concluída. Exemplos de frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

UI correspondente: guia de configuração/frota de computação - opcional

Timeout

(nome da ação) 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é o mesmo que o tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Environment

(nome da ação) / Environment

(Optional)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/ambiente/conta/função

Name

(nome da ação) / Environment / Name

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Connections

(nome da ação) / Environment / Connections

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: nenhuma

Name

(nome da *ação*) */Environment/Connections/ Name*

(Optional)

Especifique o nome da conexão da conta.

UI correspondente: guia de configuração/'Ambiente/conta/função'/ conexão AWS de conta

Role

(nome da *ação*) */Environment/Connections/ Role*

(Optional)

Especifique o nome da função do IAM que essa ação usa para acessar e operar em AWS serviços como Amazon S3 e Amazon ECR. Certifique-se de que essa função seja adicionada à conexão da sua conta. Para adicionar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da CodeCatalystWorkflowDevelopmentRole-*spaceName* função aqui, desde que ela tenha permissões suficientes. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a CodeCatalystWorkflowDevelopmentRole-*spaceName* função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

Warning

Limite as permissões às exigidas pela GitHub ação Ação. Usar uma função com permissões mais amplas pode representar um risco de segurança.

UI correspondente: guia Configuração/'Ambiente/Conta/Função'/Função

Inputs

(*nome da ação*/) Inputs

(Optional)

A Inputs seção define os dados 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 contam para esse total.

Se você precisar se referir a arquivos que residem em entradas diferentes (digamos, uma fonte e um artefato), a entrada de origem é a entrada primária e o artefato é a entrada secundária. As referências a arquivos em entradas secundárias usam um prefixo especial para diferenciá-las das primárias. Para obter detalhes, consulte [Exemplo: referenciando arquivos em vários artefatos](#).

UI correspondente: guia Entradas

Sources

(nome da *ação*) /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 seu arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma fonte, deverá especificar pelo menos um artefato de entrada abaixo. *action-name*/Inputs/Artifacts

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - opcional

Artifacts - input

(nome da *ação*) /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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Note

Se a lista suspensa Artefacts - opcional não estiver disponível (editor visual) ou se você receber erros ao validar seu YAML (editor YAML), talvez seja porque a ação suporta apenas uma entrada. Nesse caso, tente remover a entrada da fonte.

UI correspondente: guia Entradas/Artefatos - opcional

Variables - input

(nome da *ação*) /Inputs/ **Variables**

(Optional)

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de entradas/variáveis - opcional

Outputs

(*nome da ação*/) Outputs

(Optional)

Define os dados que são gerados pela ação durante a execução de um fluxo de trabalho.

UI correspondente: guia Saídas

Artifacts - output

(nome da *ação*) */Outputs/ Artifacts*

(Optional)

Especifique o nome de um artefato gerado pela ação. Os nomes dos artefatos devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (_). Espaços, hífen (-) e outros caracteres especiais não são permitidos. Você não pode usar aspas para ativar espaços, hífen e outros caracteres especiais nos nomes dos artefatos de saída.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos

Name

(nome da *ação*) */Outputs/Artifacts/ Name*

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique o nome de um artefato gerado pela ação. Os nomes dos artefatos devem ser exclusivos em um fluxo de trabalho e estão limitados a caracteres alfanuméricos (a-z, A-Z, 0-9) e sublinhados (_). Espaços, hífen (-) e outros caracteres especiais não são permitidos. Você não pode usar aspas para ativar espaços, hífen e outros caracteres especiais nos nomes dos artefatos de saída.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Criar nome do artefato

Files

(nome da *ação*) */Outputs/Artifacts/ Files*

(Obrigatório se [Artifacts - output](#) estiver incluído)

Especifique os arquivos CodeCatalyst incluídos no artefato gerado pela ação. Esses arquivos são gerados pela ação do fluxo de trabalho quando ela é executada e também estão disponíveis no seu 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. Você pode usar padrões globais 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 obter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

Note

Talvez seja necessário adicionar um prefixo ao caminho do arquivo para indicar em qual artefato ou fonte encontrá-lo. Para obter mais informações, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#).

UI correspondente: guia Saídas/Artefatos/Adicionar artefato/Arquivos produzidos pela compilação

Variables - output

(nome da *ação*) */Outputs/ Variables*

(Optional)

Especifique as variáveis que você deseja que a ação exporte para que estejam disponíveis para uso em ações subsequentes.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia Saídas/Variáveis/Adicionar variável

nome da variável-1

(nome da **ação nome da variável-1**/Outputs/Variables)

(Optional)

Especifique o nome de uma variável que você deseja que a ação exporte. Essa variável já deve estar definida na Steps seção Inputs ou da mesma ação.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de saídas/variáveis/Adicionar variável/nome

AutoDiscoverReports

(nome da **ação**) /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. Você pode limitar quais arquivos são inspecionados usando as [ExcludePaths](#) propriedades [IncludePaths](#) ou.

UI correspondente: nenhuma

Enabled

(nome da *ação*) */Outputs/AutoDiscoverReports/ Enabled*

(Optional)

Ative ou desative o recurso de descoberta automática.

Os valores válidos são true ou false.

Se Enabled for omitido, o padrão será. true

UI correspondente: guia Saídas/Relatórios/Descubra relatórios automaticamente

ReportNamePrefix

(nome da *ação*) */Outputs/AutoDiscoverReports/ ReportNamePrefix*

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e ativado)

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: guia Saídas/Relatórios/Descoberta automática de relatórios/Prefixo do relatório

IncludePaths

(nome da *ação*) */Outputs/AutoDiscoverReports/ IncludePaths*

Ou

(nome da *ação nome do* */Outputs/Reports/ relatório-1/*) IncludePaths

(Obrigatório se [AutoDiscoverReports](#) estiver incluído e ativado, 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 obter 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

Para relatórios configurados manualmente, `IncludePaths` deve ser um padrão global que corresponda a um único arquivo.

UI correspondente:

- Guia de saídas/Relatórios/Descubra relatórios automaticamente/'Incluir/Excluir caminhos'/Incluir caminhos
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/ nome-do-relatório-1 /'Incluir/Excluir caminhos'/Incluir caminhos***

`ExcludePaths`

(nome da *ação*) `/Outputs/AutoDiscoverReports/ ExcludePaths`

Ou

(nome da *ação nome do* `/Outputs/Reports/ relatório-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 `**/*` glob.

Note

Se o caminho do arquivo incluir um ou mais asteriscos (*) ou outros caracteres especiais, coloque o caminho entre aspas duplas (). "" Para obter mais informações sobre caracteres especiais, consulte [Diretrizes e convenções de sintaxe](#).

UI correspondente:

- Guia de saídas/Relatórios/Descubra relatórios automaticamente/'Incluir/Excluir caminhos'/Excluir caminhos
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/ nome-do-relatório-1 /'Incluir/Excluir caminhos'/Excluir caminhos***

SuccessCriteria

(nome da *ação*) /Outputs/AutoDiscoverReports/ ***SuccessCriteria***

Ou

(nome da *ação nome do* /Outputs/Reports/ *relatório-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 ter mais informações, consulte [Configurando critérios de sucesso para relatórios](#).

UI correspondente:

- Guia Saídas/Relatórios/Descubra relatórios automaticamente/Critérios de sucesso
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1/Critérios de sucesso***

PassRate

(nome da *ação*) /Outputs/AutoDiscoverReports/SuccessCriteria/ ***PassRate***

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de taxa de aprovação são aplicados somente aos relatórios de teste. Para obter mais informações sobre relatórios de teste, consulte [Relatórios de teste](#).

UI correspondente:

- Guia Saídas/Relatórios/Descobrir relatórios automaticamente/Critérios de sucesso/Taxa de aprovação
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Critérios de sucesso/ Taxa de aprovação***

LineCoverage

*(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage***

Ou

*(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura de linha são aplicados somente aos relatórios de cobertura de código. Para obter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente:

- Guia Saídas/Relatórios/Descubra automaticamente os relatórios/Critérios de sucesso/Cobertura de linha

- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Critérios de sucesso/ Cobertura de linha***

BranchCoverage

(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage**

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) /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 incluem números decimais. Por exemplo: 50, 60.5. Os critérios de cobertura da filial são aplicados somente aos relatórios de cobertura de código. Para obter mais informações sobre relatórios de cobertura de código, consulte [Relatórios de cobertura de código](#).

UI correspondente:

- Guia Saídas/Relatórios/Descubra relatórios automaticamente/Critérios de sucesso/Cobertura de filiais
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Critérios de sucesso/ Cobertura da filial***

Vulnerabilities

(nome da ação) /Outputs/AutoDiscoverReports/SuccessCriteria/ **Vulnerabilities**

Ou

(nome da ação nome do relatório-1/Outputs/Reports/) /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, você deve especificar:

- A gravidade mínima das vulnerabilidades que você deseja incluir na contagem. Os valores válidos, do mais para o menos severo, são: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Por exemplo, se você escolherHIGH, HIGH CRITICAL as vulnerabilidades serão computadas.

- O número máximo de vulnerabilidades da severidade 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 da SCA. Para obter mais informações sobre relatórios SCA, consulte[Relatórios de análise de composição de software](#).

Para especificar a severidade mínima, use a Severity propriedade. Para especificar o número máximo de vulnerabilidades, use a Number propriedade.

Para obter mais informações sobre relatórios SCA, consulte[Tipos de relatório de qualidade](#).

UI correspondente:

- Guia Saídas/Relatórios/Descubra automaticamente os relatórios/Critérios de sucesso/Vulnerabilidades
- ***Guia Saídas/Relatórios/Configurar manualmente os relatórios/nome-do-relatório-1 /Critérios de sucesso/ Vulnerabilidades***

Reports

(nome da *ação*) /Outputs/ **Reports**

(Optional)

Uma seção que especifica a configuração dos relatórios de teste.

UI correspondente: guia Saídas/Relatórios

nome do relatório-1

(nome da ***ação nome do relatório-1***/Outputs/Reports/)

(Obrigatório se [Reports](#) estiver incluído)

O nome que você deseja dar ao CodeCatalyst relatório que será gerado a partir de seus relatórios brutos.

UI correspondente: guia Saídas/Relatórios/Configurar manualmente os relatórios/Nome do relatório

Format

(nome da *ação nome do /Outputs/Reports/ relatório-1/*) Format

(Obrigatório se [Reports](#) estiver incluído)

Especifique o formato de arquivo que você está usando para seus relatórios. Os valores possíveis são.

- 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: guia Saídas/Relatórios/Configurar manualmente os relatórios/Adicionar relatório/nome-do-relatório-1/Tipo de relatório e formato do relatório

Configuration

(*nome da ação/*) Configuration

(Obrigatório) Uma seção na qual você pode definir as propriedades de configuração da ação.

UI correspondente: guia Configuração

Steps

(nome da *ação*) */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 é semelhante ao seguinte.

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 em *itálico vermelho* e adicionar o código em negrito.

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 de fluxo de trabalho YAML](#) para obter informações sobre como você pode portar seu GitHub código para CodeCatalyst o. As etapas detalhadas de 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 obter mais informações sobre como especificar caminhos de arquivo, consulte [Fazendo referência a arquivos em um repositório de origem](#) e [Fazendo referência a arquivos em um artefato](#)

UI correspondente: guia de GitHub configuração/ações YAML

Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho

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 a 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 de frota sob demanda](#)
- [Propriedades da frota provisionada](#)
- [Criação de uma frota provisionada](#)
- [Editando uma frota provisionada](#)
- [Excluindo uma frota provisionada](#)
- [Atribuindo uma frota provisionada ou computação sob demanda a uma ação](#)
- [Compartilhamento de computação entre ações](#)
- [Especificando imagens do Docker do ambiente de execução](#)

Tipos de computação

CodeCatalyst oferece os seguintes tipos de computação:

- Amazon EC2
- AWS Lambda

O Amazon EC2 oferece flexibilidade otimizada durante a execução de ações e o Lambda oferece velocidades otimizadas de inicialização de ações. O Lambda suporta execuções mais rápidas de ações de fluxo de trabalho devido à menor latência de inicialização. O Lambda permite que você execute fluxos de trabalho básicos que podem criar, testar e implantar aplicativos sem servidor com tempos de execução comuns. Esses tempos de execução 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 oferece suporte a imagens do ambiente de execução de um registro especificado.
- O Lambda não oferece suporte a ferramentas que exigem permissões de root. Para ferramentas como yum ou rpm, use o tipo de computação Amazon EC2 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 também não são compatíveis com a computação Lambda. É possível usar alternativas que não exijam permissões de raiz, como o Podman.

- O Lambda não oferece suporte à gravação em arquivos externos. /tmp Ao configurar suas ações de fluxo de trabalho, você pode reconfigurar suas ferramentas para instalação ou gravação. /tmp Se você tiver uma ação de compilação que seja instaladanpm, certifique-se de configurá-la para /tmp instalação em.
- O Lambda não suporta tempos de execução superiores a 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 paga apenas pelo número de minutos em que está executando 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 pelo Amazon EC2 que são mantidas pela. CodeCatalyst Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Com frotas provisionadas, suas máquinas estão sempre funcionando e terão custos enquanto forem provisionadas.

Para criar, atualizar ou excluir uma frota, você deve ter a função de administrador do espaço ou a função de administrador do projeto.

Propriedades de frota sob demanda

CodeCatalyst fornece as seguintes frotas sob demanda:

| Nome | Sistema operacional | Arquitetura | vCPUs | 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 |
| 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 cobrança. Para obter mais informações, consulte [Preços do](#).

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 suportadas somente na ação de criação. No momento, outras ações não oferecem suporte ao 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:

| vCPUs | 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 |

Capacity

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, que aumentam automaticamente em resposta às novas ações em execução e, em seguida, diminuem até a capacidade básica à medida que as ações são concluídas. 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 ações são colocadas em uma fila até que a máquina esteja disponível. Isso limita os custos adicionais porque nenhuma máquina adicional é alocada.

Criação de 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 usados novamente, eles serão reativados automaticamente, mas essa reativação pode causar uma latência.

Para criar uma frota provisionada

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Compute.
2. Escolha Criar frota provisionada.
3. No campo de texto Nome da frota provisionada, insira um nome para sua frota.
4. No menu suspenso Sistema operacional, escolha o sistema operacional.
5. No menu suspenso Tipo de máquina, escolha o tipo de máquina para sua 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. Selecione Create (Criar).

Depois de criar a frota provisionada, você está pronto para atribuí-la a uma ação. Para ter mais informações, consulte [Atribuindo uma frota provisionada ou computação sob demanda a uma ação](#).

Editando 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 usados novamente, eles serão reativados automaticamente, mas essa reativação pode causar uma latência.

Para editar uma frota provisionada

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Compute.
2. Na lista de frotas provisionadas, escolha a frota que você deseja editar.
3. Selecione a opção 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.

Excluindo 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 `Fleet` propriedade 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, escolha `CI/CD` e, em seguida, escolha `Compute`.
2. Na lista de frotas provisionadas, escolha a frota que você deseja excluir.
3. Escolha `Excluir`.
4. Digite **delete** para confirmar a exclusão.
5. Escolha `Excluir`.

Atribuindo uma frota provisionada ou computação sob demanda a uma ação

Por padrão, as ações de fluxo de trabalho usam a frota `Linux.x86-64.Large` sob demanda com um tipo de computação do Amazon EC2. Para usar uma frota provisionada em vez disso, ou usar uma frota sob demanda diferente, como `Linux.x86-64.2XLarge`, use as instruções a seguir.

Visual

Antes de começar

- Se quiser atribuir uma frota provisionada, você deve primeiro criar a frota provisionada. Para ter mais informações, consulte [Criação de uma frota provisionada](#).

Para 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, em seguida, escolha `Fluxos de trabalho`.

4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação à qual você deseja atribuir sua frota provisionada ou novo tipo de frota.
8. Escolha a guia Configuração.
9. Em Compute fleet, faça o seguinte:

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será `Linux.x86-64.Large`

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Antes de começar

- Se quiser atribuir uma frota provisionada, você deve primeiro criar a frota provisionada. Para ter mais informações, consulte [Criação de uma frota provisionada](#).

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho. /
5. Selecione a opção Editar.
6. Escolha YAML.
7. Encontre a ação à qual você deseja atribuir sua frota provisionada ou novo tipo de frota.
8. Na ação, adicione uma Compute propriedade e Fleet defina o nome da sua frota ou o tipo de frota sob demanda. Para obter mais informações, consulte a descrição da Fleet propriedade em [Crie e teste a definição YAML da ação](#) Para sua ação.
9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha 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 tempos de execução 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 para a duração de todas as ações nesse fluxo de trabalho. Quando a execução do fluxo de trabalho é concluída, a reserva da instância é liberada.

Tópicos

- [Executando várias ações em computação compartilhada](#)
- [Considerações sobre o compartilhamento de computação](#)
- [Ativando o compartilhamento de computação](#)
- [Exemplos](#)

Executando várias ações em computação compartilhada

Você pode usar o `Compute` atributo na definição YAML no nível do fluxo de trabalho para especificar as propriedades de compartilhamento de frota e computação das ações. Você também pode configurar propriedades computacionais usando o editor visual em 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

O compartilhamento de computação só é suportado se o tipo de computação estiver definido como EC2 e não é compatível com o sistema operacional Windows Server 2022. Para obter mais informações sobre frotas de computação, tipos de computação e propriedades, consulte [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#)

Note

Se você estiver no nível Gratuito e especificar a `Linux.x86-64.2XLarge` frota `Linux.x86-64.XLarge` ou manualmente na definição de fluxo de trabalho YAML, a ação ainda será executada na frota padrão (`Linux.x86-64.Large`). Para obter 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 fonte 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 [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#) e [Armazenando 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 fonte 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 fonte 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 uma das ações do fluxo de trabalho, da mesma forma que você pode, sem usar o compartilhamento de computação. Para ter mais informações, consulte [Fazendo referência a arquivos em um 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, os artefatos de entrada não são suportados.

Considerações sobre o compartilhamento de computação

Você pode executar fluxos de trabalho com compartilhamento de computação para acelerar a execução do 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 instâncias | Ações executadas na mesma instância | As ações são executadas em instâncias separadas |

| | Compartilhamento de computação | Sem compartilhamento de computação |
|--|--|--|
| Sistema operacional | Amazon Linux 2 | Amazon Linux 2, Windows Server 2022 (somente ação de compilação) |
| Arquivos de referência | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code> | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code> |
| Estrutura Workflow | As ações só podem ser executadas sequencialmente | As ações podem ser executadas paralelamente |
| Acessando dados em todas as ações do fluxo de trabalho | Acesse a fonte de fluxo de trabalho em cache (<code>WorkflowSource</code>) | Acesse saídas de artefatos compartilhados (requer configuração adicional) |

Ativando o compartilhamento de computação

Use as instruções a seguir para ativar o compartilhamento de computação em um fluxo de trabalho.

Visual

Para ativar o compartilhamento de computação 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. Escolha Propriedades do fluxo de trabalho.
8. No menu suspenso Tipo de computação, escolha EC2.

9. (Opcional) No menu suspenso Compute fleet - 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 obter mais informações, consulte [Criação de uma frota provisionada](#) e [Atribuindo uma frota provisionada ou computação sob demanda 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 para o fluxo de trabalho. Para ter mais informações, consulte [Configurando o comportamento de enfileiramento das execuções](#).
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para ativar o compartilhamento de computação 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Ative o compartilhamento de computação configurando o SharedInstance campo para TRUE e Type para EC2. Fleet Defina 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 obter mais informações, consulte [Criação de uma frota provisionada](#) e [Atribuindo uma frota provisionada ou computação sob demanda a uma ação](#)

Em um fluxo de trabalho YAML, 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha 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 Amazon 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 o cache. `WorkflowSource` Além disso, o artefato de saída na ação Construir não é mais necessário. A ação S3 Publish está configurada para usar a `DependsOn` propriedade explícita para manter ações sequenciais; a ação Build deve ser executada com êxito para que a ação 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: dev-bucket

```

- Ao usar o compartilhamento de computação configurando como `SharedInstanceTRUE`, você pode executar várias ações na mesma instância e compartilhar artefatos especificando uma única fonte 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: dev-bucket
```

Especificando imagens do Docker do ambiente de execução

Uma imagem do ambiente de tempo de execução é um contêiner do Docker dentro do 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 ter mais informações, consulte [Atribuindo uma imagem Docker de um ambiente de tempo de execução personalizado a uma ação](#).

Tópicos

- [Imagens ativas](#)
- [E se uma imagem ativa não incluir as ferramentas de que preciso?](#)
- [Atribuindo uma imagem Docker de um ambiente de tempo de execução personalizado 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.

Se uma ação usa uma 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 Container seção na definição de YAML que especifica explicitamente uma imagem de março de 2024.](#) Opcionalmente, você pode remover a Container seção 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 Container seção na definição de YAML e, conseqüentemente, usarão uma imagem [de novembro de 2022](#). Você pode manter a imagem de novembro de 2022 ou atualizá-la. Para atualizar a imagem, abra a ação no editor visual, escolha a guia Configuração e selecione a imagem de março de 2024 na lista suspensa de imagens docker do ambiente Runtime. Essa seleção adicionará uma Container seção à definição 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](#), independentemente de quando foram adicionadas ao fluxo de trabalho. Atualmente, não é possível atualizar essas ações para usar uma imagem de março de 2024.

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 de tipo de computação/frota.

A tabela a seguir mostra as ferramentas instaladas em cada imagem de março de 2024.

Ferramentas de imagem de março de 2024

| Ferramenta | CodeCatalyst Amazon EC2 para Linux x86_64 - CodeCatalystLinux_x86_64:2024_03 | CodeCatalyst Lambda para Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2024_03 | CodeCatalyst Amazon EC2 para Linux Arm64 - CodeCatalystLinux_Arm64:2024_03 | CodeCatalyst Lambda para Linux Arm64 - CodeCatalystLinuxLambda_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 | Correto 17 | Correto 17 | Correto 17 | Correto 17 |
| 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 |
| 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 combinação de tipo de computação/frota. Também há uma imagem do Windows de novembro de 2022 disponível com a ação de criação se você tiver 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 Amazon EC2 para Linux x86_64 - CodeCatalystLinux_x86_64:2022_11 | CodeCatalyst Lambda para Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2022_11 | CodeCatalyst Amazon EC2 para Linux Arm64 - CodeCatalystLinux_Arm64:2022_11 | CodeCatalyst Lambda para Linux Arm64 - CodeCatalystLinuxLambda_Arm64:2022_11 |
|--------------------|--|--|--|--|
| 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 |
| 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 | Correto 17 | Correto 17 | Correto 17 | Correto 17 |
| 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 |

| Ferramenta | CodeCatalyst
Amazon EC2 para
Linux x86_64
- CodeCatalystLinux_
x86_64:2022_11 | CodeCatalyst
Lambda para
Linux x86_64
- CodeCatalystLinuxL
ambda_x86
_64:2022_11 | CodeCatalyst
Amazon EC2
para Linux Arm64
- CodeCatalystLinux_
Arm64:2022_11 | CodeCatalyst
Lambda para
Linux Arm64 - CodeCatalystLinux_
ambda_Ar
64:2022_11 |
|------------|---|---|---|---|
| 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 Docker de ambiente de tempo de execução personalizada que inclua as ferramentas necessárias. Para ter mais informações, consulte [Atribuindo uma imagem Docker de um ambiente de tempo de execução personalizado a uma ação](#).

Note

Se você quiser fornecer uma imagem Docker de ambiente de tempo de execução personalizada, certifique-se de que sua imagem personalizada tenha o Git instalado nela.

- Você pode fazer com que a ação de criação ou teste do seu fluxo de trabalho instale as ferramentas necessárias.

Por exemplo, você pode incluir as seguintes instruções na Steps seção do código YAML da ação de compilação ou teste:

```
Configuration:
  Steps:
    - Run: ./setup-script
```

A instrução `setup-script` 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 obter mais informações sobre a ação de construção YAML, consulte [Crie e teste a definição YAML da ação](#).

Atribuindo uma imagem Docker de um ambiente de tempo de execução personalizado 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, certifique-se de que ela tenha o Git instalado nela. A imagem pode residir no Docker Hub, no Amazon Elastic Container Registry ou em qualquer repositório público.

Para saber como criar uma imagem personalizada do Docker, consulte [Containerize an application](#) na documentação do Docker.

Use as instruções a seguir para atribuir a imagem do Docker do seu ambiente de tempo de execução personalizado 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 AWS 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 Docker de ambiente de tempo de execução personalizado usando o editor visual

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha Visual.
6. No diagrama do fluxo de trabalho, escolha a ação que usará a imagem Docker do seu ambiente de tempo de execução personalizado.
7. Escolha a guia Configuração.
8. Na parte inferior, preencha os campos a seguir.

Imagem do Docker do ambiente de execução - opcional

Especifique o registro em que sua 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, você deve criar uma [função do IAM](#)

que inclua as seguintes permissões e a política de confiança personalizada. (Você pode modificar uma função existente para incluir as permissões e a política, se quiser.)

A função do IAM deve incluir as seguintes permissões em sua política de função:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

A função do IAM deve incluir a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obter mais informações sobre a criação de funções do IAM, consulte [Como criar uma função usando políticas de confiança personalizadas \(console\)](#) no Guia do usuário do IAM.

Depois de criar a função, você deve atribuí-la à ação por meio de um ambiente. Para ter mais informações, consulte [Associando um ambiente, uma conexão de conta e uma função do IAM a uma ação de fluxo de trabalho](#).

URL da imagem ECR, imagem do Docker Hub ou URL da imagem

Especifique um dos seguintes:

- Se você estiver usando um CODECATALYST registro, 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 com o nome da imagem do Docker Hub e a tag opcional.

Exemplo: `postgres:latest`

- Se você 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 você estiver usando um registro personalizado, defina a imagem com o valor esperado pelo registro personalizado.

9. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
10. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para atribuir uma imagem do Docker de um ambiente de tempo de execução personalizado usando o editor YAML

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.

2. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
3. Selecione a opção Editar.
4. Escolha YAML.
5. Encontre a ação à qual você deseja atribuir uma imagem do Docker do ambiente de tempo de execução.
6. Na ação, adicione uma Container seção Registry e Image propriedades subjacentes. Para obter mais informações, consulte a Container descrição Registry e Image as propriedades do [Ações](#) para sua ação.
7. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
8. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Exemplos

Os exemplos a seguir mostram como atribuir uma imagem Docker de ambiente de tempo de execução personalizada a uma ação no arquivo de definição do fluxo de trabalho.

Tópicos

- [Exemplo: usando uma imagem Docker de ambiente de tempo de execução personalizado para adicionar suporte ao Node.js 18 com o Amazon ECR](#)
- [Exemplo: usando uma imagem Docker de ambiente de tempo de execução personalizado para adicionar suporte ao Node.js 18 com o Docker Hub](#)

Exemplo: usando uma imagem Docker de ambiente de tempo de execução personalizado para adicionar suporte ao Node.js 18 com o Amazon ECR

O exemplo a seguir mostra como usar uma imagem Docker de ambiente de tempo de execução personalizada para adicionar suporte ao Node.js 18 com o [Amazon ECR](#).

```
Configuration:
  Container:
    Registry: ECR
    Image: public.ecr.aws/amazonlinux/amazonlinux:2023
```

Exemplo: usando uma imagem Docker de ambiente de tempo de execução personalizado para adicionar suporte ao Node.js 18 com o Docker Hub

O exemplo a seguir mostra como usar uma imagem Docker de ambiente de tempo de execução personalizada para adicionar suporte ao Node.js 18 com o [Docker Hub](#).

```
Configuration:
  Container:
    Registry: DockerHub
    Image: node:18.18.2
```

Conectando um fluxo de trabalho a um repositório de origem

Uma fonte, também chamada de fonte de entrada, é um repositório de origem ao qual uma [ação de fluxo](#) de trabalho se conecta para obter os arquivos necessários para realizar suas operações. Por exemplo, uma ação de fluxo de trabalho pode se conectar a um repositório de origem para obter arquivos de origem do aplicativo a fim de criar um aplicativo.

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 e repositórios Bitbucket — Para obter mais informações, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#)

Tópicos

- [Especificando a fonte que armazenará o arquivo de definição do fluxo de trabalho](#)
- [Especificando a fonte que uma ação de fluxo de trabalho usará](#)
- [Fazendo referência a arquivos em um repositório de origem](#)
- [Variáveis produzidas pela fonte \(" BranchName "e" CommidId "\)](#)

Especificando a fonte que armazenará o arquivo de definição do fluxo de trabalho

Use as instruções a seguir para especificar o repositório de CodeCatalyst origem em que você deseja armazenar seu arquivo de definição de fluxo de trabalho. Se você preferir especificar um

repositório GitHub ou Bitbucket, 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

Você especifica 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.

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha Criar fluxo de trabalho e crie o fluxo de trabalho. Para ter mais informações, consulte [Para criar um fluxo de trabalho usando o editor visual](#).

Durante o processo de criação do fluxo de trabalho, você deverá especificar o CodeCatalyst repositório em que deseja armazenar o arquivo de definição do fluxo de trabalho.

Especificando a fonte que uma ação de fluxo de trabalho usará

Use as instruções a seguir para especificar um repositório de origem a ser usado com uma ação de fluxo de trabalho. Na inicialização, a ação agrupa os arquivos no repositório de origem configurado em um artefato, baixa o artefato para a [imagem Docker do ambiente de tempo de execução](#) em que a ação está sendo executada e, em seguida, 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 `.codecatalyst/workflows/` diretório). Esse repositório de origem é representado pelo rótulo `WorkflowSource`.

Visual

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação em que você deseja especificar a fonte.
8. Escolha Entradas.
9. Em Fontes - 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 seu arquivo de definição de fluxo de trabalho está armazenado.

Se você omitir uma fonte, deverá especificar pelo menos um artefato de entrada abaixo.

`action-name/Inputs/Artifacts`

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Em uma ação, adicione um código semelhante ao seguinte:

```
action-name:  
  Inputs:  
    Sources:  
      - WorkflowSource
```

Para obter mais informações, consulte a descrição da Sources propriedade em [Definição de fluxo de trabalho YAML](#) Para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Fazendo referência a arquivos em um 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 [Fazendo referência a arquivos em um artefato](#).

Para referenciar um arquivo em um repositório de origem

- Na ação em que você deseja referenciar 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 `my-app` diretório na raiz do repositório de `WorkflowSource` origem para encontrar e exibir o `file1.jar` arquivo.

Variáveis produzidas pela fonte ("BranchName" e "CommitId")

A CodeCatalyst fonte produz e define as variáveis "BranchName" e "CommitId" quando seu fluxo de trabalho é executado. Elas são conhecidas como variáveis predefinidas. Consulte a tabela a seguir para obter informações sobre essas variáveis.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

| Chave | Valor |
|----------|---|
| CommitId | <p>O ID do commit que representa o estado do repositório no momento em que a execução do fluxo de trabalho foi iniciada.</p> <p>Exemplo: <code>example3819261db00a3ab59468c8b</code></p> <p>Consulte também: Exemplo: referenciando a variável predefinida CommitId</p> |

| Chave | Valor |
|------------|--|
| BranchName | <p>O nome da ramificação na qual a execução do fluxo de trabalho foi iniciada.</p> <p>Exemplos: main, feature/branch , test-LiJuan</p> <p>Consulte também: Exemplo: referenciando a variável predefinida BranchName ""</p> |

Publicar e importar pacotes usando um fluxo 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 popular)
- 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 obter 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

- [Especificando repositórios de CodeCatalyst pacotes em fluxos de trabalho](#)
- [Exemplos de especificação de repositórios de pacotes em fluxos de trabalho](#)

Especificando 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

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação com a qual você deseja configurar com um repositório de pacotes.
8. Escolha Pacotes.
9. No menu suspenso Adicionar configuração, escolha a configuração do pacote que você deseja usar com suas ações de fluxo de trabalho.
10. Escolha 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 obter mais informações sobre repositórios de pacotes, consulte [Repositórios de pacotes](#).

12. (Opcional) Em Escopos - opcional, especifique uma sequência de escopos que você deseja definir no registro do pacote.

Ao definir 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 ser prefixado com “@”.

Se Scopes for omitido, o repositório de pacotes especificado será configurado como o registro padrão para todos os pacotes usados pela ação.

Para obter mais informações sobre escopos, consulte [Namespaces de pacotes](#) Pacotes com [escopo definido](#).

13. Escolha Adicionar.
14. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
15. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Em uma ação, adicione um código semelhante ao seguinte:

```
action-name:  
  Configuration:  
    Packages:  
      NpmConfiguration:  
        PackageRegistries:
```

```
- PackagesRepository: package-repository
  Scopes:
    - "@scope"
```

Para obter mais informações, consulte a descrição da Packages propriedade em [Crie e teste a definição YAML da ação](#) Para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Exemplos de especificação de repositórios de pacotes em fluxos de trabalho

Os exemplos a seguir mostram como referenciar pacotes no arquivo de definição do fluxo de trabalho.

Tópicos

- [Exemplo: definição de pacotes com NpmConfiguration](#)
- [Exemplo: Substituindo o registro padrão](#)
- [Exemplo: substituição de escopos em seu registro de pacotes](#)

Exemplo: definição de pacotes com **NpmConfiguration**

O exemplo a seguir mostra como definir um pacote NpmConfiguration em seu arquivo de definição de 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 `main-repo` como definido sem um escopo. `@scope1` O escopo está configurado em `PackageRegistries` `for` `scoped-repo`.

Exemplo: Substituindo o 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 no 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 foi configurado `@scope2` `PackageRegistries` é `paramy-repo-2`. Isso substitui o escopo `@scope2` configurado para `my-repo-1`.

Invocando uma AWS Lambda função 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 de AWS Lambda invocação ao seu fluxo de trabalho. A ação de AWS Lambda invocação invoca a função Lambda que você especifica.

[Além de invocar sua função, a ação de AWS Lambda invocação também converte cada chave de nível superior na carga de resposta recebida da função Lambda em uma variável de saída do fluxo de trabalho.](#) Essas variáveis podem então ser referenciadas em ações subsequentes do fluxo de trabalho. Se você não quiser que todas as chaves de nível superior sejam convertidas em variáveis, você pode usar filtros para especificar as chaves exatas. Para obter mais informações, consulte [ResponseFilters](#) a descrição da propriedade no [definição YAML da ação “AWS Lambda invocar”](#).

Quando usar essa ação

Use essa ação se quiser adicionar funcionalidade ao seu fluxo de trabalho que é encapsulada e executada por uma função Lambda.

Por exemplo, talvez você queira que seu fluxo de trabalho envie uma `Build started` notificação para um canal do Slack antes de começar a compilar seu aplicativo. Nesse caso, seu fluxo de trabalho incluiria uma ação de AWS Lambda invocação para invocar um Lambda para enviar a notificação do Slack e uma [ação de criação para criar](#) seu aplicativo.

Como outro exemplo, talvez você queira que seu fluxo de trabalho realize uma verificação de vulnerabilidade em seu aplicativo antes de ser implantado. Nesse caso, você usaria uma ação de construção para criar seu aplicativo, uma ação de invocação para AWS Lambda invocar um Lambda para verificar vulnerabilidades e uma ação de implantação para implantar o aplicativo escaneado.

Tópicos

- [Exemplo de fluxo de trabalho que invoca uma função Lambda](#)
- [Adicionando a AWS Lambda ação “invocar”](#)
- [Variáveis produzidas pela AWS Lambda ação “invocar”](#)
- [definição YAML da ação “AWS Lambda invocar”](#)

Exemplo de fluxo de trabalho que invoca uma função Lambda

Note

O exemplo de fluxo de trabalho a seguir é apenas para fins ilustrativos e requer trabalho adicional de configuração para funcionar corretamente. O objetivo é dar um exemplo de como pode ser um fluxo de trabalho quando configurado com a ação de AWS Lambda invocação.

O fluxo de trabalho a seguir inclui a ação de AWS Lambda invocação, 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 AWS CloudFormation modelo. O fluxo de trabalho consiste nos seguintes blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação de AWS Lambda invocação (LambdaNotify) — No gatilho, essa ação invoca a função Notify-Start Lambda na AWS conta e região especificadas (e). my-aws-account us-west-2 Na invocação, a função Lambda envia uma notificação do Slack indicando que a implantação foi iniciada.
- Uma ação Deploy AWS CloudFormation stack (Deploy) — Ao concluir a ação de AWS Lambda invocação, a ação Deploy AWS CloudFormation stack executa o template (cfn-template.yml) para implantar sua pilha de aplicativos. Para obter mais informações sobre a ação Deploy AWS CloudFormation stack, consulte [Implantando uma AWS CloudFormation pilha com um fluxo de trabalho](#).

```
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 AWS Lambda ação “invocar”

Use as instruções a seguir para adicionar a ação de AWS Lambda invocação 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 obter mais informações, consulte o tópico da [função de execução do Lambda](#) no Guia do AWS Lambda desenvolvedor.

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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de AWS Lambda invocação e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha AWS Lambda invocar. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [definição YAML da ação “AWS Lambda invocar”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

12. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de AWS Lambda invocação e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

 - Escolha AWS Lambda invocar. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [definição YAML da ação “AWS Lambda invocar”](#).
11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Variáveis produzidas pela AWS Lambda ação “invocar”

Por padrão, a ação de AWS Lambda invocação produz uma variável por chave de nível superior na carga de resposta do Lambda.

Por exemplo, se a carga de resposta for semelhante a esta:

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

... então a ação geraria as seguintes variáveis.

| Chave | Valor |
|------------|--|
| name | Saanvi |
| local | Seattle |
| department | {"empresa": "Amazon", "equipe": "AWS"} |

Note

Você pode alterar quais variáveis são geradas usando a propriedade `ResponseFilters` YAML. Para obter mais informações, consulte o [ResponseFilters](#) no [definição YAML da ação “AWS Lambda invocar”](#).

As variáveis produzidas e definidas pela ação “AWS Lambda invocar” em tempo de execução são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

definição YAML da ação “AWS Lambda invocar”

A seguir está a definição YAML da ação de AWS Lambda invocação. Para saber como usar essa ação, consulte [Invocando uma AWS Lambda função 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: `Lambda_Invoke_Action_Workflow_nn`.

UI correspondente: guia de configuração/nome da ação

Identifier

(*LambdaInvoke*/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

Padrão: `aws/lambda-invoke@v1`.

UI correspondente: diagrama de fluxo de trabalho/ LambdaInvoke _nn/aws/lambda-invoke @v1 label

DependsOn

(*LambdaInvoke*/DependsOn)

(Optional)

Especifique uma ação, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*LambdaInvoke*/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(*LambdaInvoke*/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)

Otimizado para flexibilidade durante as corridas 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: guia de configuração/tipo de computação

Fleet

(*LambdaInvoke*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

UI correspondente: guia de 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é o mesmo que o tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Inputs

(*LambdaInvoke*/Inputs)

(Obrigatório)

A `Inputs` seção define os dados que a ação de AWS Lambda invocação precisa durante a execução de um fluxo de trabalho.

Note

Somente uma entrada (uma fonte ou um artefato) é permitida por AWS Lambda ação de invocação. As variáveis não contam para esse total.

UI correspondente: guia Entradas

Sources

(*LambdaInvoke*/Inputs/Sources)

(Obrigatório se [RequestPayloadFile](#) for fornecido)

Se você quiser passar um arquivo JSON de carga útil de solicitação para a ação de AWS Lambda invocação, 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 da solicitação não estiver contido em um repositório de origem, ele deverá residir em um artefato gerado por outra ação.

Para obter mais informações sobre o arquivo de carga, consulte [RequestPayloadFile](#).

Note

Em vez de especificar um arquivo de carga, você pode adicionar o código JSON da carga diretamente à ação usando a propriedade `RequestPayload`. Para ter mais informações, consulte [RequestPayload](#).

Para obter mais informações sobre fontes, consulte [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - opcional

Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Obrigatório se [RequestPayloadFile](#) for fornecido)

Se você quiser passar um arquivo JSON de carga útil de solicitação para a ação de AWS Lambda invocação, e esse arquivo de carga estiver contido em um artefato de [saída de uma ação anterior](#), [especifique esse artefato](#) aqui.

Para obter mais informações sobre o arquivo de carga, consulte [RequestPayloadFile](#).

Note

Em vez de especificar um arquivo de carga, você pode adicionar o código JSON da carga diretamente à ação usando a propriedade. `RequestPayload` Para ter mais informações, consulte [RequestPayload](#).

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Optional)

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de entradas/variáveis - opcional

Environment

(*LambdaInvoke*/Environment)

(Obrigatório)

Especifique o CodeCatalyst ambiente a ser usado com a ação.

Para obter mais informações sobre ambientes, consulte [Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst Criar um ambiente](#) e.

UI correspondente: guia de configuração/'Ambiente/Conta/Função'/Ambiente

Name

(*LambdaInvoke*/Environment/Name)

(Obrigatório se [Environment](#) estiver incluído)

Especifique o nome de um ambiente existente que você deseja associar à ação.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Ambiente

Connections

(*LambdaInvoke*/Environment/Connections)

(Obrigatório se [Environment](#) estiver incluído)

Especifique a conexão da conta a ser associada à ação. Você pode especificar no máximo uma conexão de conta em `Environment`.

Para obter mais informações sobre conexões de conta, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#). Para obter informações sobre como associar uma conexão de conta ao seu ambiente, consulte [Criar um ambiente](#).

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Conexão

Name

(*LambdaInvoke*/Environment/Connections/Name)

(Obrigatório)

Especifique o nome da conexão da conta.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Conexão

Role

(*LambdaInvoke*/Environment/Connections/Role)

(Obrigatório)

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 que essa função inclua:

- A seguinte política de permissões:

Warning

Limite as permissões às mostradas na política a seguir. Usar uma função com permissões mais amplas pode representar um risco de segurança.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:aws-account:function:function-
name"
    }
  ]
}
```

- A seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Verifique se essa função está associada à conexão da sua conta. Para saber mais sobre como associar uma função do IAM a uma conexão de conta, consulte [Adicionar funções do IAM às conexões da conta](#).

Note

Você pode especificar o nome da `CodeCatalystWorkflowDevelopmentRole-spaceName` função aqui, se quiser. Para obter mais informações sobre essa função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço](#). Entenda que a `CodeCatalystWorkflowDevelopmentRole-spaceName` função tem permissões muito amplas que podem representar um risco de segurança. Recomendamos que você use essa função somente em tutoriais e cenários em que a segurança seja menos preocupante.

UI correspondente: guia Configuração/'Ambiente/Conexão/Função'/Função

Configuration

(*LambdaInvoke*/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.


UI 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 seu Amazon Resource Name (ARN). Você pode encontrar o nome ou o ARN no console Lambda.

 Note

A AWS conta na qual a função Lambda reside pode ser diferente da conta especificada em. `Connections`:

UI correspondente: guia/função de configuração

AWSRegion

(*LambdaInvoke*/Configuration/AWSRegion)

(Obrigatório)

Especifique a AWS região em que sua AWS Lambda função reside. Para obter uma lista de códigos de região, consulte [Endpoints regionais](#) no Referência geral da AWS.

UI correspondente: guia de configuração/bucket de destino - opcional

RequestPayload

(*LambdaInvoke*/Configuration/RequestPayload)

(Optional)

Se você quiser passar uma carga de solicitação para a ação de AWS Lambda invocação, especifique a carga útil da solicitação aqui, no formato JSON.

Exemplo de carga útil de solicitação:

```
'{ "key": "value" }'
```

Se você não quiser passar uma carga de solicitação para sua função Lambda, omita essa propriedade.

Note

Você pode especificar `RequestPayload` ou `RequestPayloadFile`, mas não os dois.

Para obter mais informações sobre a carga útil da solicitação, consulte o tópico [Invocar](#) na Referência da AWS Lambda API.

UI correspondente: guia de configuração/carga útil de solicitação - opcional

RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Se você quiser passar uma carga de solicitação para a ação de AWS Lambda invocação, especifique o caminho para esse arquivo de carga de solicitação aqui. O arquivo deve 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 passar uma carga de solicitação para sua função Lambda, omita essa propriedade.

Note

Você pode especificar `RequestPayload` ou `RequestPayloadFile`, mas não os dois.

Para obter mais informações sobre o arquivo de carga útil da solicitação, consulte o tópico [Invocar](#) na Referência da AWS Lambda API.

UI correspondente: guia de configuração/arquivo de carga útil de solicitação - opcional

ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Especifique se você deseja marcar a ação de AWS Lambda invocação como bem-sucedida, mesmo que a função invocada AWS Lambda falhe. Considere definir essa propriedade `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 Lambda falhar (“desativada” no editor visual ou `false` no editor YAML).

UI correspondente: guia de configuração/ Continuar em caso de erro

LogType

(*LambdaInvoke*/Configuration/LogType)

(Optional)

Especifique se você deseja incluir registros de erros na resposta da função 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`— registros de devolução
- `None`— não retorne registros

O padrão é `Tail`.

Para obter mais informações sobre o tipo de registro, consulte o tópico [Invocar](#) na Referência da AWS Lambda API.

Para obter mais informações sobre como visualizar logs, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

UI correspondente: guia de configuração/tipo de log

ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Optional)

Especifique quais chaves na carga de resposta do Lambda você deseja converter em variáveis de saída. Em seguida, você pode referenciar as variáveis de saída em ações subsequentes em seu fluxo de trabalho. Para obter mais informações sobre variáveis em CodeCatalyst, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

Por exemplo, se sua carga de resposta for semelhante a esta:

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

... e seus filtros de resposta têm a seguinte aparência:

```
Configuration:
  ...
  ResponseFilters: '{"name": ".name", "company": ".department.company"}'
```

... então a ação gera as seguintes variáveis de saída:

| Chave | Valor |
|---------|--------|
| name | Saanvi |
| company | Amazon |

Em seguida, você pode referenciar as company variáveis name e em ações subsequentes.

Se você não especificar nenhuma chave `ResponseFilters`, a ação converterá cada chave de nível superior na resposta do Lambda em uma variável de saída. Para ter mais informações, consulte [Variáveis produzidas pela AWS Lambda ação “invocar”](#).

Considere usar filtros de resposta para limitar as variáveis de saída geradas somente às que você realmente deseja usar.

UI correspondente: guia de 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.

UI 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia Saídas/Artefatos/Nome do artefato de construção

Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Optional)

Especifique o nome do artefato que conterá a carga de resposta do Lambda que é retornada pela função 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 [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

UI correspondente: guia Saídas/Artefatos/Nome do artefato de construção

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.

UI correspondente: guia de saídas/artefatos/arquivos produzidos pela compilação

Modificar um arquivo de definição de tarefas do Amazon ECS usando um fluxo de trabalho

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 de definição de tarefas `Render Amazon ECS` ao seu fluxo de trabalho. Essa ação atualiza o campo de imagem no arquivo de definição da tarefa com um nome de imagem do Docker fornecido pelo seu fluxo de trabalho em tempo de execução.

Note

Você também pode usar essa ação para atualizar o `environment` campo da definição da tarefa com variáveis de ambiente.

Quando usar essa ação

Use isso 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 contiver um valor de imagem que sempre permaneça o mesmo. Nesse caso, você pode inserir manualmente o nome da sua imagem no arquivo de definição da tarefa.

Como funciona a ação “Renderizar a definição de tarefas do Amazon ECS”

Você deve usar a ação de definição de tarefas Render Amazon ECS com as ações de compilação e implantação 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 ter a seguinte aparência:

```
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 `docker build -t` diretiva indica criar a imagem do Docker e marcá-la com o ID do commit no tempo de execução da ação. O nome da imagem gerada pode ter a seguinte aparência:

```
MyDockerImage:a37bd7e
```

2. A ação de definição de tarefas Render 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 de definição de tarefas Render Amazon ECS adicione variáveis de ambiente à definição da tarefa, da seguinte forma:

```

{
  "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 obter mais informações sobre variáveis de ambiente, consulte [Especificação de variáveis de ambiente](#) no Amazon Elastic Container Service Developer Guide.

3. A ação Deploy to 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 no MyDockerImage:a37bd7e Amazon ECS.

Tópicos

- [Exemplo de fluxo de trabalho que modifica um arquivo de definição de tarefas do Amazon ECS](#)
- [Adicionando a ação “Render Amazon ECS task definition”](#)
- [Visualizando o arquivo de definição de tarefa atualizado](#)
- [Variáveis produzidas pela ação “Render Amazon ECS task definition”](#)
- [Referência de definição YAML da ação “Render Amazon ECS task definition”](#)

Exemplo de fluxo de trabalho que modifica um arquivo de definição de tarefas do Amazon ECS

Veja a seguir um exemplo de um fluxo de trabalho completo que inclui a ação de definição de tarefas Render Amazon ECS, junto com ações de criação e implantação. O objetivo do fluxo de trabalho é criar e implantar uma imagem Docker em seu cluster Amazon ECS. O fluxo de trabalho consiste nos seguintes blocos de construção 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 obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).
- Uma ação de criação (BuildDocker) — No gatilho, a ação cria a imagem do Docker usando o Dockerfile, a marca com um ID de confirmação e envia a imagem para o Amazon ECR. Para obter mais informações sobre a ação de criação, consulte [Construindo com fluxos de trabalho](#).
- Uma ação de definição de tarefa Render Amazon ECS (RenderTaskDef) — Ao concluir a ação de criação, essa ação atualiza uma existente `taskdef.json` localizada na raiz do seu repositório de origem com um valor de `image` campo que inclui o ID de confirmação correto. Ele salva o arquivo atualizado com um novo nome de arquivo (`task-definition-random-string.json`) e, em seguida, 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 de definição da tarefa Render Amazon ECS, a ação Implantar no Amazon ECS procura o artefato de saída gerado pela ação de renderização (`TaskDefArtifact`), encontra o `task-definition-random-string.json` arquivo dentro dele e o registra no seu serviço Amazon ECS. Em seguida, o serviço Amazon ECS segue as instruções no `task-definition-random-string.json` arquivo para executar as tarefas do Amazon ECS — e os contêineres de imagem 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}
```

Adicionando a ação “Render Amazon ECS task definition”

Use as instruções a seguir para adicionar a ação de definição de tarefas Render Amazon ECS ao seu fluxo de trabalho.

Pré-requisito

Antes de começar, certifique-se de ter um fluxo de trabalho que inclua uma ação de criação que gere dinamicamente uma imagem do Docker. Consulte o [exemplo anterior de fluxo de trabalho](#) para obter detalhes.

Visual

Para adicionar a ação “Render Amazon ECS task definition” 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de definição de tarefa Render Amazon ECS e faça o seguinte:
 - Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Render a definição de tarefa do Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 obter uma descrição de cada campo, consulte [Referência de definição YAML da ação “Render Amazon ECS task definition”](#) o. 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) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

YAML

Para adicionar a ação “Render Amazon ECS task definition” 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. No canto superior esquerdo, escolha + Ações para abrir o catálogo de ações.
8. Na lista suspensa, escolha Amazon. CodeCatalyst
9. Pesquise a ação de definição de tarefa Render Amazon ECS e faça o seguinte:

- Escolha o sinal de adição (+) para adicionar a ação ao diagrama do fluxo de trabalho e abrir seu painel de configuração.

Ou

- Escolha Render a definição de tarefa do Amazon ECS. A caixa de diálogo de detalhes da ação é exibida. Nessa caixa de diálogo:
 - (Opcional) Escolha Exibir fonte para [ver o código-fonte da ação](#).
 - Escolha 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 suas necessidades. Uma explicação de cada propriedade disponível é fornecida no [Referência de definição YAML da ação "Render Amazon ECS task definition"](#).
11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Próximas etapas

Depois de adicionar a ação de renderização, adicione a ação Deploy to Amazon ECS ao seu fluxo de trabalho seguindo as instruções em [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) 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 [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

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* é 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 obter mais informações sobre variáveis, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

Para obter mais informações sobre como configurar a ação de implantação, consulte o [exemplo anterior de fluxo](#) de trabalho.

Visualizando o 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 de definição de tarefa Render 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, em seguida, escolha Fluxos de trabalho.
 - d. Escolha o nome do fluxo de trabalho que contém a ação de renderização. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
 - e. Escolha uma execução que inclua a ação de renderização concluída.
2. No diagrama do fluxo de trabalho, escolha a ação de renderização.
 3. Escolha Saídas.
 4. Escolha Variáveis.
 5. O nome do arquivo de definição da tarefa é exibido. Parece semelhante `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, em seguida, escolha Fluxos de trabalho.
 - d. Escolha o nome do fluxo de trabalho que contém a ação de renderização. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome 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, escolha Saídas do fluxo de trabalho.
3. Na seção Artefatos, escolha 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 ver o arquivo.json de definição da tarefa.

Variáveis produzidas pela ação “Render Amazon ECS task definition”

A ação de definição de tarefas Render Amazon ECS produz e define as seguintes variáveis em tempo de execução. Elas são conhecidas como variáveis predefinidas.

Para obter informações sobre como referenciar essas variáveis em um fluxo de trabalho, consulte [Usando variáveis predefinidas](#).

| Chave | Valor |
|---------------------|---|
| definição de tarefa | <p>O nome dado ao arquivo de definição de tarefas que foi atualizado pela ação de definição de tarefas Render Amazon ECS. 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> |

Referência de definição YAML da ação “Render Amazon ECS task definition”

A seguir está a definição YAML da ação de definição de tarefas Render Amazon ECS. Para saber como usar essa ação, consulte [Modificar um arquivo de definição de tarefas do Amazon ECS 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 de fluxo de trabalho YAML](#).

Note

A maioria das propriedades YAML a seguir tem elementos de interface de usuário correspondentes no editor visual. Para pesquisar um elemento de interface do usuário, use Ctrl+F. O elemento será listado com sua 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 das ações devem ser exclusivos no fluxo de trabalho. Os nomes das ações são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes das ações.

Padrão: `ECSRenderTaskDefinition_nn`.

UI correspondente: guia de configuração/nome da ação

Identifier

(ECSRenderTaskDefinition/Identifier)

(Obrigatório)

Identifica a ação. Não altere essa propriedade, a menos que você queira alterar a versão. Para ter mais informações, consulte [Especificar a versão principal, secundária ou de patch de uma ação](#).

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, grupo de ação ou porta que deve ser executada com êxito para que essa ação seja executada.

Para obter mais informações sobre a funcionalidade “depende de”, consulte. [Configurando ações para depender de outras ações](#)

UI correspondente: guia Entradas/Depende de - opcional

Compute

(*ECSRenderTaskDefinition*/Compute)

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos. Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Type

(*ECSRenderTaskDefinition*/Compute/Type)

(Obrigatório se [Compute](#) estiver incluído)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)
Otimizado para flexibilidade durante as corridas 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: guia de configuração/tipo de computação

Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 é concluída. Exemplos de frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte [Propriedades da frota provisionada](#)

Se `Fleet` for omitido, o padrão será `Linux.x86-64.Large`

UI correspondente: guia de 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 5 minutos e o máximo está descrito em [Cotas para fluxos de trabalho](#). O tempo limite padrão é igual ao tempo limite máximo.

UI correspondente: guia de configuração/tempo limite - opcional

Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Optional)

A `Inputs` seção define os dados `ECSRenderTaskDefinition` necessários durante a execução de um fluxo de trabalho.

Note

Somente uma entrada (uma fonte ou um artefato) é permitida por ação de definição de tarefa do Render Amazon ECS. As variáveis não contam para esse total.

UI 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 desse repositório de origem. Atualmente, o único rótulo compatível é `WorkflowSource`.

Se seu arquivo de definição de tarefa 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 [Conectando um fluxo de trabalho a um repositório de origem](#).

UI correspondente: guia Entradas/Fontes - 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 contido em um artefato gerado por uma ação anterior, especifique esse artefato aqui. Se o arquivo de definição de tarefas não estiver contido em um artefato, ele deverá residir no seu repositório de origem.

Para obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI correspondente: guia de configuração/artefatos - opcional

Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Obrigatório)

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífen (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para habilitar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de entradas/variáveis - opcional

Configuration

(ECSRenderTaskDefinition/Configuration)

(Obrigatório)

Uma seção na qual você pode definir as propriedades de configuração da ação.

UI 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 no 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 obter mais informações sobre arquivos de definição de tarefas, consulte [Definições de tarefas](#) no Amazon Elastic Container Service Developer Guide.

UI correspondente: guia de 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 no nome campo `containerDefinitions`, em seu arquivo de definição de tarefa. Para obter mais informações, consulte [Nome](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

UI correspondente: guia de configuração/nome do contêiner

image

(*ECSRenderTaskDefinition*/Configuration/image)

(Obrigatório)

Especifique o nome da imagem do Docker que você deseja que a ação de definição de tarefas Render Amazon ECS adicione ao seu arquivo de definição de tarefas. A ação adiciona esse nome ao `image` campo `containerDefinitions`, em seu arquivo de definição de tarefa. Se um valor já existir no `image` campo, 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, em que *commit-id* é 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` está o URI de um Amazon Elastic Container Registry (ECR) e `date +%m-%d-%y-%h-%m-%s` é um timestamp no formato gerado em tempo de execução pelo fluxo de trabalho. month-day-year-hour-minute-second

Para obter mais informações sobre o `image` campo, consulte a [imagem](#) no Amazon Elastic Container Service Developer Guide. Para obter mais informações sobre variáveis, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

UI correspondente: guia de configuração/nome da imagem

environment-variables

(*ECSRenderTaskDefinition*/Configuration/environment-variables)

(Obrigatório)

Especifique as variáveis de ambiente que você deseja que a ação de definição de tarefas Render Amazon ECS adicione ao seu arquivo de definição de tarefas. A ação adiciona as variáveis ao `environment` campo `containerDefinitions`, 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 obter mais informações sobre as variáveis de ambiente do Amazon

ECS, consulte [Especificação de variáveis de ambiente](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

UI correspondente: guia de 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.

UI 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 obter mais informações sobre artefatos, incluindo exemplos, consulte [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#).

UI 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 Deploy to Amazon ECS. Para entender como adicionar esse artefato como entrada para a ação Deploy to Amazon ECS, consulte [Exemplo de fluxo de trabalho que modifica um arquivo de definição de tarefas do Amazon ECS](#)

UI 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.

UI 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 de definição de tarefa (editor visual) ou **task-definition** (editor yaml) da ação Deploy to Amazon ECS. Para entender como adicionar essa variável à ação Deploy to Amazon ECS, consulte [Exemplo de fluxo de trabalho que modifica um arquivo de definição de tarefas do Amazon ECS](#).

Padrão: `task-definition`

UI correspondente: guia de saídas/variáveis/campo de nome

Configurando e usando variáveis em um fluxo de trabalho

Uma variável é um par de valores-chave que contém informações que você pode referenciar em seu CodeCatalyst fluxo de trabalho.

Há dois tipos de variáveis que você pode usar em um fluxo de trabalho:

- Variáveis definidas pelo usuário — Esses são pares de valores-chave que você define.
- Variáveis predefinidas — Esses são pares de valores-chave que são emitidos automaticamente por um fluxo de trabalho. Não há necessidade de você defini-las.

Note

CodeCatalyst também suporta [parâmetros GitHub de saída](#), que se comportam como variáveis e podem ser referenciados em outras ações. Para obter mais informações, consulte [Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo](#) e [Fazendo referência a um parâmetro GitHub de saída](#)

Tópicos

- [Usando variáveis definidas pelo usuário](#)
- [Usando variáveis predefinidas](#)
- [Lista de variáveis predefinidas](#)

Usando variáveis definidas pelo usuário

As variáveis definidas pelo usuário são pares de valores-chave que você define. Há dois tipos:

- Variáveis de texto simples, ou simplesmente variáveis — Esses são pares de valores-chave que você define em texto sem formatação 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. Você só especifica 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

Para resumir, este guia usa o termo variável para significar variável de texto simples.

Tópicos

- [Definindo uma variável](#)
- [Definindo um segredo](#)
- [Exportar uma variável para que outras ações possam usá-la](#)
- [Fazendo referência a uma variável na ação que a define](#)

- [Fazendo referência a uma saída variável por outra ação](#)
- [Referenciando um segredo](#)
- [Exemplos de variáveis definidas pelo usuário](#)

Definindo uma variável

Você pode definir variáveis de duas maneiras:

- Na Inputs seção de uma ação de fluxo de trabalho — consulte [Para definir uma variável na seção “Entradas”](#)
- Na Steps seção 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 ver exemplos, consulte [Exemplos de variáveis definidas pelo usuário](#).

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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação em que você deseja definir a variável.
8. Escolha Entradas.
9. Em Variáveis - opcional, escolha Adicionar variável e, em seguida, faça o seguinte:

Especifique uma sequência de pares de nome/valor que definam as variáveis de entrada que você deseja disponibilizar para a ação. Os nomes das variáveis são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos espaços. Você não pode usar aspas para habilitar caracteres especiais e espaços em nomes de variáveis.

Para obter mais informações sobre variáveis, incluindo exemplos, consulte [Configurando e usando variáveis em um fluxo de trabalho](#).

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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 definidas pelo usuário](#). Para obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.

9. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação em que você deseja definir 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 Steps seção.
 - Para definir uma variável implicitamente, especifique-a em um arquivo referenciado na seção da Steps ação.

Para ver exemplos, consulte [Exemplos de variáveis definidas pelo usuário](#). Para obter mais informações, consulte a [Definição de fluxo de trabalho YAML](#) seção Para a ação.

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Em uma ação de fluxo de trabalho, defina uma variável na Steps seção da ação, explícita ou implicitamente.
 - Para definir a variável explicitamente, inclua-a em um comando bash diretamente na Steps seção.
 - Para definir uma variável implicitamente, especifique-a em um arquivo referenciado na seção da Steps ação.

Para ver exemplos, consulte [Exemplos de variáveis definidas pelo usuário](#). Para obter mais informações, consulte a [Definição de fluxo de trabalho YAML](#) seção Para a ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Definindo um segredo

Você define um segredo na página Segredos do CodeCatalyst console. Para ter mais informações, consulte [Configurando e usando segredos em um fluxo de trabalho](#).

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: referenciando um segredo](#).

Exportar 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, observe 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 oferecem suporte à exportação de variáveis. Para determinar se sua ação oferece suporte a 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 suportada.
- Para exportar uma variável de uma GitHub Ação, consulte [Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo](#).

Pré-requisito

Verifique se você definiu a variável que deseja exportar. Para ter mais informações, consulte [Definindo 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha a ação da qual você deseja exportar a variável.
8. Escolha Saídas.
9. Em Variáveis - opcional, escolha Adicionar variável e, em seguida, faça o seguinte:

Especifique o nome de uma variável que você deseja que a ação exporte. Essa variável já deve estar definida na Steps seção Inputs ou da mesma ação.

10. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
11. Escolha 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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 definidas pelo usuário](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Fazendo 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 [Fazendo referência a um parâmetro GitHub de saída](#).

Pré-requisito

Certifique-se de ter definido a variável que você deseja referenciar. Para ter mais informações, consulte [Definindo uma variável](#).

Visual

Não disponível. Escolha YAML para ver as instruções do 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha 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 definidas pelo usuário](#). Para obter mais informações, consulte as informações de referência para sua ação no [Definição de fluxo de trabalho YAML](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Fazendo referência a uma saída variável por outra ação

Use as instruções a seguir para referenciar variáveis geradas por outras ações.

Note

Para referenciar uma saída variável de uma GitHub Ação, consulte [Fazendo referência a um parâmetro GitHub de saída](#).

Pré-requisito

Certifique-se de ter exportado a variável que você deseja referenciar. Para ter mais informações, consulte [Exportar uma variável para que outras ações possam usá-la](#).

Visual

Não disponível. Escolha YAML para ver as instruções do YAML.

YAML

Para referenciar uma saída 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.

- Escolha YAML.
- 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 definidas pelo usuário](#). Para obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) para sua ação.

- (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
- Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Referenciando um segredo

Para obter instruções sobre como fazer referência a um segredo no arquivo de definição do fluxo de trabalho, consulte [Usando um segredo](#).

Para ver um exemplo, consulte [Exemplo: referenciando um segredo](#).

Exemplos de variáveis definidas pelo usuário

Os exemplos a seguir mostram como definir e referenciar variáveis no arquivo de definição do fluxo de trabalho.

Exemplos

- [Exemplo: definição de uma variável usando a propriedade Inputs](#)
- [Exemplo: definição de uma variável usando a propriedade Steps](#)
- [Exemplo: exportar uma variável usando a propriedade Outputs](#)
- [Exemplo: referenciando uma variável definida na mesma ação](#)
- [Exemplo: referenciando uma variável definida em outra ação](#)
- [Exemplo: referenciando um segredo](#)

Exemplo: definição de uma variável usando a propriedade Inputs

O exemplo a seguir mostra como definir duas variáveis VAR1 e VAR2, em uma Inputs seção.

```
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 Steps

O exemplo a seguir mostra como definir explicitamente uma DATE variável na Steps seção.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: DATE=$(date +%m-%d-%y)
```

Exemplo: exportar uma variável usando a propriedade Outputs

O exemplo a seguir mostra como definir duas variáveis REPOSITORY-URI e TIMESTAMP exportá-las usando a Outputs seção.

```

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: referenciando uma variável definida na mesma ação

O exemplo a seguir mostra como especificar uma VAR1 variável em eMyBuildAction, em seguida, 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: referenciando uma variável definida em outra ação

O exemplo a seguir mostra como especificar uma TIMESTAMP variável emBuildActionA, exportá-la usando a Outputs propriedade e, em seguida, 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: referenciando um segredo

O exemplo a seguir mostra como referenciar um `my-password` segredo. Essa `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 ter mais informações, consulte [Configurando e usando segredos em um fluxo de trabalho](#).

```

Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: curl -u LiJuan:${Secrets.my-password} https://example.com

```

Usando variáveis predefinidas

Variáveis predefinidas são pares de valores-chave que são emitidos automaticamente por um fluxo de trabalho e disponibilizados para você usar em ações de fluxo de trabalho.

Você pode usar variáveis predefinidas em qualquer ação do fluxo de trabalho.

Tópicos

- [Referenciando uma variável predefinida](#)
- [Determinar quais variáveis predefinidas seu fluxo de trabalho emite](#)
- [Exemplos de variáveis predefinidas](#)

Referenciando uma variável predefinida

Use as instruções a seguir para referenciar uma variável predefinida.

Pré-requisito

Determine o nome da variável predefinida que você deseja referenciar, como `CommitId`. Para ter mais informações, consulte [Determinar quais variáveis predefinidas seu fluxo de trabalho emite](#).

Visual

Não disponível. Escolha YAML para ver as instruções do YAML.

YAML

Para referenciar uma variável predefinida (editor YAML)

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Selecione o projeto.
3. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Em uma CodeCatalyst ação, adicione a referência de variável predefinida usando a seguinte sintaxe:

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```

Substitua:

- *action-group-name* com o 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* com:

O nome da ação que gera a variável.

ou

WorkflowSource, se a variável for a CommitId variável BranchName ou.

- *variable-name* com o 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, consulte [Exemplos de variáveis predefinidas](#). Para obter mais informações, consulte [Definição de fluxo de trabalho YAML](#) para sua ação.

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Determinar quais variáveis predefinidas seu fluxo de trabalho emite

Você pode determinar quais variáveis predefinidas seu fluxo de trabalho emite de duas maneiras:

- 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 ter mais informações, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).
- Consulte [Lista de variáveis predefinidas](#) o. 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](#). Se o tamanho total exceder o máximo, a ação que ocorre após o máximo ser atingido poderá falhar.

Exemplos de variáveis predefinidas

Os exemplos a seguir mostram como referenciar variáveis predefinidas no arquivo de definição do fluxo de trabalho.

Exemplos

- [Exemplo: referenciando a variável predefinida CommitId](#) ""
- [Exemplo: referenciando a variável predefinida BranchName](#) ""

Exemplo: referenciando a variável predefinida CommitId ""

O exemplo a seguir mostra como se referir à variável CommitId predefinida na MyBuildAction ação. A CommitId variável é gerada automaticamente por CodeCatalyst.

Embora o exemplo mostre a variável que está sendo usada na ação de criação, você pode CommitId usá-la 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: referenciando a variável predefinida BranchName ""

O exemplo a seguir mostra como se referir à variável BranchName predefinida na CDKDeploy ação. A BranchName variável é gerada automaticamente por CodeCatalyst.

Embora o exemplo mostre a variável que está sendo usada na ação de AWS CDK implantação, você pode BranchName usá-la em qualquer ação.

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    StackName: app-stack-${WorkflowSource.BranchName}
```

Lista de variáveis predefinidas

Consulte as seções a seguir para ver as variáveis predefinidas produzidas automaticamente pelas CodeCatalyst ações.

Note

[Essa lista inclui apenas variáveis predefinidas emitidas pela CodeCatalyst fonte e CodeCatalyst pelas ações.](#) Se você estiver usando outros tipos de ações, como GitHub ações do Actions ou CodeCatalyst Labs, consulte em vez disso [Determinar quais variáveis predefinidas seu fluxo de trabalho emite.](#)

Lista

Note

Nem todas as CodeCatalyst ações produzem variáveis predefinidas. Se a ação não estiver na lista, ela não produzirá variáveis.

- [Variáveis produzidas pela fonte \(" BranchName "e" CommidId "\)](#)
- [Variáveis produzidas pela ação "Deploy AWS CloudFormation stack"](#)
- [Variáveis produzidas pela ação "Deploy to Amazon ECS"](#)
- [Variáveis produzidas pela ação "Implantar no cluster Kubernetes"](#)
- [Variáveis produzidas pela ação "AWS CDK implantar"](#)
- [Variáveis produzidas pela ação "AWS CDK bootstrap"](#)
- [Variáveis produzidas pela AWS Lambda ação "invocar"](#)
- [Variáveis produzidas pela ação "Render Amazon ECS task definition"](#)

Configurando e usando segredos em um fluxo de trabalho

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

- [Criando um segredo](#)
- [Editando um segredo](#)
- [Usando um segredo](#)
- [Excluindo um segredo](#)

Criando um segredo

Use o procedimento a seguir para criar um segredo. O segredo contém as informações confidenciais que você deseja ocultar.

Note

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

Para criar um segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Segredos.
3. Escolha Create secret (Criar segredo).
4. Insira as seguintes informações:

Nome

Insira um nome para o seu segredo.

Valor

Insira o valor do segredo. Essas são as informações confidenciais que você deseja ocultar. Por padrão, o valor não é exibido. Para exibir o valor, escolha Mostrar valor.

Descrição

(Opcional) Insira uma descrição para seu segredo.

5. Escolha Create (Criar).

Editando um segredo

Use o procedimento a seguir para editar um segredo.

Para editar um segredo

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

3. Na lista de segredos, escolha o segredo que você deseja editar.
4. Selecione a opção Editar.
5. Edite as seguintes propriedades:

Valor

Insira o valor do segredo. Esse é o valor que você deseja ocultar da exibição. Por padrão, o valor não é exibido.

Descrição

(Opcional) Insira uma descrição para seu segredo.

6. Escolha Salvar.

Usando um segredo

Para usar um segredo em uma ação de fluxo de trabalho, você deve obter o identificador de referência do segredo e usar esse identificador na ação do fluxo de trabalho.

Tópicos

- [Obtendo o identificador de um segredo](#)
- [Fazendo referência a um segredo em um fluxo de trabalho](#)

Obtendo o identificador de um segredo

Use o procedimento a seguir para obter o identificador de referência do segredo. Você adicionará esse identificador ao seu fluxo de trabalho.

Para obter o identificador de referência do segredo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha CI/CD e, em seguida, escolha Segredos.
3. Na lista de segredos, encontre o segredo que você deseja usar.
4. Na coluna ID de referência, copie o identificador do segredo. A seguir está a sintaxe da ID de referência:

```
#{Secrets.<name>}
```


Fazendo referência a um segredo em um fluxo de trabalho

Use o procedimento a seguir para referenciar um segredo em um fluxo de trabalho.

Para referenciar um segredo

1. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
2. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
3. Selecione a opção Editar.
4. Escolha 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 `curl` comando, você usaria um Run comando semelhante ao seguinte:

```
- Run: curl -u <username-secret-identifier>:<password-secret-identifier> https://example.com
```

6. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
7. Escolha Confirmar, insira uma mensagem de confirmação e escolha Confirmar novamente.

Excluindo um segredo

Use o procedimento a seguir para excluir um segredo e o identificador de referência secreto.

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.

Para 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, em seguida, escolha Fluxos de trabalho.

3. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
4. Selecione a opção Editar.
5. Escolha YAML.
6. Pesquise no fluxo de trabalho a seguinte sequência de caracteres:

```
`${Secrets}.
```

Isso encontra todos os identificadores de referência de todos os segredos.

7. Exclua o identificador de referência do segredo escolhido ou substitua-o por um valor de texto simples.
8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha 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, em seguida, escolha Segredos.
3. Na lista de segredos, escolha o segredo que você deseja excluir.
4. Escolha Excluir.
5. Digite **delete** para confirmar a exclusão.
6. Escolha Excluir.

Visualizando o status do 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 ter mais informações, consulte [Visualizando o status e os detalhes da execução do fluxo de trabalho](#).

Para obter uma lista dos possíveis estados do fluxo de trabalho, consulte [Estados do fluxo de trabalho](#).

Para ver 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, em seguida, escolha Fluxos de trabalho.
4. Encontre o fluxo de trabalho cujo status você deseja visualizar. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.

O status é exibido com o fluxo de trabalho na lista.

5. (Opcional) Escolha o nome do fluxo de trabalho e localize o campo Definição do fluxo de trabalho. Ele mostra o status do fluxo de trabalho.


Cotas para fluxos de trabalho

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 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 de um valor de variável de saída | 500 caracteres ou mais, dependendo da ação que emite o valor.

<div data-bbox="829 1121 1507 1339"><p> Note
Os valores podem ser truncados se excederem o limite da ação.</p></div> |
| Número máximo de dias para manter os artefatos gerados durante a execução de um 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 resultados 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 | Para as ações de criação e teste, o tempo limite é de 8 horas.

Para todas as outras ações, o tempo limite é de 1 hora. |
| Número máximo de ambientes associados a um 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

A execução de um fluxo de trabalho pode estar em um dos seguintes estados:

- Bem-sucedido — A execução do fluxo de trabalho foi processada com êxito.
- Falha — Uma ou mais ações na execução do fluxo de trabalho falharam.
- Em andamento — A execução do fluxo de trabalho está sendo processada no momento.

- Parado — Uma pessoa interrompeu 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 substituído](#). A execução do fluxo de trabalho foi cancelada CodeCatalyst porque uma execução posterior do fluxo de trabalho a substituiu.

Estados do fluxo de trabalho

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, as duas condições a seguir devem ser verdadeiras:

- O arquivo de definição do fluxo de trabalho deve ser válido.
- O fluxo de trabalho não deve ter acionadores, acionadores push ou um gatilho push que seja executado usando os arquivos na ramificação atual. Para ter mais informações, consulte [Ação e considerações ao ramificar](#).
- Não válido — O arquivo de definição do fluxo de trabalho não é válido. O fluxo de trabalho não pode ser executado manualmente ou automaticamente por meio de acionadores. *Os fluxos de trabalho que não são válidos aparecem com uma **definição de fluxo de trabalho com uma mensagem de erro** (ou similar) no CodeCatalyst console.*

Para que um fluxo de trabalho seja marcado como inválido, a seguinte condição deve ser verdadeira:

- O arquivo de definição do fluxo de trabalho deve estar configurado incorretamente.

Para corrigir um arquivo de definição de fluxo de trabalho mal configurado, consulte [Como faço para corrigir os erros “A definição do fluxo de trabalho não tem erros”?](#)

- Inativo — A definição do fluxo de trabalho é válida, mas não pode ser executada manualmente ou automaticamente por meio de acionadores.

Para que um fluxo de trabalho seja marcado como inativo, as duas condições a seguir devem ser verdadeiras:

- O arquivo de definição do fluxo de trabalho deve ser válido.
- O arquivo de definição do fluxo de trabalho deve incluir um acionador push que especifique uma ramificação diferente daquela em que o arquivo de definição do fluxo de trabalho está atualmente. Para ter mais informações, consulte [Ação e considerações ao ramificar](#).

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 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 de fluxo de trabalho YAML

A seguir está a documentação de referência para o arquivo de definição do fluxo de trabalho.

Um arquivo de definição de fluxo de trabalho é um arquivo YAML que descreve seu fluxo de trabalho. O arquivo é armazenado em uma `~/ .codecatalyst/workflows/` pasta na raiz do seu [repositório de origem](#). O arquivo pode ter uma extensão `.yaml` ou `.yml`.

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 ter 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 do usuário, use Ctrl+F. O elemento será listado com sua 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 simples de definição de fluxo de trabalho. Inclui algumas propriedades de nível superior, uma Triggers seção e uma Actions seção com duas ações: Build e Test. Para ter 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 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 é novo no YAML, aqui estão algumas diretrizes rápidas para garantir que você esteja fornecendo 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 usar 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, A-Z, 0-9), hífen (-) 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 stringmy 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: Prefácio 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, TriggersType, e. Branches
- Uma seção é qualquer propriedade que tenha subpropriedades. No trecho de código a seguir, há uma Triggers seção.

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

A seguir está a documentação de referência para as propriedades de nível superior no 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 registros. O nome do fluxo de trabalho e o nome do arquivo de definição do fluxo de trabalho podem coincidir, ou você pode nomeá-los de forma diferente. Os nomes dos fluxos de trabalho não precisam ser exclusivos. Os nomes dos fluxos de trabalho são limitados a caracteres alfanuméricos (a-z, A-Z, 0-9), hífens (-) e sublinhados (_). Não são permitidos

espaços. Você não pode usar aspas para ativar caracteres especiais e espaços nos nomes dos fluxos de trabalho.

UI correspondente: editor visual/propriedades do fluxo de trabalho/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.

UI correspondente: nenhuma

RunMode

(Optional)

Como CodeCatalyst lida com várias execuções. Você pode usar um dos seguintes valores:

- QUEUED— Várias execuções são colocadas em fila e executadas uma após a outra. Você pode 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 (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 ter mais informações, consulte [Configurando o comportamento de enfileiramento das execuções](#).

UI correspondente: editor visual/Propriedades do fluxo de trabalho/Avançado/Modo de execução

Compute

(Optional)

O mecanismo de computação usado para executar suas ações de fluxo de trabalho. Você pode especificar a computação no nível do fluxo de trabalho ou no nível da ação, mas não em ambos.

Quando especificada no nível do fluxo de trabalho, a configuração computacional se aplica a todas as ações definidas no fluxo de trabalho. No nível do fluxo de trabalho, você também pode executar várias ações na mesma instância. Para ter mais informações, consulte [Compartilhamento de computação entre ações](#).

Para obter mais informações sobre computação, consulte [Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho](#).

UI correspondente: nenhuma

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
  SharedInstance: true | false
```

Type

(Compute/Type)

(Obrigatório se Compute estiver configurado)

O tipo de mecanismo de computação. Você pode usar um dos seguintes valores:

- EC2 (editor visual) ou EC2 (editor YAML)
Otimizado para flexibilidade durante as corridas 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 visual/Propriedades do fluxo de trabalho/Avançado/Tipo de computação

Fleet

(Compute/Fleet)

(Optional)

Especifique a máquina ou frota que executará seu fluxo de trabalho ou 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 frotas sob demanda: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Para obter mais informações sobre frotas sob demanda, consulte. [Propriedades de frota sob demanda](#)

Com frotas provisionadas, você configura um conjunto de máquinas dedicadas para executar suas ações de fluxo de trabalho. Essas máquinas permanecem ociosas, prontas para processar ações imediatamente. Para obter mais informações sobre frotas provisionadas, consulte. [Propriedades da frota provisionada](#)

Se Fleet for omitido, o padrão será. `Linux.x86-64.Large`

Para obter mais informações sobre frotas de computação, consulte. [Frotas de computação](#)

UI correspondente: editor visual/Propriedades do fluxo de trabalho/Avançado/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 tempo de execução). Você pode usar um dos seguintes valores:

- `TRUE` significa que a imagem do ambiente de tempo de execução é compartilhada entre as ações do fluxo de trabalho.
- `FALSE` significa que uma imagem separada do ambiente de tempo de execução 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 obter mais informações sobre compartilhamento de computação, consulte [Compartilhamento de computação entre ações](#).

UI correspondente: nenhuma

Triggers

(Optional)

Uma sequência de um ou mais gatilhos para esse fluxo de trabalho. Se um gatilho não for especificado, você deverá iniciar manualmente seu fluxo de trabalho.

Para obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).

UI correspondente: editor visual/diagrama de fluxo de trabalho/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)

(Obrigatório se Triggers estiver configurado)

Especifique o tipo de gatilho. Você pode usar um dos seguintes valores:

- Push (editor visual) ou PUSH (editor YAML)

Um gatilho push inicia a execução de um fluxo de trabalho quando uma alteração é enviada ao 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).

- Pull request (editor visual) ou PULLREQUEST (editor YAML)

Um gatilho de pull request inicia a execução de um fluxo de trabalho quando uma pull request é aberta, atualizada ou fechada no seu repositório de origem. A execução do fluxo de trabalho usará os arquivos na ramificação da qual você está extraíndo (ou seja, a ramificação de origem).

- Agenda (editor visual) ou SCHEDULE (editor YAML)

Um gatilho de agendamento inicia a execução do fluxo de trabalho em um cronograma definido 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 `Branches` propriedade (editor YAML).)

Ao configurar um gatilho de agendamento, siga estas diretrizes:

- Use apenas um gatilho de agendamento 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 ter mais informações, consulte [Expression](#).

Para ver exemplos, consulte [Exemplos de gatilhos](#).

UI correspondente: editor visual/diagrama de fluxo de trabalho/acionadores/tipo de gatilho

Events

(Triggers/Events)

(Obrigatório se o gatilho Type estiver definido como PULLREQUEST)

Especifique o tipo de eventos de pull request que iniciarão a execução de um fluxo de trabalho. A seguir estão os valores válidos:

- A solicitação pull é criada (editor visual) ou OPEN (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma pull request é criada.

- A solicitação pull está fechada (editor visual) ou CLOSED (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma pull request é fechada. O comportamento do CLOSED evento é complicado e é melhor compreendido por meio de um exemplo. Consulte [Exemplo: um gatilho com um puxão, galhos e um evento "FECHADO"](#) Para mais informações.

- Uma nova revisão é feita para pull request (editor visual) ou REVISION (editor YAML)

A execução do fluxo de trabalho é iniciada quando uma revisão de uma pull request é criada. A primeira revisão é criada quando a pull request é criada. Depois disso, uma nova revisão é criada toda vez que alguém envia um novo commit para a ramificação de origem especificada na pull request. Se você incluir o REVISION evento em seu gatilho de pull request, poderá omitir o OPEN evento, pois REVISION é um superconjunto de OPEN

Você pode especificar vários eventos no mesmo gatilho de pull request.

Para ver exemplos, consulte [Exemplos de gatilhos](#).

UI correspondente: editor visual/diagrama de fluxo de trabalho/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 acionador:

- Para um gatilho de pressão, 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 pull request, 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 agendamento, especifique as ramificações que contêm os arquivos que você deseja que sua execução programada use. 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:

- O galho para o qual você está empurrando (para gatilhos de pressão). Para ter mais informações, consulte [Exemplo: um simples gatilho de envio de código](#).
- A ramificação da qual você está retirando (para acionadores de pull request). Para ter mais informações, consulte [Exemplo: um simples gatilho de pull request](#).
- Todas as filiais (para acionadores de agendamento). Uma execução de fluxo de trabalho será iniciada por ramificação em seu repositório de origem. Para ter mais informações, consulte [Exemplo: um simples gatilho de agendamento](#).

Para obter mais informações sobre ramificações e gatilhos, consulte [Ação de considerações ao ramificar](#)

Para obter mais exemplos, consulte [Exemplos de gatilhos](#).

UI correspondente: editor visual/diagrama de fluxo de trabalho/acionadores/ramificações

FilesChanged

(Triggers/FilesChanged)

(Opcional se o gatilho Type estiver definido como PUSH, ou PULLREQUEST. Não é suportado 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 combinar nomes ou caminhos de arquivos.

Para ver exemplos, consulte [Exemplos de gatilhos](#).

UI correspondente: editor visual/diagrama de fluxo de trabalho/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 agendadas ocorram.

As expressões Cron CodeCatalyst usam a seguinte sintaxe de seis campos, em que cada campo é separado por um espaço:

minutos horas days-of-month mês days-of-week ano

Exemplos de expressões cron

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano | Significado |
|---------|-------|-------------|-----|----------------|-----|--|
| 0 | 0 | ? | * | SEG-SEX | * | Executa um fluxo de trabalho à meia-noite (UTC +0) de segunda a sexta-feira. |
| 0 | 2 | * | * | ? | * | Executa um fluxo |

| Minutos | Horas | Dias do mês | Mês | Dias da semana | Ano | Significado |
|---------|-------|-------------|-----|----------------|-----|---|
| | | | | | | de trabalho às 2:00 da manhã (UTC+0) todos os dias. |
| 15 | 22 | * | * | ? | * | Executa 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 no dia inicial e 2h no 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 SCHEDULE acionador.
- Coloque a expressão cron entre aspas duplas (") no editor YAML.
- Especifique a hora 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 um 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 obter exemplos de acionadores de agendamento, consulte. [Exemplos de gatilhos](#)

UI correspondente: editor visual/diagrama de fluxo de trabalho/acionadores/agenda

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 `Identifier` propriedade que indica o ID exclusivo e codificado da ação. Por exemplo, `aws/build@v1` identifica a ação de criação.
- uma `Configuration` seção que contém propriedades específicas da ação.

Para obter mais informações sobre cada tipo de ação, consulte [Tipos de ação](#). O [Tipos de ação](#) tópico tem links para a documentação de cada ação.

A seguir está 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)

Substitua *action-name* pelo nome que você deseja dar à ação. Os nomes das ações devem ser exclusivos no fluxo de trabalho e devem incluir somente caracteres alfanuméricos, hífens e sublinhados. Para obter mais informações sobre regras de sintaxe, consulte [Diretrizes de sintaxe YAML](#).

Para obter mais informações sobre práticas de nomenclatura para ações, incluindo restrições, consulte o [action-or-gate-name](#)

UI correspondente: editor visual/nome da ação/guia **de 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 de ações ajuda a manter seu 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 dos grupos de ação devem ser exclusivos no fluxo de trabalho e devem incluir somente caracteres alfanuméricos, hífens e sublinhados. Para obter mais informações sobre regras de sintaxe, consulte [Diretrizes de sintaxe YAML](#).

Para obter mais informações sobre grupos de ação, consulte [Agrupando ações em grupos de ação](#).

UI 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 peça do trabalho é mantida em um registro distinto chamado de edição. Você pode dividir um problema em objetivos menores adicionando uma lista de tarefas a ele. Cada edição pode ter uma descrição, responsável, status e outras propriedades, que você pode pesquisar, agrupar e filtrar. Você pode visualizar seus problemas usando as visualizações padrão ou criar suas próprias visualizações com filtragem, classificação ou agrupamento personalizados. Para obter mais informações sobre conceitos relacionados a problemas, consulte [Conceitos de questões](#). Para saber como criar sua primeira edição, consulte [Criando um problema no CodeCatalyst](#).

Aqui está um possível fluxo de trabalho para uma equipe que usa problemas:

Jorge Souza é desenvolvedor e trabalha em um projeto. Ele e seus colegas do projeto, Li Juan, Mateo Jackson e Wang Xiulan, colaboram para determinar qual trabalho precisa ser feito. Todos os dias, ele e seus colegas desenvolvedores realizam uma reunião de sincronização, liderada por Wang Xiulan. Eles abrem o quadro navegando até a visualização do quadro de uma de suas 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. Sua 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 tarefas a serem concluídas, ele planeja seu 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 seus problemas com um status de rascunho, como lista de pendências, pois ele não planeja começar a resolvê-los imediatamente. Os problemas em um status de rascunho aparecem na visualização Rascunhos, onde devem ser planejados e priorizados. Quando Jorge estiver pronto para começar o trabalho, ele move o problema correspondente para o quadro, atualizando seu status para um status em outra categoria (Não iniciado, iniciado ou concluído). À medida que 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, as edições concluídas se acumulam no status Concluído. Wang Xiulan decide

removê-los da visualização arquivando-os usando o botão de arquivamento rápido, para que os desenvolvedores possam se concentrar nos problemas relacionados ao trabalho atual e futuro.

Ao planejar seu trabalho, os desenvolvedores que trabalham no projeto escolhem Classificar por e Agrupar por para encontrar os problemas que desejam mover da lista de pendências para o quadro. Eles podem optar por adicionar problemas ao quadro com base nas solicitações de maior prioridade dos clientes, então agrupam o quadro por uma etiqueta de 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 a lista de pendências para ajudar a garantir que a prioridade reflita com precisão a importância de cada problema para o sucesso do projeto.

Tópicos

- [Conceitos de questões](#)
- [Rastreamento o trabalho com problemas](#)
- [Organizando o trabalho com listas de pendências, etiquetas e quadros](#)
- [Cotas para edições em CodeCatalyst](#)

Conceitos de questões

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 ajudá-lo a 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 de 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 obter mais informações sobre status e categorias de status, consulte [Status e categorias de status](#).

Você pode visualizar todas as questões ativas em seu projeto a partir da visualização padrão de questões ativas.

Problemas arquivados

Um problema arquivado é um problema que não é mais relevante para o 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 a edição está atribuída. Se a pessoa não aparecer na lista quando você pesquisar por ela, ela não foi adicionada ao seu projeto. Para adicioná-los, consulte [Convidar um usuário para um projeto](#). Para habilitar vários responsáveis por um problema, consulte [Ativando ou desativando vários destinatários](#). Problemas com vários responsáveis aparecerão em seu 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 ágil, 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 obter mais informações sobre os tipos de estimativa e como configurá-los, consulte [Configurando a 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 seu projeto. Se você estiver usando o desenvolvimento ágil, um problema também pode descrever uma história épica ou de um usuário.

Rótulo

O rótulo é usado para agrupar, classificar e filtrar problemas. Você pode inserir um novo nome de etiqueta ou escolher uma das etiquetas 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 [Categorizando o 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 de status. As categorias de status são usadas para ajudar a organizar seus status e preencher as visualizações padrão do problema.

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 suas categorias de status entre parênteses: Lista de pendências (rascunho), A fazer (não iniciado), Em andamento (Iniciado), Em revisão (Iniciado) e Concluído (Concluído).

Para obter mais informações sobre como trabalhar com status, consulte [Rastreamento o 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 suas tarefas como concluídas.

Visões

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 de status Rascunho aparece nessa exibição. Essa visualização pode ser usada pelas equipes para ver quais problemas ainda estão sendo definidos ou aguardando para serem atribuídos e resolvidos.
- A visualização de problemas ativos é uma visão geral de todos os problemas que estão sendo resolvidos no momento. Qualquer problema com um status nas categorias de status Não iniciado, Iniciado ou Concluído aparecerá nessa exibição.
- A visualização Todas as questões é uma visualização em grade que mostra todas as questões do projeto, tanto rascunhos quanto questões ativas.
- A visualização Arquivado mostra todas as edições arquivadas.

Rastreando o trabalho com problemas

Você pode planejar e acompanhar seu trabalho em um projeto usando problemas. Cada edição 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 ajudá-lo a acompanhar o trabalho relacionado, adicionar rótulos para ajudá-lo a organizar e categorizar 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 pode estimar o trabalho, atribuí-lo aos usuários e adicionar comentários para ajudar outras pessoas a entender o trabalho e seu progresso. Você também pode exportar edições para ajudar a trazer as informações que elas contêm para outros formatos.

Criando um problema no CodeCatalyst

As equipes de desenvolvimento criam problemas para ajudar a monitorar e gerenciar seu trabalho. Você pode criar problemas em um projeto com base nas suas necessidades. Por exemplo, você pode criar um problema para rastrear a atualização de uma variável no seu código. Você pode atribuir problemas a outros usuários no projeto, usar rótulos para ajudá-lo a 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 em que você deseja criar um problema.
3. Na página inicial do projeto, escolha Criar problema. Como alternativa, no painel de navegação, escolha Problemas.
4. Escolha Criar problema.

Note

Você também pode adicionar problemas em linha ao usar uma visualização em grade.

5. Insira um título para a edição.
6. (Opcional) Insira uma descrição. Você pode usar o 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, escolha + 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 escolhendo a caixa de seleção ou reordená-las escolhendo e arrastando a tarefa do lado esquerdo da caixa de seleção.
9. (Opcional) Adicione uma etiqueta existente ou crie uma nova etiqueta e adicione-a escolhendo + Adicionar etiqueta.
 - a. Para adicionar uma etiqueta existente, escolha a etiqueta 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 novo 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 escolhendo + Adicionar um destinatário. Você pode se adicionar rapidamente como destinatário escolhendo + Me adicionar.

Tip

Você pode escolher atribuir um problema ao Amazon Q para que o Amazon Q tente resolver o problema. Para ter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#). Esse recurso está disponível somente na região Oeste dos EUA (Oregon).

Essa funcionalidade exige que recursos generativos de IA estejam habilitados para o espaço. Para obter mais informações, consulte [Gerenciamento de recursos generativos de IA](#).

11. (Opcional) Adicione um campo personalizado existente ou crie um novo 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 novo campo personalizado e adicioná-lo, digite o nome do campo personalizado que você deseja criar no campo de pesquisa e pressione enter. Em seguida, escolha o tipo de campo personalizado que você deseja criar e defina um valor.
12. Escolha Criar problema. Uma notificação aparece no canto inferior direito: se o problema foi criado com sucesso, uma mensagem de confirmação será exibida informando que o problema foi criado com sucesso. Se o problema não tiver sido criado com êxito, uma mensagem de erro com o motivo da falha será exibida. Em seguida, você pode escolher Tentar novamente para editar e tentar criar o problema novamente ou escolher Descartar para descartar o problema. Ambas as opções ignorarão a notificação.

Note

Você não pode vincular uma pull request a um problema ao criá-la. No entanto, você pode [editá-lo](#) depois de criá-lo para adicionar links às pull requests.

Melhores práticas 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, é 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. Frequentemente, 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 à Amazon Q, ela 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

Desenvolvido pelo Amazon Bedrock: AWS implementa a detecção [automática de abusos](#). Como o recurso Atribuir problemas ao Amazon Q com o Amazon Q Developer Agent para desenvolvimento de software é criado no Amazon Bedrock, os usuários podem aproveitar ao máximo os controles implementados no Amazon Bedrock para reforçar a segurança e o uso responsável da inteligência artificial (IA).


O Amazon Q tem melhor desempenho em questões simples e problemas simples. Para obter melhores resultados, use uma linguagem simples para explicar claramente o que você quer que seja feito. A seguir estão algumas das melhores práticas para ajudar você a criar problemas otimizados para o Amazon Q trabalhar.

Important

Os recursos generativos de IA estão disponíveis somente na região Oeste dos EUA (Oregon).

- Mantenha as coisas 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 excessivamente 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 das APIs que você deseja alterar, os métodos que deseja atualizar, os testes que precisam de alterações e quaisquer outros detalhes que você possa imaginar.

- Certifique-se de ter todos os detalhes incluídos no título e na descrição do problema antes de atribuí-lo ao Amazon Q. Você não pode alterar o título ou a descrição de uma edição depois de atribuí-la à Amazon Q, portanto, certifique-se de ter todas as informações necessárias em uma edição antes de atribuí-la à 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 para repositórios vinculados. Certifique-se de que o problema exija apenas alterações em um único repositório de origem antes de atribuí-lo ao Amazon Q.
- Use o padrão sugerido pela 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 pull request criado por ele. Isso proporciona uma experiência mais interativa com o Amazon Q, que ajuda você a ajustar sua abordagem e refinar o código que ele cria para resolver o problema.

 Note

O Amazon Q não responde a comentários individuais sobre problemas ou pull requests, mas os revisará quando solicitado a reconsiderar sua abordagem ou criar uma revisão.

- Sempre analise cuidadosamente a abordagem sugerida pela Amazon Q. Depois de aprovar sua abordagem, a Amazon Q começará a trabalhar na geração de código com base nessa abordagem. Certifique-se de que a abordagem pareça correta e inclua todos os detalhes que você espera antes de pedir à Amazon Q que continue.
- Certifique-se de permitir 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 pull request. Nesse caso, qualquer pull request criada pela Amazon Q que inclua a criação ou atualização do fluxo de trabalho YAML pode iniciar uma execução desses fluxos de trabalho incluídos na pull request. Como prática recomendada, não escolha permitir 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 pull request criada.

Para obter mais informações, consulte [Tutorial: Usando recursos CodeCatalyst generativos de IA para acelerar seu trabalho de desenvolvimento](#) [Gerenciando recursos generativos de IA](#).

Estimando um problema

No desenvolvimento ágil, 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 seus problemas, você deve primeiro escolher o tipo de estimativa que deseja usar para seu projeto. Por padrão, há dois tipos para escolher. Para usar o tamanho da camiseta ou o sequenciamento de Fibonacci de forma eficaz, sua equipe deve se alinhar ao que cada tamanho representa. Decida juntos o que cada estimativa representa para você e, em seguida, 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, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha 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 estimativas ocultas. Se a configuração de estimativa do projeto estiver definida como Ocultar estimativas, não haverá um campo Estimativa nas edições 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 de conversão é mostrado na tabela a seguir.

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| XS | 1 |
| XS | 2 |
| S | 3 |
| M | 5 |

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| L | 8 |
| XL | 13 |

Para adicionar ou alterar uma estimativa para um problema, você pode [editar o problema](#).

Editando e colaborando em questões em CodeCatalyst

Sumário

- [Editando um problema](#)
- [Trabalhar com anexos](#)
 - [Visualizando e gerenciando anexos](#)
- [Gerenciando tarefas sobre problemas](#)
- [Marcar um problema como bloqueado ou desbloqueado](#)
- [Adicionar, editar ou excluir comentários](#)
 - [Usando menções em um comentário](#)

Editando 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 uma edição

1. Escolha o problema que você deseja editar para ver os detalhes do problema. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Para editar o título da edição, escolha o título, insira um novo título e pressione enter.
3. Para editar a descrição, escolha a descrição, insira uma nova descrição e pressione enter. Você pode usar o Markdown para adicionar formatação.
4. Em Tarefas, você pode visualizar e gerenciar as tarefas do problema. Para ter mais informações, consulte [Gerenciando tarefas sobre problemas](#).
5. Para editar o Status, a Estimativa ou a Prioridade, escolha uma opção nos respectivos menus suspensos.

6. Em Rótulos, você pode adicionar um rótulo existente, criar um novo rótulo ou remover um rótulo.
 - a. Para adicionar uma etiqueta existente, escolha + Adicionar etiqueta e escolha a etiqueta 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 novo rótulo e adicioná-lo, escolha + Adicionar rótulo, digite o nome do rótulo que você deseja criar no campo de pesquisa e pressione enter.
 - c. Para remover uma etiqueta, escolha o ícone X ao lado da etiqueta que você deseja remover. Se você remover uma etiqueta de todas as edições, a etiqueta aparecerá na seção Etiquetas não usadas na seção Etiquetas das configurações da edição. 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 uma ocorrência, escolha + Adicionar um destinatário na seção Destinatário, depois pesquise e escolha o destinatário na lista. Você pode escolher + Me adicionar para se adicionar rapidamente como destinatário.
8. Em Anexos, você pode adicionar, baixar ou remover anexos. Para ter mais informações, consulte [Trabalhar com anexos](#).
9. Para vincular uma pull request, escolha Vincular pull request e, em seguida, escolha uma pull request na lista ou insira sua URL ou ID. Para desvincular uma pull request, escolha o ícone de desvinculação.

 Tip

Depois de adicionar um link a uma pull request para um problema, você pode navegar rapidamente até ele escolhendo seu ID na lista de pull requests vinculados. Você pode usar a URL de uma pull request para vincular pull requests que estão em projetos diferentes do quadro de problemas, mas somente usuários que são membros desse projeto poderão visualizar ou navegar até essa pull request.

10. (Opcional) Adicione e defina um campo personalizado existente, crie um novo campo personalizado 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 novo campo personalizado e adicioná-lo, digite o nome do campo personalizado que você deseja criar no campo de pesquisa e pressione enter. Em seguida, 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. Use o procedimento a seguir para gerenciar anexos de um problema.

O tamanho dos anexos adicionados às edições conta para as cotas de armazenamento do seu espaço. Para obter informações sobre como visualizar e gerenciar anexos do seu projeto, consulte.

[Visualizando e gerenciando 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. Certifique-se de que os usuários estejam cientes das melhores práticas 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 você deseja gerenciar anexos. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Para adicionar um anexo, escolha Carregar arquivo. Navegue até o arquivo no explorador de arquivos do seu sistema operacional e selecione-o. Escolha Abrir para adicioná-lo como anexo. Para obter informações sobre cotas, como tamanho máximo de anexo, consulte [Cotas para edições em CodeCatalyst](#).

Observe as seguintes restrições aos nomes dos arquivos anexos e aos tipos de conteúdo:

- Os seguintes caracteres não são permitidos nos nomes dos arquivos:

- Personagens 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 o seguinte 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 elipses 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 escolha Copiar URL.
5. Para remover um anexo, escolha o menu de elipses ao lado do anexo que você deseja remover e escolha Excluir.

Visualizando e gerenciando anexos

Você pode ver uma tabela com cada anexo adicionado aos problemas do seu 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, seu 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. Certifique-se de que os usuários estejam cientes das melhores práticas

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, escolha Problemas.
2. Escolha o ícone de reticências e escolha Configurações.
3. Escolha a guia Anexos.

Gerenciando tarefas sobre problemas

As tarefas podem ser adicionadas aos problemas para detalhar, organizar e acompanhar ainda mais o trabalho desse problema.

Para gerenciar tarefas em um problema

1. Escolha o problema para o qual você deseja gerenciar tarefas. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Em Tarefas, você pode visualizar e gerenciar tarefas para o problema.
 1. Para adicionar uma tarefa, insira o nome da tarefa no campo de texto e pressione enter.
 2. Para marcar uma tarefa como concluída, escolha a caixa de seleção da tarefa.
 3. Para visualizar ou atualizar os detalhes de uma tarefa, escolha-a na lista.
 4. Para reordenar as tarefas, escolha e arraste a tarefa do lado esquerdo da caixa de seleção.
 5. Para remover uma tarefa, escolha o menu de elipses da tarefa e escolha Remover.

Marcar um problema como bloqueado ou desbloqueado

Se algo estiver impedindo você de resolver um problema, convém marcá-lo como bloqueado. Por exemplo, seu problema pode ser bloqueado se depender de uma alteração em outra parte da sua 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 obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Escolha Ações e, em seguida, escolha Marcar como bloqueado.

Para desbloquear um problema

1. Abra o problema que você deseja desbloquear. Para obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Escolha Ações e, em seguida, escolha Marcar como desbloqueado.

Adicionar, editar ou excluir 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, escolha Problemas.
3. Escolha o problema ao qual você deseja adicionar o comentário. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
4. Insira o comentário no campo Comentários. Você pode usar o Markdown para adicionar formatação.
5. Selecione Enviar.

Para editar um comentário

Você pode editar os comentários que você faz sobre problemas. Você só pode editar comentários de sua autoria.

1. Navegue até o projeto.
2. Na barra de navegação, escolha Problemas.

3. Escolha o problema em que você deseja editar um comentário. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
4. Para editar um comentário, encontre o comentário que você deseja editar.

 Tip

Você pode classificar os comentários primeiro pelos mais antigos ou pelos mais novos. Os comentários são carregados 10 por vez.

5. Escolha o ícone de reticências e, em seguida, escolha Editar.
6. Edite o comentário. Você pode usar o Markdown para adicionar formatação.
7. Selecione Save (Salvar). O comentário agora está atualizado.

Para excluir um comentário

Você pode excluir os comentários que você faz sobre problemas. Você só pode excluir comentários 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, escolha Problemas.
3. Escolha o problema em que você deseja excluir um comentário. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
4. Escolha o ícone de reticências, escolha Excluir e, em seguida, escolha Confirmar.


Usando 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 @mention em um comentário

1. Navegue até o projeto.
2. Na barra de navegação, escolha Problemas.
3. Escolha o problema que você deseja editar para ver os detalhes do problema. Para obter ajuda sobre como encontrar seu problema, consulte [Encontrando e visualizando problemas](#).

4. Escolha 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 aos seus `@mention` será preenchida à medida que você digita.

9. Escolha o item desejado que você gostaria de mencionar. O caminho que mostra onde o item está localizado será preenchido na caixa de texto do comentário.
10. Conclua seu comentário e escolha Enviar.

Encontrando e visualizando problemas

As seções a seguir descrevem como pesquisar e visualizar problemas em um CodeCatalyst projeto com eficiência.

Procurando por um problema

Você pode encontrar um problema pesquisando parâmetros específicos. Para obter mais informações sobre como refinar sua pesquisa, consulte [Pesquise código, problemas, projetos e usuários em CodeCatalyst](#).

Para pesquisar um problema

1. Navegue até o projeto.
2. Use a barra de pesquisa para pesquisar problemas ou informações relacionadas a problemas. Você pode usar parâmetros de consulta para refinar sua pesquisa. Para ter mais informações, consulte [Pesquise código, problemas, projetos e usuários em CodeCatalyst](#).

Problemas de classificação

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 na ordem Manual para alterar sua ordem. Essa opção de classificação é útil para organizar o acúmulo 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 do quadro |
|--|---|
| Pedido manual | Pedido manual |
| Última atualização | Última atualização |
| Prioridade | Prioridade |
| Estimativa | Estimativa |
| Cargo | Cargo |
| 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, escolha Problemas. A visualização padrão é o Quadro.
3. (Opcional) Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegar 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 para classificar. Os problemas serão classificados em ordem crescente.
5. Para classificar uma visualização do quadro, escolha o menu suspenso Classificar por e escolha um parâmetro para classificar. Os problemas serão classificados em ordem crescente.

Problemas de agrupamento

O agrupamento é usado para organizar os problemas no quadro por meio de vários parâmetros, como destinatário, rótulos e prioridade.

Para agrupar questões

1. Navegue até o projeto.
2. No painel de navegação, escolha Problemas. A visualização padrão é o Quadro.
3. (Opcional) Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegar 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, escolha a ordem do Grupo.
 - Se você escolher Rótulo, escolha os rótulos e, em seguida, escolha Ordem do grupo.
6. (Opcional) Escolha a opção Mostrar grupos vazios para mostrar ou ocultar grupos que não têm problemas atualmente atribuídos a eles.
7. A exibição é atualizada à medida que você faz suas escolhas. Um problema aparece somente no grupo que corresponde aos parâmetros configurados.

Problemas de filtragem

Use a filtragem para encontrar problemas que contenham um nome, prioridade, rótulo, campos personalizados ou destinatário especificados.

Para filtrar problemas

1. Navegue até o projeto.
2. No painel de navegação, escolha Problemas.
3. (Opcional) Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegar 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. Escolha Filtro e, em seguida, escolha + Adicionar filtro.
5. Escolha os parâmetros para filtrar. Você pode 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 exibição será atualizada para mostrar os problemas que correspondem ao filtro.

Progredindo em um problema

Cada problema tem um ciclo de vida. Em CodeCatalyst, os problemas geralmente começam como um rascunho na lista de pendências. Quando o trabalho para resolver esse problema deve ser iniciado, ele é movido para outra categoria de status e passa por vários status até ser concluído e, em seguida, é arquivado. Você pode mover ou progredir um problema em seu ciclo de vida das seguintes maneiras:

- Você pode mover um problema entre a lista de pendências e o quadro.
- Você pode mover um problema em andamento por vários estágios de conclusão.
- Você pode arquivar um problema que foi concluído.

Transferindo um problema entre a lista de pendências e o quadro

Você pode mover um problema da lista de pendências para o quadro assim que começar a trabalhar nele. Você também pode mover um problema de volta para a lista de pendências se o trabalho for adiado.

Para mover um problema entre a lista de pendências e o quadro

1. Navegue até o projeto.
2. No painel de navegação, escolha Problemas. A visualização padrão é o Quadro.
3. Para mover um problema do quadro para a lista de pendências:
 - a. Escolha o problema que você deseja mover. Para obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
 - b. Escolha Backlog no menu suspenso Status.
4. Para mover um problema da lista de pendências para o quadro:
 - a. Para navegar até a lista de pendências, escolha Quadro e escolha Lista de pendências.
 - b. Escolha o problema que você deseja mover. Para obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
 - c. Escolha Adicionar ao quadro ou escolha um status diferente de Lista de pendências.

Progrida um problema nos estágios do ciclo de vida no quadro

Você pode mover um problema dentro de um quadro por diferentes status até ser concluído.

Para mover um problema para o quadro

1. No painel de navegação, escolha Problemas. A visualização padrão é o Quadro.
2. Execute um destes procedimentos:
 - Arraste e solte um problema em outro status.
 - Escolha um problema e, em seguida, escolha um status no menu suspenso Status.
 - Escolha um problema e, em seguida, escolha Mover para: **próximo status**.

Para obter informações sobre como arquivar um problema, consulte [Arquivando um problema](#).

Transferindo 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-alvo.

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 uma edição do grupo Wang Xiulan para o grupo Saanvi Sarkar atualizará o destinatário da edição para Saanvi Sarkar.

Arquivando 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 visualizados na visualização padrão dos problemas arquivados, na qual podem ser desarquivados, se necessário.

Você arquiva um problema se:

- Você concluiu o problema e não precisa mais dele na coluna Concluído.
- Você não tem planos de trabalhar nisso.
- Você o criou por engano.
- Você atingiu o número máximo de problemas ativos.

Para arquivar um problema

1. Abra o problema que você deseja arquivar. Para obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Escolha Ações e, em seguida, escolha Mover para arquivamento.
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 escolha 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 obter ajuda para encontrar seu problema, consulte [Encontrando e visualizando problemas](#).
2. Escolha Desarquivar.

Problemas de exportação

Você pode exportar problemas em sua exibição atual para um arquivo.xlsx. Para exportar problemas, execute as etapas a seguir.

Para exportar edições

1. Navegue até o projeto.
2. Na barra de navegação, escolha Problemas.
3. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e navegue até a exibição que contém os problemas que você deseja exportar. Somente as edições mostradas na exibição serão exportadas.
4. Escolha o menu de elipses e escolha Exportar para o Excel.
5. O arquivo.xlsx é baixado. Por padrão, ela é intitulada como o nome do projeto e a data em que a exportação foi concluída.

Organizando o trabalho com listas de pendências, etiquetas e quadros

Nem todas as equipes trabalham da mesma forma. 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 os problemas para que todos os usuários usem o mesmo método de estimativa. 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.

Categorizando o trabalho com rótulos

Você pode personalizar rótulos para problemas. Isso inclui editar o rótulo e alterar a cor. As etiquetas podem ajudar você a categorizar 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 uma etiqueta](#)
- [Editando um rótulo](#)
- [Excluindo um rótulo](#)

Criação de uma etiqueta

Em CodeCatalyst, você cria rótulos adicionando-os ao criar um novo problema ou ao editar um problema existente. Para ter mais informações, consulte [Criando um problema no CodeCatalyst](#) e [Editando e colaborando em questões em CodeCatalyst](#).

Editando um rótulo

Use o procedimento a seguir para alterar o nome ou a cor de uma etiqueta existente.

Para editar um rótulo

1. No painel de navegação, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha Configurações.
3. No quadro Rótulos, há uma lista dos rótulos usados no projeto. Escolha o ícone de edição ao lado da etiqueta que você deseja editar. Faça uma ou mais das coisas a seguir:
 - a. Edite o nome da etiqueta.
 - b. Para alterar a cor, escolha a roda de cores. Use o seletor para escolher uma nova cor.
4. Para salvar as alterações feitas na etiqueta, escolha o ícone de marca de seleção.
5. A etiqueta alterada agora está visível na sua lista de etiquetas disponíveis. Você também pode ver quantos problemas estão usando esse rótulo.

Note

Você pode escolher o número exibido ao lado de cada etiqueta para navegar até a página Todas as edições e ver todas as edições que contêm essa etiqueta.

Excluindo um rótulo

No momento, você não pode excluir um rótulo de problemas em CodeCatalyst. Se você remover uma etiqueta de todas as edições, a etiqueta aparecerá na seção Etiquetas não usadas na seção Etiquetas das configurações da edição. 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.

Organizando o trabalho com campos personalizados

Você pode criar campos personalizados para ajudar a organizar e visualizar o trabalho do seu 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 nomes e valores. Você filtra pelo nome do campo personalizado e, em seguida, 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](#).

Excluindo um campo personalizado

Para excluir um campo personalizado, você deve remover o campo personalizado de cada edição à qual ele é adicionado. Quando um campo personalizado for excluído, você não verá mais o campo personalizado no Filtro. Você pode usar filtros para visualizar todos os problemas com um campo personalizado e removê-los editando os problemas. Para obter mais informações, consulte [Encontrando e visualizando problemas](#) e [Editando um problema](#)

Rastreando o trabalho com status personalizados

Você pode adicionar status personalizados ao 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 obter mais informações sobre status e categorias de status, consulte [Status e categorias de status](#) e para obter mais informações sobre exibições, consulte [Encontrando e visualizando problemas](#).

Para criar um status

1. No painel de navegação, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha Configurações.
3. Em Status, escolha o ícone de adição ao lado da categoria em que você deseja que o status esteja.
4. Nomeie o status e escolha 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, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha Configurações.
3. Em Status, escolha o ícone de edição ao lado do status que você deseja editar ou alterar.
4. Edite o status e escolha o ícone de marca de seleção.

O status editado agora está visível no seu quadro.

Para mover um status

1. No painel de navegação, escolha Problemas.
2. Escolha o ícone de reticências e escolha 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 em seu quadro.

Para desativar um status

1. No painel de navegação, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha Configurações.
3. Em Status, escolha um status que você deseja desativar.
4. No status que você deseja desativar, escolha a opção de alternar o status. O status agora está acinzentado.

Note

O status de desativado aparece no quadro até que todas as edições sejam retiradas 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 acinzentado.

Note

Deve haver pelo menos um status ativo em cada categoria. Se houver apenas um status na categoria, você não poderá desativá-lo.

Configurando a 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, escolha Problemas.
2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha 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 estimativas ocultas. Quando o tipo de estimativa for atualizado, nenhum dado será perdido e o valor estimado de todos os problemas será convertido automaticamente. O mapeamento de conversão é mostrado na tabela a seguir.

| Tamanho da camiseta | Sequência de Fibonacci |
|---------------------|------------------------|
| XS | 1 |
| XS | 2 |
| S | 3 |
| M | 5 |
| L | 8 |
| XL | 13 |

Ativando ou desativando vários destinatários

Siga estas etapas para definir a configuração de vários destinatários para problemas em CodeCatalyst

Para ativar ou desativar vários destinatários

1. No painel de navegação, escolha Problemas.

2. Escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolha Configurações.
3. Em Destinatário, na seção Configurações básicas, alterne o indicador para permitir que vários responsáveis sejam atribuídos ao mesmo problema. Uma edição pode ter até 10 responsáveis. Se você não ativar essa opção, só poderá atribuir um destinatário a uma ocorrência.

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 você filtrou, agrupou ou classificou anteriormente.

Para criar uma visualização de problemas

1. No painel de navegação, escolha Problemas.
2. (Opcional) Dependendo do seu caso de uso, talvez você queira criar uma visualização a partir de uma visualização existente. Para navegar até uma visualização diferente, escolha Problemas ativos para abrir o menu suspenso do alternador de visualização de problemas e escolher a visualização.
3. (Opcional) Configure filtros, agrupamento e classificação antes de criar sua exibição. Você pode adicioná-los ao criar uma exibição, mas se fizer isso antes, poderá visualizar o que é mostrado na exibição 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, escolha o + na coluna do quadro. Para criar uma visualização em grade em que os problemas sejam exibidos em uma lista, escolha o + na coluna Grade. Você pode alterar o tipo de exibição antes de criá-la, caso mude de ideia.
5. Na caixa de diálogo Criar exibição, insira um Nome para a exibição.
6. Os campos Filtrar, Agrupar problemas por e Classificar problemas por são preenchidos com base nas configurações da exibição atual. Atualize-os, se necessário.
7. Escolha Criar visualização para criar a visualização e ser transferida para ela.

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 CodeCatalyst](#)

| Recurso | Cota padrão |
|--|--|
| Problemas ativos | Máximo de 1.000 por projeto. |
| Tamanho do anexo | Máximo de 500 MB por anexo.

O armazenamento total máximo de anexos é afetado pelos limites gerais de armazenamento do seu espaço. Para obter mais informações, consulte Preços do . |
| Número total de problemas (ativos e arquivados) | Máximo de 100.000 por projeto. |
| Visualizações salvas | Máximo de 50 visualizações de edição salvas por projeto. |
| Número de pull requests que você pode vincular a um problema | Máximo de 50 pull requests por edição. |
| Status (por projeto) | Máximo de 50 por projeto. |
| Status (por edição) | Máximo de 50 por edição. |
| Etiquetas (por projeto) | Máximo de 200 por projeto. |
| Etiquetas (por edição) | Máximo de 50 por edição. |
| Campos personalizados (por edição) | Máximo de 50 por edição. |
| Destinatários | Máximo de 10 por edição. |
| Comentários | Máximo de 1.000 por edição. |
| Tarefas | Máximo de 100 por edição. |

Configure identidade, permissões e acesso no CodeCatalyst

Ao entrar na Amazon CodeCatalyst pela primeira vez, você cria uma ID de AWS construtor. As IDs do construtor não existem em AWS Identity and Access Management. O nome de usuário que você escolhe durante seu primeiro login se torna seu 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. As funções do projeto, como administrador do projeto e colaborador, são específicas de um projeto, então você pode ter uma função em um projeto e uma função 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 a função 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 da função de Colaborador para realizar suas tarefas. Para ter mais informações, consulte [Concedendo acesso com funções 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

- [Concedendo acesso com funções de usuário](#)
- [Conceda 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](#)

Concedendo acesso com funções de usuário

Na Amazon CodeCatalyst, você pode atribuir funções aos usuários tanto no nível do projeto quanto no nível do espaço. Em um projeto, uma função 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 seus trabalhos.

Você pode atribuir funções aos usuários no espaço. Você também pode atribuir funções aos usuários nos projetos em que eles são membros. Cada usuário só pode ter uma função em um projeto ou espaço, mas os usuários podem ter funções diferentes em cada projeto e espaço. Por exemplo, um usuário pode ter a função de administrador do projeto em um projeto e a função de colaborador em outro projeto.

Tópicos

- [Entendendo as funções do usuário em espaços e projetos](#)
- [Visualizando as permissões disponíveis para cada função](#)
- [Visualizando e alterando funções de usuário](#)

Entendendo as funções do usuário em espaços e projetos

Há três funções disponíveis para um espaço:

- Administrador de espaço
- Usuário avançado
- Acesso limitado

Os usuários que aceitam um convite para um projeto têm a função de acesso limitado automaticamente atribuída a eles no espaço que contém o projeto.

Há quatro funções disponíveis para membros em um projeto:

- Administrador do projeto
- Contributor (Colaborador)
- Avaliador
- Somente para 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.

Função de administrador de 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 a função de administrador de espaço são os únicos usuários que podem adicionar ou remover outros usuários da função de administrador de 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 essa função a pelo menos um outro usuário que possa atuar nessa função caso o criador do espaço original não esteja disponível.

Função 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 ser capazes de criar projetos em um espaço e ajudar a gerenciar os usuários e recursos do espaço. Atribua a função de usuário avançado a usuários que são líderes de equipe ou gerentes que precisam da capacidade de criar projetos e gerenciar usuários no espaço como parte de seu trabalho.

Função de acesso limitado

A função de acesso limitado é a função que a maioria dos usuários terá nos CodeCatalyst espaços. É a função atribuída automaticamente aos usuários quando eles aceitam um convite para um

projeto em um espaço. Ele fornece as permissões limitadas de que eles precisam para trabalhar no espaço que contém esse projeto. Atribua a função de acesso limitado aos usuários que você convida diretamente para o espaço, a menos que o trabalho deles exija que eles gerenciem algum aspecto do espaço.

Função de administrador do projeto

A função de administrador do projeto é a função mais poderosa em um CodeCatalyst projeto. Atribua essa função 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.

As funções do projeto não têm nenhuma permissão no nível do espaço. Portanto, os usuários com a função de administrador do projeto não podem criar projetos adicionais. Somente usuários com a função de administrador do espaço ou usuário avançado podem criar projetos.

Note

A função de administrador do Space tem todas as permissões em CodeCatalyst.

Função de colaborador

A função de Colaborador é destinada à maioria dos membros de um CodeCatalyst projeto. Atribua essa função aos usuários que precisam ser capazes de trabalhar com código, fluxos de trabalho, problemas e ações em um projeto.

Função de revisor

A função de revisor é destinada a usuários que precisam 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 a função de revisor aos usuários que precisam ser capazes de aprovar e comentar pull requests, criar, atualizar, resolver e comentar problemas, além de visualizar o código e os fluxos de trabalho em um projeto.

Função somente para leitura

A função Somente leitura é destinada 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 a função Somente leitura aos usuários que precisam visualizar os recursos e o estado do projeto, mas não interagir diretamente com ele.















Visualizando as permissões disponíveis para cada função

























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](#)
- [Custom blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|-----------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|-----------------------------|
|-----------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|-----------------------------|

Permissões de espaço


















































| | | | | | | | |
|--------------------|---|---|---|---|---|---|---|
| Crie espaço |  |  |  |  |  |  |  |
| Editar detalhes de |  |  |  |  |  |  |  |


















































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| cobrança de espaço | | | | | | | |
| Configurar e ativar o login único |  |  |  |  |  |  |  |
| Remover o login único |  |  |  |  |  |  |  |
| Habilite recursos generativos de IA para um espaço |  |  |  |  |  |  |  |
| Desative recursos generativos de IA para um espaço |  |  |  |  |  |  |  |
| Excluir espaço |  |  |  |  |  |  |  |




































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Adicionar outros usuários à função de administrador do Space |  |  |  |  |  |  |  |
| Remover outros usuários da função de administrador do Space |  |  |  |  |  |  |  |
| Criar equipe |  |  |  |  |  |  |  |
| Excluir equipe |  |  |  |  |  |  |  |
| Equipe de atualização |  |  |  |  |  |  |  |























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Desative os recursos da máquina para o espaço |  |  |  |  |  |  |  |
| Habilite os recursos da máquina para o espaço |  |  |  |  |  |  |  |
| Criar projeto |  |  |  |  |  |  |  |
| Associe uma conexão de conta da AWS ao espaço |  |  |  |  |  |  |  |











































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Atualizar uma conexão de conta da AWS |  |  |  |  |  |  |  |
| Desassociar uma conexão de conta da AWS do espaço |  |  |  |  |  |  |  |
| Exclua uma conexão de conta da AWS e remova-a do espaço |  |  |  |  |  |  |  |
| Convide outras pessoas para o espaço |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Criar conexão VPC |  |  |  |  |  |  |  |
| Editar conexão VPC |  |  |  |  |  |  |  |
| Excluir conexão VPC |  |  |  |  |  |  |  |
| Visualize registros de atividades no espaço |  |  |  |  |  |  |  |
| Veja as conexões da conta da AWS |  |  |  |  |  |  |  |
| Exibir incidentes para CodeCatalyst |  |  |  |  |  |  |  |
| Exibir espaço |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|------------------------------|---|---|---|---|---|---|---|
| Exibir equipes |  |  |  |  |  |  |  |
| Exibir conexões VPC |  |  |  |  |  |  |  |
| Permissões de extensões | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Instalar extensões |  |  |  |  |  |  |  |
| Atualizar extensões |  |  |  |  |  |  |  |
| Excluir extensões |  |  |  |  |  |  |  |
| Conectar uma conta GitHub |  |  |  |  |  |  |  |
| Desconectar uma conta GitHub |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Connect a um site do Jira |  |  |  |  |  |  |  |
| Desconectar um site do Jira |  |  |  |  |  |  |  |
| Exibir detalhes de configuração das extensões instaladas |  |  |  |  |  |  |  |
| Exibir extensões |  |  |  |  |  |  |  |
| Permissões do projeto | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Editar configurações do projeto |  |  |  |  |  |  |  |


















































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Desative os recursos da máquina para o projeto |  |  |  |  |  |  |  |
| Habilite os recursos da máquina para o projeto |  |  |  |  |  |  |  |
| Excluir projeto |  |  |  |  |  |  |  |
| Convidar usuários para um projeto |  |  |  |  |  |  |  |
| Alterar as funções dos usuários em um projeto |  |  |  |  |  |  |  |


















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---------------------------------------|---|---|---|---|---|---|---|
| Remover usuários de um projeto |  |  |  |  |  |  |  |
| Adicionar equipe a um projeto |  |  |  |  |  |  |  |
| Remover equipe de um projeto |  |  |  |  |  |  |  |
| Alterar a função da equipe no projeto |  |  |  |  |  |  |  |
| Exibir projeto |  |  |  |  |  |  |  |
| Exibir a atividade do projeto |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|-------------------------------------|---|---|---|---|---|---|---|
| Exibir equipes no projeto |  |  |  |  |  |  |  |
| Exibir plantas |  |  |  |  |  |  |  |
| Permissões do repositório de origem | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Crie repositórios |  |  |  |  |  |  |  |
| Vincular repositórios |  |  |  |  |  |  |  |
| Desvincular repositórios |  |  |  |  |  |  |  |
| Excluir repositórios |  |  |  |  |  |  |  |





















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|-------------------------------------|---|---|---|---|---|---|---|
| Editar configurações do repositório |  |  |  |  |  |  |  |
| Exibir repositórios |  |  |  |  |  |  |  |
| Exibir configurações do repositório |  |  |  |  |  |  |  |
| Clonar repositórios |  |  |  |  |  |  |  |
| Crie filiais |  |  |  |  |  |  |  |
| Crie regras de ramificação |  |  |  |  |  |  |  |
| Alterar ramificação padrão |  |  |  |  |  |  |  |






















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---------------------------------|-----------------------------------|----------------------------|---------------------------|------------------------------------|-----------------------|-------------------|-----------------------------|
| Excluir filiais | | | | | | | |
| Mesclar ramificações | | | | | | | |
| Atualizar regras de ramificação | | | | | | | |
| Exibir filiais | | | | | | | |
| Exibir regras de filiais | | | | | | | |
| Crie pastas | | | | | | | |
| Excluir pastas | | | | | | | |
| Editar pastas | | | | | | | |
| Exibir pastas | | | | | | | |
| Crie arquivos | | | | | | | |






















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Excluir arquivos |  |  |  |  |  |  |  |
| Editar arquivos |  |  |  |  |  |  |  |
| Exibir arquivos |  |  |  |  |  |  |  |
| Crie e envie commits |  |  |  |  |  |  |  |
| Exibir confirmações |  |  |  |  |  |  |  |
| Crie pull requests |  |  |  |  |  |  |  |
| Crie regras de aprovação para pull requests |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Substitua os requisitos de mesclagem para pull requests |  |  |  |  |  |  |  |
| Atualizar pull requests |  |  |  |  |  |  |  |
| Atualize as regras de aprovação para pull requests |  |  |  |  |  |  |  |
| Visualizar solicitações pull |  |  |  |  |  |  |  |
| Veja as regras de aprovação para pull requests |  |  |  |  |  |  |  |
| Fechar pull requests |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Aprovar pull requests |  |  |  |  |  |  |  |
| Comente sobre pull requests |  |  |  |  |  |  |  |
| Interaja com o Amazon Q nos comentários sobre pull requests |  |  |  |  |  |  |  |
| Crie uma revisão para uma pull request criada pela Amazon Q |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Vincular um problema a uma pull request |  |  |  |  |  |  |  |
| Desvincular um problema de uma pull request |  |  |  |  |  |  |  |
| Permissões do ambiente de desenvolvimento | | | | | | | |
| Crie seu próprio ambiente de desenvolvimento |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Pare seu próprio ambiente de desenvolvimento |  |  |  |  |  |  |  |
| Pare os ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |
| Retome seu próprio ambiente de desenvolvimento |  |  |  |  |  |  |  |





















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Veja seus próprios ambientes de desenvolvimento |  |  |  |  |  |  |  |
| Exibir ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |
| Edite seu próprio ambiente de desenvolvimento |  |  |  |  |  |  |  |


| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Editar ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |
| Exclua seu próprio ambiente de desenvolvimento |  |  |  |  |  |  |  |
| Excluir ambientes de desenvolvimento criados por outros usuários |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Crie um devfile para ambientes de desenvolvimento |  |  |  |  |  |  |  |
| Editar um devfile para Dev Environments |  |  |  |  |  |  |  |
| Excluir um devfile para Dev Environments |  |  |  |  |  |  |  |
| Exibir um devfile para Dev Environments |  |  |  |  |  |  |  |











| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Repositório de pacotes e permissões de pacotes | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Criar repositório de pacotes |  |  |  |  |  |  |  |
| Exibir repositórios de pacotes |  |  |  |  |  |  |  |
| Editar repositório de pacotes |  |  |  |  |  |  |  |
| Excluir repositório de pacotes |  |  |  |  |  |  |  |









| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Criar repositório de pacotes de gateway |  |  |  |  |  |  |  |
| Exibir repositórios de pacotes de gateway |  |  |  |  |  |  |  |
| Excluir repositório de pacotes do gateway |  |  |  |  |  |  |  |
| Adicionar repositório de pacotes upstream |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Editar a ordem de pesquisa dos repositórios upstream |  |  |  |  |  |  |  |
| Remover repositório de pacotes upstream |  |  |  |  |  |  |  |
| Conecte-se a um repositório de pacotes |  |  |  |  |  |  |  |
| Leia pacotes de um repositório de pacotes |  |  |  |  |  |  |  |





























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Publicar pacotes em um repositório de pacotes |  |  |  |  |  |  |  |
| Leia e retenha pacotes de um repositório upstream |  |  |  |  |  |  |  |
| Visualizar pacotes |  |  |  |  |  |  |  |
| Exibir versões do pacote |  |  |  |  |  |  |  |
| Exibir ativos da versão do pacote |  |  |  |  |  |  |  |









| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Listar dependências da versão do pacote |  |  |  |  |  |  |  |
| Atualizar o status da versão do pacote |  |  |  |  |  |  |  |
| Atualizar a configuração de origem do pacote |  |  |  |  |  |  |  |
| Excluir versão do pacote |  |  |  |  |  |  |  |




































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|------------------------------------|---|---|---|---|---|---|---|
| permissões de fluxo de trabalho | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Crie um fluxo de trabalho |  |  |  |  |  |  |  |
| Atualizar fluxo de trabalho |  |  |  |  |  |  |  |
| Excluir fluxo de trabalho |  |  |  |  |  |  |  |
| Iniciar fluxo de trabalho |  |  |  |  |  |  |  |
| Parar fluxo de trabalho |  |  |  |  |  |  |  |
| Crie segredos do fluxo de trabalho |  |  |  |  |  |  |  |






















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Atualizar segredos do fluxo de trabalho |  |  |  |  |  |  |  |
| Excluir segredos do fluxo de trabalho |  |  |  |  |  |  |  |
| Criar ambientes |  |  |  |  |  |  |  |
| Excluir ambientes |  |  |  |  |  |  |  |
| Crie uma frota |  |  |  |  |  |  |  |
| Atualizar frota |  |  |  |  |  |  |  |
| Excluir frota |  |  |  |  |  |  |  |























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Gerencie recursos de fluxo de trabalho em outras contas |  |  |  |  |  |  |  |
| Associar uma conexão VPC a um ambiente |  |  |  |  |  |  |  |
| Desassociar uma conexão VPC com um ambiente |  |  |  |  |  |  |  |
| Associe um ambiente conectado à VPC a um fluxo de trabalho |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Desassociar um ambiente conectado à VPC com um fluxo de trabalho |  |  |  |  |  |  |  |
| Aprovar execuções de fluxo de trabalho |  |  |  |  |  |  |  |
| Acompanhe uma confirmação em um fluxo de trabalho |  |  |  |  |  |  |  |
| Visualizar ambientes |  |  |  |  |  |  |  |









































| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Exibir registros de ações de compilação |  |  |  |  |  |  |  |
| Exibir frotas |  |  |  |  |  |  |  |
| Exibir registros de ações de teste |  |  |  |  |  |  |  |
| Exibir fluxo de trabalho |  |  |  |  |  |  |  |
| Visualizar execuções do fluxo de trabalho |  |  |  |  |  |  |  |
| Exibir resultados de execução do fluxo de trabalho |  |  |  |  |  |  |  |























| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--------------------------------------|---|---|---|---|---|---|---|
| Exibir segredos do fluxo de trabalho |  |  |  |  |  |  |  |
| Emitir permissões | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Criar problema |  |  |  |  |  |  |  |
| Problema de atualização |  |  |  |  |  |  |  |
| Exibir problemas |  |  |  |  |  |  |  |
| Arquivar um problema |  |  |  |  |  |  |  |


| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Atribua um problema à Amazon Q |  |  |  |  |  |  |  |
| Interaja com o Amazon Q nos comentários sobre um problema |  |  |  |  |  |  |  |
| Cancelar a atribuição do Amazon Q a um problema |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Problemas de atualização criados por outros usuários |  |  |  |  |  |  |  |
| Exibir comentários sobre um problema |  |  |  |  |  |  |  |
| Crie um comentário sobre um problema |  |  |  |  |  |  |  |
| Atualizar um comentário sobre um problema |  |  |  |  |  |  |  |
| Crie uma etiqueta |  |  |  |  |  |  |  |






















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Atualizar um rótulo |  |  |  |  |  |  |  |
| Exibir rótulos |  |  |  |  |  |  |  |
| Adicionar um rótulo a um problema |  |  |  |  |  |  |  |
| Remover uma etiqueta de uma edição |  |  |  |  |  |  |  |
| Crie um status personalizado para problemas |  |  |  |  |  |  |  |
| Atualizar um status personalizado |  |  |  |  |  |  |  |





| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|-----------------------------------|---|---|---|---|---|---|---|
| Exibir um status personalizado |  |  |  |  |  |  |  |
| Mover um status personalizado |  |  |  |  |  |  |  |
| Desativar um status personalizado |  |  |  |  |  |  |  |
| Adicionar um anexo a uma edição |  |  |  |  |  |  |  |
| Exibir um anexo de problema |  |  |  |  |  |  |  |
| Remover um anexo de uma edição |  |  |  |  |  |  |  |



















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Vincular uma pull request a um problema |  |  |  |  |  |  |  |
| Desvincular uma pull request de um problema |  |  |  |  |  |  |  |
| Vincular um projeto do Jira |  |  |  |  |  |  |  |
| Desvincular um projeto do Jira |  |  |  |  |  |  |  |
| Permissões personalizadas do blueprint | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Crie um projeto de blueprint personalizado |  |  |  |  |  |  |  |
| Publique uma prévia do blueprint personalizado |  |  |  |  |  |  |  |
| Cancelar a publicação de uma prévia do blueprint personalizado |  |  |  |  |  |  |  |
| Publique um blueprint personalizado |  |  |  |  |  |  |  |
















| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Cancelar a publicação de um blueprint personalizado |  |  |  |  |  |  |  |
| Adicionar um blueprint personalizado a um catálogo de blueprints espaciais |  |  |  |  |  |  |  |
| Remover um blueprint personalizado de um catálogo de blueprints espaciais |  |  |  |  |  |  |  |








| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Gerencie as permissões de publicação para um blueprint personalizado |  |  |  |  |  |  |  |
| Gerencie a versão do catálogo para um blueprint personalizado |  |  |  |  |  |  |  |
| Atualizar um blueprint personalizado |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|--|---|---|---|---|---|---|---|
| Excluir uma versão personalizada do blueprint |  |  |  |  |  |  |  |
| Excluir um blueprint personalizado |  |  |  |  |  |  |  |
| Aplicar um blueprint personalizado a um projeto |  |  |  |  |  |  |  |
| Desassociar um blueprint personalizado de um projeto |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Atualizar a versão de um blueprint personalizado aplicado |  |  |  |  |  |  |  |
| Edite o alias de um blueprint personalizado |  |  |  |  |  |  |  |
| Veja os esquemas personalizados publicados |  |  |  |  |  |  |  |
| Permissões de notificações | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|-------------------------------------|---|---|---|---|---|---|---|
| Configurar canal de notificação |  |  |  |  |  |  |  |
| Remover canal de notificação |  |  |  |  |  |  |  |
| Editar configurações de notificação |  |  |  |  |  |  |  |
| Exibir configurações de notificação |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---|---|---|---|---|---|---|---|
| Receba notificações automáticas sobre CodeCatalyst incidentes |  |  |  |  |  |  |  |
| Configurar notificações por e-mail para sua conta de e-mail associada |  |  |  |  |  |  |  |
| Permissões de pesquisa | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
| Pesquise dentro de um projeto |  |  |  |  |  |  |  |

| Permissão | Função de administrador de espaço | Função de usuário avançado | Função de acesso limitado | Função de administrador do projeto | Função de colaborador | Função de revisor | Função somente para leitura |
|---------------------------|---|---|---|---|---|---|---|
| Pesquise em todo o espaço |  |  |  |  |  |  |  |

Visualizando e alterando funções de usuário

Você pode ver a função atribuída a um usuário. Isso ajuda você a entender quais ações eles podem realizar em um projeto. Você também pode alterar a função deles se precisarem de permissões adicionais.

Para visualizar a função de um usuário em um projeto

1. Navegue até o projeto em que você deseja visualizar as funções associadas a cada membro do projeto.

Tip

Você pode escolher qual projeto visualizar na barra de navegação superior.

2. No painel de navegação, escolha Configurações do projeto.
3. Na guia Membros, a função de cada membro do projeto é exibida em Papel.

Para alterar as funções dos usuários em um projeto

1. Navegue até o projeto em que você deseja alterar as funções associadas aos membros do projeto.

Tip

Você pode escolher qual projeto 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 cujas funções você deseja alterar. Escolha Ação e, em seguida, escolha Editar função.
4. Em Função, escolha a função do projeto e, em seguida, escolha Confirmar.

Visualizando e alterando funções 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 as funções dos usuários em Acesso limitado ao administrador do espaço para gerenciar melhor seu espaço e seus 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 essa função somente aos usuários que precisam desse nível de acesso ao seu espaço. Para ter mais informações, consulte [Função de administrador de espaço](#).

Para alterar a função 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 cuja função você deseja alterar e, em seguida, escolha Alterar função.
5. Em Alterar função, escolha a função que você deseja atribuir e escolha Confirmar.

Conceda 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. Os PATs que você cria são 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 PATs que 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, os PATs expiram em 1 ano.

Criação de PATs

Os PATs estão associados à sua identidade de usuário em CodeCatalyst. Você só pode copiar um segredo PAT no momento em que o cria.

Criação de PATs (console)

Você pode usar o console para criar PATs no 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 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 PAT, escolha Copiar. Armazene o segredo do PAT onde você poderá recuperá-lo.

 Important

O segredo do PAT só é exibido uma vez. Você não pode recuperá-lo depois de fechar a janela. Se você não salvou o segredo do PAT em um local seguro, você pode criar outro.

Criação de 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 create-access-token comando da seguinte maneira.

```
aws codecatalyst create-access-token
```

Se for bem-sucedido, o comando retornará informações sobre o PAT criado, como no exemplo a seguir.

```
{
```

```
"secret": "value",  
"name": "marymajor-2222EXAMPLE",  
"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 perdeu um segredo PAT ou está preocupado com o fato de ele não estar armazenado com segurança, você pode criar outro.

Você pode visualizar os PATs associados à sua conta de usuário usando o AWS CLI. 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 ter mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

Visualizando PATs

Você pode ver os PATs em CodeCatalyst. A lista mostra todas as 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 PATs expirados não são exibidos porque são excluídos após expirarem.

Visualizando PATs (console)

Você pode usar o console para visualizar os PATs associados à sua identidade de usuário no 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 PATs atuais.

Visualizando 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 `list-access-tokens` comando da seguinte maneira.

```
aws codecatalyst list-access-tokens
```

Se for bem-sucedido, o comando retornará informações sobre os PATs associados à sua conta de usuário, como no exemplo a seguir.

```
{
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
      "name": "marymajor-22222EXAMPLE",
      "expiresTime": "2024-02-04T01:56:04.402000+00:00"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
      "name": "marymajor-11111EXAMPLE",
      "expiresTime": "2023-03-12T01:58:40.694000+00:00"
    }
  ]
}
```

Excluindo PATs

Você pode excluir PATs associados à sua identidade de usuário em CodeCatalyst.

Excluindo PATs (console)

Você pode usar o console para excluir PATs em 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, em seguida, escolha Excluir.

<name>No Delete PAT:? página, 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 o AWS CLI Para fazer isso, você deve fornecer o ID do PAT, que pode ser visualizado usando o delete-access-token comando.

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](#).

Para excluir um token de acesso pessoal (AWS CLI)

- No terminal ou na linha de comando, execute o `delete-access-token` comando, 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 for bem-sucedido, esse comando não retornará nenhuma resposta.

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. GitHub Para

obter mais informações sobre extensões, consulte [Adicione funcionalidade a projetos com extensões no CodeCatalyst](#).

Criando 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.

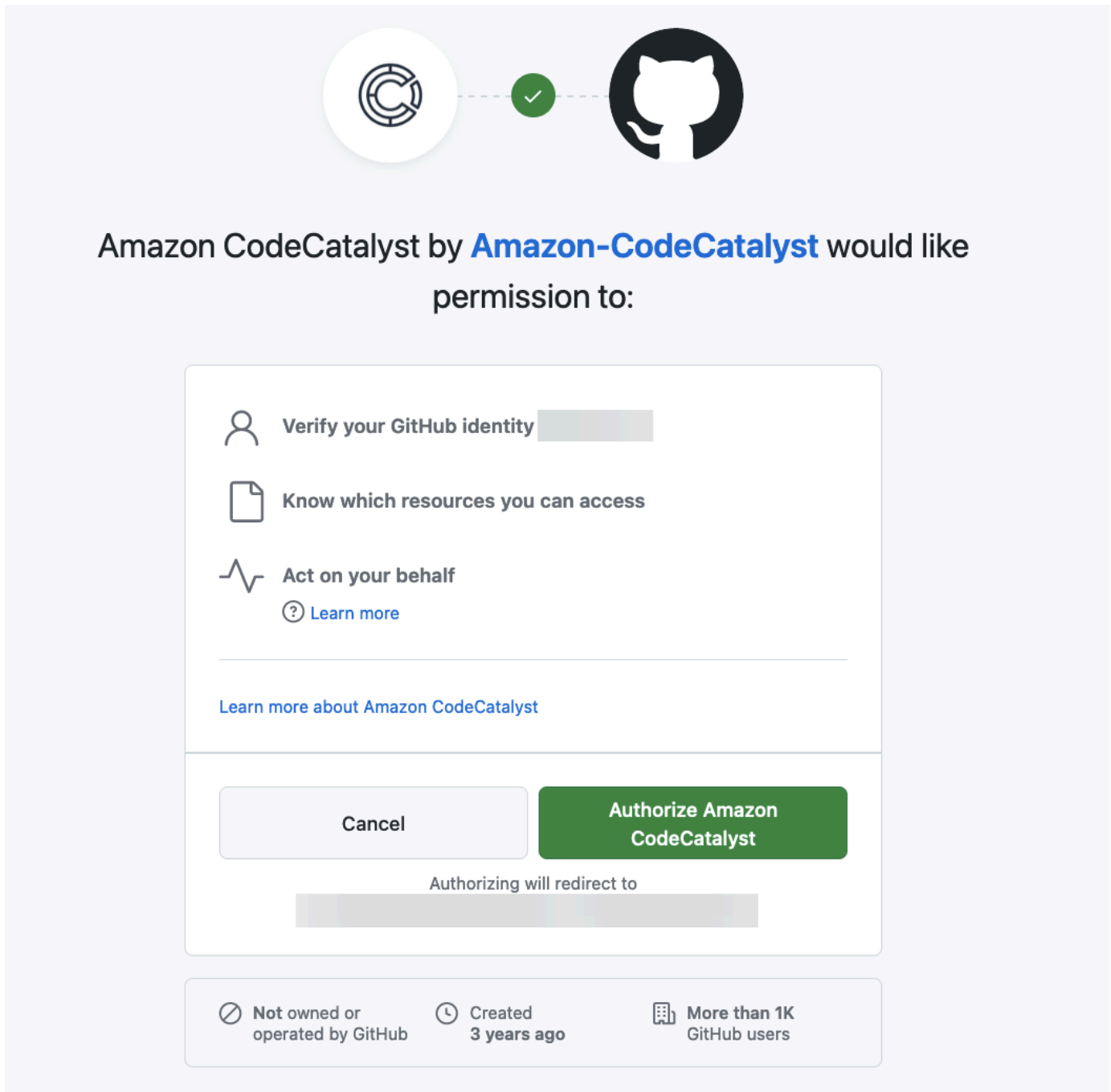
Tip

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, escolha Criar.

A página Criar conexão é exibida.

4. Selecione Create (Criar). A página Criar conexão é exibida.
5. Na página Criar conexão, em Provedor, escolha GitHub. Em Nome da conexão, digite um nome para sua conexão. Selecione Create (Criar).
6. Se solicitado, faça login na sua GitHub conta.
7. Na página de confirmação da conexão, escolha Aceitar.
8. Na página de confirmação da instalação, escolha o botão de autorização para confirmar que você deseja instalar o aplicativo do conector.



Excluindo 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 nova conexão pessoal, a instalação do aplicativo 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, em seguida, escolha Excluir.

Na conexão Excluir:? <name> página, 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, escolha Configurações e, em seguida, escolha Aplicativos.
2. Na guia GitHub Aplicativos autorizados, na lista de aplicativos autorizados, visualize o aplicativo instalado para CodeCatalyst. Para revogar a instalação, escolha 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 para registrar seu novo dispositivo para autenticação multifator (MFA).

Note

Recomendamos que você primeiro baixe o aplicativo autenticador apropriado em seu 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, em seguida, escolha Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, escolha Gerenciar perfil e segurança. A página do perfil AWS Builder ID é aberta.
4. No lado esquerdo da página, escolha Segurança.
5. Na página Autenticação multifator, escolha Registrar dispositivo.
6. Na página Registrar dispositivo de MFA, escolha 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 seu dispositivo for registrado com sucesso, você terá a opção de associar um nome de exibição amigável ao seu 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 o aplicativo de MFA suportar vários dispositivos, escolha a opção de criar um novo dispositivo de MFA.
 - b. Determine se o aplicativo MFA suporta códigos QR e, em seguida, execute uma das seguintes ações na página Set up the authenticator app:
 - i. No assistente, escolha Show QR code e, em seguida, use o app para digitalizar o código de QR. Por exemplo, você pode escolher o ícone da câmera ou escolher uma opção semelhante a Scan code. Em seguida, use a câmera do dispositivo para digitalizar o código.
 - ii. Escolha mostrar chave secreta e, em seguida, insira essa chave secreta em seu aplicativo de MFA.

 Important

Ao configurar um dispositivo de MFA para AWS Builder ID, 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 o aplicativo 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 de MFA agora pode começar a gerar senhas de uso único e agora está pronto para uso.

Aplicativos autenticadores

Os aplicativos autenticadores são autenticadores terceirizados baseados em senhas de uso único (OTP). Os usuários podem usar um aplicativo autenticador instalado em seu dispositivo móvel ou tablet como um dispositivo de MFA autorizado. O aplicativo autenticador de terceiros deve estar em conformidade com o RFC 6238, que é um algoritmo TOTP (senha única baseada em tempo) baseado em padrões capaz de gerar códigos de autenticação de seis dígitos.

Quando a MFA for solicitada, você deve inserir um código válido do seu aplicativo autenticador na caixa de entrada apresentada. Cada dispositivo MFA atribuído a um usuário deve ser exclusivo. Dois aplicativos autenticadores podem ser registrados para qualquer usuário.

Aplicativo autenticador testado

Embora qualquer aplicativo compatível com TOTP funcione com o IAM Identity Center MFA, a tabela a seguir lista aplicativos autenticadores terceirizados conhecidos para você escolher.

| Sistema operacional | Aplicativo autenticador testado |
|---------------------|---|
| Android | Authy , Duo Mobile , Autenticador , Microsoft LastPass Authenticator , Google Authenticator |
| iOS | Authy , Duo Mobile , Autenticador , Microsoft LastPass Authenticator , Google Authenticator |

Alterando seus dispositivos de MFA

Depois de registrar um dispositivo de MFA, você pode alterar seu 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](#).

Renomeando um dispositivo de MFA

Como renomear seu 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, em seguida, escolha Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, escolha Gerenciar perfil e segurança. A página do perfil AWS Builder ID é aberta.
4. Escolha Autenticação multifator no lado esquerdo da página. Você verá que a opção Renomear fica esmaecida quando você chega à página.
5. Selecione o dispositivo de MFA que você deseja alterar. Escolha Rename (Renomear). Em seguida, um modal aparece.
6. No prompt que é aberto, insira o novo nome em Nome do dispositivo MFA e escolha Renomear. O dispositivo renomeado aparece em Dispositivos de autenticação multifator (MFA).

Excluindo 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, em seguida, escolha Perfil do usuário. A página CodeCatalyst Perfil é aberta.
3. Na página do perfil, escolha Gerenciar perfil e segurança. A página do perfil AWS Builder ID é aberta.
4. Escolha Autenticação multifator no lado esquerdo da página. Você verá que a opção Excluir fica esmaecida quando você chega à 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 aparece mais em Dispositivos de 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

- [Proteção de dados na Amazon CodeCatalyst](#)
- [Identity and Access Management e Amazon CodeCatalyst](#)
- [Validação de conformidade para a Amazon CodeCatalyst](#)
- [Resiliência na Amazon CodeCatalyst](#)
- [Segurança de infraestrutura na Amazon CodeCatalyst](#)
- [Análise de configuração e vulnerabilidade na Amazon CodeCatalyst](#)
- [Seus dados e privacidade na Amazon CodeCatalyst](#)
- [Melhores práticas para ações de fluxo de trabalho na Amazon CodeCatalyst](#)
- [Entendendo o modelo de CodeCatalyst confiança](#)

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 que hospeda 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 multifator 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 os endereços de e-mail de seus 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 conectadoContas da AWS. Por exemplo, não insira informações confidenciais ou sigilosas como parte dos nomes do espaço, do projeto ou da frota de implantação. Todos os dados inseridos em tags, nomes ou campos de formato livre usados para nomes podem ser usados para registros de faturamento ou diagnóstico ou podem ser incluídos em caminhos de URL. Isso se aplica ao uso do console, da APIAWS CLI, do CodeCatalyst Action Development Kit ou de qualquer AWS SDKs.

Se você fornecer uma URL para um servidor externo, recomendamos que você não inclua nenhuma informação de credenciais de segurança na URL para validar sua solicitação para 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 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. Todos 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 uma 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 obter mais informações, consulte a documentação do serviço vinculado [Melhores práticas para ações de fluxo de trabalho na Amazon CodeCatalyst](#) e.

Gerenciamento de chaves

CodeCatalyst não oferece suporte ao gerenciamento de chaves.

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 deixam isso 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 local Região da AWS ao criar seu espaço, seus dados não serão replicados

para regiões nas 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 para o espaço que auxiliam na navegação entre espaços
- O Região da AWS local onde o espaço está localizado
- Os IDs exclusivos de todos os projetos no espaço
- O ID da função que indica a função 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. Certifique-se de não incluir 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 entanto, CodeCatalyst se integra com o 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. Isso é chamado de [função 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 Guia do CodeCatalyst Administrador da Amazon.

Tópicos

- [Políticas baseadas em identidade no IAM](#)
- [Ações políticas no IAM](#)
- [Recursos de política no IAM](#)
- [Chaves de condição de política no IAM](#)
- [Exemplos de políticas baseadas em identidade para conexões CodeCatalyst](#)
- [Usando tags para controlar o acesso aos recursos de conexão da conta](#)
- [CodeCatalyst referência de permissões](#)
- [Usar perfis vinculados a serviço do 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

Políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade. Essa identidade pode ser um usuário, um grupo de usuários ou uma função. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em quais condições. Saiba como criar uma política baseada em identidade consultando [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 função à qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elementos da 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 políticas no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual diretor pode realizar quais ações em quais recursos e sob quais condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar 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. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
  "prefix:action1",  
  "prefix:action2"
```

]

Recursos de política no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual diretor pode realizar quais ações em quais recursos e sob quais condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no 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 política no IAM

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual diretor pode realizar quais ações em quais recursos e sob quais condições.

O elemento `Condition` (ou bloco `Condition`) permite especificar condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. Você pode criar expressões condicionais que usem [operadores de condição](#), como “igual a” ou “menor que”, para corresponder a condição da política aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, 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 de 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 AWS chaves de condição globais da , consulte [AWS Chaves de contexto de condição globais da](#) 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 Contas da AWS a um espaço. Políticas baseadas em identidade são usadas nos conectados. Contas da AWS

Por padrão, usuários e funções não têm permissão para criar ou modificar CodeCatalyst recursos. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, 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.

O exemplo a seguir de políticas do IAM concede permissões para ações relacionadas às conexões da conta. 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 AWS Management Console com a mesma conta especificada na solicitação.

```
{
  "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 a CodeCatalystWorkflowDevelopmentRole-*spaceName* função escolhendo a opção no AWS Management Console. No exemplo de política, a condição para a `iam:PassRole` ação inclui os princípios de serviço para CodeCatalyst. Somente funções com esse acesso serão criadas no AWS Management Console.

```
{
  "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

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*"
    }
  ]
}

```

Usando tags para controlar o acesso aos recursos de conexão da conta

As tags podem ser anexadas ao recurso ou passadas na solicitação para serviços que oferecem suporte à marcação. Os recursos nas políticas podem ter tags, e algumas ações nas políticas podem incluir tags. As chaves de condição de marcação incluem as chaves de `aws:ResourceTag` condição `aws:RequestTag` e. Ao criar uma política do IAM, você poderá usar chaves de condição de tag para controlar o seguinte:

- Quais usuários podem realizar ações em um recurso de conexão, com base nas tags que ele 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 política 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.

```
{
  "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 realizar ações e obter informações sobre os recursos de conexão da conta.

Para fazer isso, ele permite ações específicas se a conexão tiver uma tag nomeada `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.)

```
{
  "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 conta

As permissões a seguir são necessárias para trabalhar com conexões de conta.

| 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 | Obrigatório para associar uma função 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 | Obrigatório para desassociar uma função do IAM de 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 | Obrigatório para descrever a autorização de cobrança 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> |
| GetConnection | Necessário para obter uma conexão com a conta. Esta | arn:aws:codecatalyst:region: <i>account_ID</i> |

| CodeCatalyst permissões para conexões de contas | Permissões obrigatórias | Recursos |
|---|--|--|
| | é apenas uma permissão de política do IAM, e não uma ação de API. | <i>D</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 conta que não estão 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 | Obrigatório para listar as funções do IAM 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> |
| ListTagsForResource | Obrigatório para listar as 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> |

| CodeCatalyst permissões para conexões de contas | Permissões obrigatórias | Recursos |
|---|---|--|
| PutBillingAuthorization | Necessário para criar ou atualizar a autorização de cobrança 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 aplicativos do IAM Identity Center

As permissões a seguir são necessárias para trabalhar com aplicativos do IAM Identity Center.

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias | Recursos |
|---|---|--|
| AssociateIdentityCenterApplicationToSpace | Necessário 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 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> |
| CreateIdentityCenterApplication | Necessário para criar um aplicativo do IAM Identity | arn:aws:codecatalyst:region: <i>account_ID</i> |

| CodeCatalyst permissões para aplicativos do IAM Identity Center | Permissões obrigatórias | Recursos |
|---|---|--|
| | Center. Esta é apenas uma permissão de política do IAM, e não uma ação de API. | <i>D</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 um 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> |
| 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 obter informações sobre um 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> |
| ListIdentityCenterApplications | Obrigatório para visualizar uma lista de todos os aplicativos do IAM Identity Center 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 um aplicativo do IAM Identity Center com o repositório de identidade 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 um 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> |

Usar perfis vinculados a serviço do CodeCatalyst

A Amazon CodeCatalyst usa AWS Identity and Access Management funções [vinculadas a serviços](#) (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a CodeCatalyst. As funções vinculadas ao serviço são predefinidas CodeCatalyst e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração CodeCatalyst porque você não precisa adicionar manualmente as permissões necessárias. CodeCatalyst define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só CodeCatalyst pode

assumir suas funções. 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 CodeCatalyst recursos porque você não pode remover inadvertidamente a permissão 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 função vinculadas ao serviço para CodeCatalyst

CodeCatalyst usa a função vinculada ao serviço chamada `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`— Permite que a Amazon tenha acesso CodeCatalyst somente de leitura aos perfis de instância do aplicativo e aos usuários e grupos de diretórios associados em seu nome.

A função vinculada ao serviço

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` confia nos seguintes serviços para aceitar a função:

- `codecatalyst.amazonaws.com`

A política de permissões de função nomeada

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` CodeCatalyst permite concluir 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 uma função vinculada ao serviço para CodeCatalyst

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um espaço na AWS Management Console, na ou na AWS CLI, CodeCatalyst cria a função vinculada ao 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 CodeCatalyst serviço antes de 17 de novembro de 2023, quando ele começou a oferecer suporte a funções vinculadas ao serviço, CodeCatalyst criou a `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` função 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, você poderá usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um espaço, CodeCatalyst cria a função vinculada ao serviço para você novamente.

Você também pode usar o console do IAM para criar uma função vinculada ao serviço com os perfis de instância do aplicativo View e o caso de uso de usuários e grupos do diretório associado. 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 um perfil vinculado a serviço](#) no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Editando uma função vinculada ao serviço para CodeCatalyst

CodeCatalyst não permite que você edite a função `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` vinculada ao serviço. 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 da função usando o IAM. Para ter mais informações, consulte [Editar um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Excluindo uma função vinculada ao serviço para CodeCatalyst

Você não precisa excluir manualmente a função `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Quando você exclui

um espaço na AWS Management Console, na AWS CLI ou na AWS API, CodeCatalyst limpa os recursos e exclui a função vinculada 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 CodeCatalyst serviço estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir CodeCatalyst recursos usados pelo `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`

- [Exclua o espaço](#).

Como excluir manualmente o perfil vinculado 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 suportadas para funções vinculadas a CodeCatalyst serviços

CodeCatalyst suporta o uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Regiões e endpoints da AWS](#).

CodeCatalyst não oferece suporte ao uso de funções vinculadas ao serviço 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 | Support em 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 | Support em CodeCatalyst |
|--|----------------------|-------------------------|
| Oeste dos EUA (Norte 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 | Support em 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 |

AWSPolíticas gerenciadas para a Amazon CodeCatalyst

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da AWS usarem. Recomendamos que você reduza ainda mais as permissões definindo [políticas gerenciadas pelo cliente](#) específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em um política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [AWSPolíticas gerenciadas pela](#) no Manual do usuário do IAM.

AWSPolítica gerenciada da : 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 cobrança

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.

```
{
  "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": "*"
    }
  ]
}
```

```

]
}

```

AWSPolítica gerenciada da : AmazonCodeCatalystFullAccess

Esta é uma política que concede permissões para gerenciar seu CodeCatalyst espaço e contas conectadas na página Amazon CodeCatalyst Spaces noAWS Management Console. 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 noAWS Management Console.

```

{
  "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"
        ]
    }
}
]
}

```

AWSPolítica gerenciada da : AmazonCodeCatalystReadOnlyAccess

Essa é uma política que concede permissões para visualizar e listar informações de espaços e contas conectadas na página Amazon CodeCatalyst Spaces noAWS Management Console. 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. AWS Management Console

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:Get*",
        "codecatalyst:List*"
      ],
      "Resource": "*"
    }
  ]
}

```


AWSPolítica gerenciada da :

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy

Você não pode

anexarAmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy; às suas entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite CodeCatalyst realizar ações em seu nome. Para ter mais informações, consulte [Usar perfis vinculados a serviço do 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:ListApplications",
        "sso:ListApplicationAssignments",
        "sso:DescribeInstance",
        "sso:DescribeApplication"
      ],
      "Resource": "*"
    }
  ]
}
```

```

}
]
}

```

Atualizações do CodeCatalyst para políticas gerenciadas pela AWS

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 |
|--|---|------------------------|
| AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy – Nova política | CodeCatalyst adicionou a política.

Concede permissões para permitir que CodeCatalyst os usuários visualizem perfis de instâncias de aplicativos e usuários e grupos de diretórios associados. | 17 de novembro de 2023 |
| AmazonCodeCatalystSupportAccess – Nova política | CodeCatalyst adicionou a política.

Concede permissões para permitir que CodeCatalyst os usuários pesquisem, criem e resolvam casos de suporte, bem como visualizem comunicações e detalhes relacionados. | 20 de abril de 2023 |
| AmazonCodeCatalystFullAccess – Nova política | CodeCatalyst adicionou a política.

Concede acesso total CodeCatalyst a. | 20 de abril de 2023 |

| Alteração | Descrição | Data |
|--|---|---------------------|
| AmazonCodeCatalystReadOnlyAccess – Nova política | CodeCatalyst adicionou a política.

Concede acesso somente para leitura a. CodeCatalyst | 20 de abril de 2023 |
| CodeCatalyst começou a rastrear 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 as seguintes funções de serviço e associá-las ao conectar sua conta.

Para obter mais informações sobre os elementos que você usa 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 escolher criar e usar a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço ou criar funções de serviço personalizadas e configurar essas políticas e funções do IAM manualmente. Como prática recomendada, atribua a essas funções a menor quantidade de permissões necessárias.

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 `AWSRoleForCodeCatalystSupportserviço` que permite que CodeCatalyst os usuários acessem o suporte. Para obter mais informações sobre suporte para um CodeCatalyst espaço, consulte [AWS Support para Amazon CodeCatalyst](#).

Entendendo a função `CodeCatalystWorkflowDevelopmentRole-spaceName` de serviço

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. Isso é chamado de [função de serviço](#). A maneira mais simples de criar uma função de serviço é adicionar uma ao criar o espaço e escolher a `CodeCatalystWorkflowDevelopmentRole-spaceName` opção para essa função. 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. A função de serviço tem como escopo o espaço, não os projetos individuais. Para criar essa função, consulte [Criando a `CodeCatalystWorkflowDevelopmentRole-spaceName` função para sua conta e espaço](#). Você só pode criar uma função 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 à `CodeCatalystWorkflowDevelopmentRole-spaceName` função foi projetada para funcionar com projetos criados com plantas no espaço. Ele permite que os usuários desses projetos desenvolvam, criem, testem e implantem código usando recursos conectados Conta da AWS. 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/*"
        }
      }
    }
  ]
}
```

Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName* função para sua conta e espaço

Siga estas etapas para criar a CodeCatalystWorkflowDevelopmentRole-*spaceName* função que será usada para fluxos de trabalho em seu espaço. Para cada conta que você deseja que tenha funções do IAM para uso em projetos, no seu espaço, você deve adicionar uma função, como a função 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 AWS Management Console, 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.

- Escolha o link para Conta da AWS onde você deseja criar a função. A página de Conta da AWS detalhes é exibida.
- Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

- Escolha Criar função CodeCatalyst de administrador de desenvolvimento no IAM. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento. O papel terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço](#).

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.

- Escolha Criar função 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.

Entendendo a função `AWSRoleForCodeCatalystSupport` de serviço

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 [função de serviço](#) para suporte. A maneira mais simples de criar uma função de serviço para suporte é adicionar uma ao criar o espaço e escolher a `AWSRoleForCodeCatalystSupport` opção para essa função. 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. A função de serviço tem como escopo o espaço, não os projetos individuais. Para criar essa função, consulte [Criando a `AWSRoleForCodeCatalystSupport` função para sua conta e espaço](#).

A política anexada à `AWSRoleForCodeCatalystSupport` função é uma política gerenciada que fornece acesso às permissões de suporte. Para ter mais informações, consulte [AWSPolítica gerenciada da : AmazonCodeCatalystSupportAccess](#).

A função de confiança da política CodeCatalyst permite assumir a função.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst.amazonaws.com",
          "codecatalyst-runner.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Criando a `AWSRoleForCodeCatalystSupport` função para sua conta e espaço

Siga estas etapas para criar a `AWSRoleForCodeCatalystSupport` função que será usada para casos de suporte em seu espaço. A função deve ser adicionada à conta de cobrança 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 AWS Management Console, 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 de Conta da AWS detalhes é exibida.

4. Escolha Gerenciar funções em AWS Management Console.

A página Adicionar função do IAM ao CodeCatalyst espaço da Amazon é aberta no AWS Management Console. Esta é a página do Amazon CodeCatalyst Spaces. Talvez seja necessário fazer login para acessar a página.

5. Em detalhes do CodeCatalyst espaço, escolha Adicionar função de CodeCatalyst Support. Essa opção cria uma função de serviço que contém a política de permissões e a política de confiança para a função de desenvolvimento de pré-visualização. A função terá um nome `AWSRoleForCodeCatalystSupport` com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função AWSRoleForCodeCatalystSupport de serviço](#).
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 obter instruções sobre como criar funções de exemplo, consulte [Criação manual de funções para ações de fluxo de trabalho](#). Depois de criar sua função do IAM, copie o ARN da função para adicionar a função do IAM à conexão da sua conta e associá-la ao ambiente do projeto. Para saber mais, consulte [Adicionar funções 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 AWS CloudFormation recursos em seu Conta da AWS.

Essa função dá permissões para fazer o seguinte:

- Grave nos buckets do Amazon S3.

- Support a construção de recursos com AWS CloudFormation. Isso requer acesso ao Amazon S3.

Essa função usa a seguinte política:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  }]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

CodeCatalyst criar função para AWS 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 AWS CloudFormation recursos em seu Conta da AWS.

Essa função dá permissões para fazer o seguinte:

- Support a construção de recursos com AWS CloudFormation. Isso é necessário junto com a função de CodeCatalyst criação para acesso ao Amazon S3 e a função de CodeCatalyst implantação para. AWS CloudFormation

As seguintes políticas AWS gerenciadas devem ser anexadas a essa função:

- AWSCloudFormationFullAccess
- IAM FullAccess
- Amazon S3 FullAccess
- API da Amazon GatewayAdministrator
- 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 AWS CloudFormation recursos em seu. Conta da AWS

Essa função dá permissões para fazer o seguinte:

- Grave nos buckets do Amazon S3.
- Support a construção de construções de CDK e pilhas de AWS 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.

Essa função usa a seguinte política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
```

```
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
    ],
    "Resource": "*"
}
]
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

CodeCatalyst função de implantação para AWS CloudFormation

Para ações CodeCatalyst de implantação de fluxo de trabalho que usam AWS 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 AWS CloudFormation recursos em seu Conta da AWS.

Essa função dá permissões para fazer o seguinte:

- Permita CodeCatalyst invocar uma função λ para realizar a implantação azul/verde. AWS CloudFormation
- Permite CodeCatalyst criar e atualizar pilhas e conjuntos de alterações em. AWS CloudFormation

Essa função 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 a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

CodeCatalyst função de implantação para o 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 definidas que CodeCatalyst precisa executar tarefas nos recursos do Amazon EC2 em seu. Conta da AWS A política padrão para a CodeCatalystWorkflowDevelopmentRole-*spaceName* função não inclui permissões para o Amazon EC2 ou o Amazon EC2 Auto Scaling.

Essa função dá permissões para fazer o seguinte:

- Crie implantações do Amazon EC2.
- Leia as tags em uma instância ou identifique uma instância do Amazon EC2 pelos nomes dos grupos do Auto Scaling.
- Ler, criar, atualizar e excluir grupos do Amazon EC2 Auto Scaling, ganchos do ciclo de vida e políticas de escalabilidade.
- Publique informações nos tópicos do Amazon SNS.
- Recupere informações sobre CloudWatch alarmes.
- Leia e atualize o Elastic Load Balancing.

Essa função usa a seguinte política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling>DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling>CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:PutWarmPool",
        "autoscaling:DescribeScalingActivities",
        "autoscaling>DeleteAutoScalingGroup",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:TerminateInstances",
        "tag:GetResources",
        "sns:Publish",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:RegisterTargets",
```

```
"elasticloadbalancing:DeregisterTargets"  
],  
"Resource": "resource_ARN"  
  }  
]  
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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

Essa função dá permissões para fazer 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.
- Invoque as funções do Lambda.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Essa função usa a seguinte política:

```
{
  "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 a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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

Essa função dá permissões para fazer o seguinte:

- Leia, atualize e invoque funções e aliases do Lambda.

- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre alarmes de CloudWatch eventos.
- Publique informações nos tópicos do Amazon SNS.

Essa função usa a seguinte política:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/CodeDeploy/",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
      },
      "Effect": "Allow"
    },
    {
```

```

        "Action": [
            "lambda:InvokeFunction"
        ],
        "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
        "Effect": "Allow"
    }
]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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

Essa função dá permissões para fazer o seguinte:

- Leia, atualize e invoque funções e aliases do Lambda.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Essa função usa a seguinte política:

```

*{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [

```

```

        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
},
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
    "Effect": "Allow"
}
]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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 AWS CloudFormation recursos em seu Conta da AWS.

Essa função dá permissões para fazer 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 AWS CloudFormation

Essa função usa a seguinte política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
```

```

        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
    ],
    "Resource": "*"
}
]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

CodeCatalyst função somente de leitura para o 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 definidas que CodeCatalyst precisa executar tarefas nos recursos do Amazon EC2 em seu. Conta da AWS A função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço não inclui permissões para o Amazon EC2 ou as ações descritas para a Amazon. CloudWatch

Essa função dá permissões para fazer o seguinte:

- Obtenha o status das instâncias do Amazon EC2.
- Obtenha CloudWatch métricas para instâncias do Amazon EC2.

Essa função usa a seguinte política:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe",
      "Resource": "resource_ARN"
    }
  ]
}

```

```
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe"
      ],
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe",
      "Resource": "resource_ARN"
    }
  ]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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

Essa função dá permissões para fazer o seguinte:

- Leia os conjuntos de tarefas do Amazon ECS.

- Recupere informações sobre CloudWatch alarmes.

Essa função usa a seguinte política:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:DescribeServices",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeRules"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
      },
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": [
```

```

        "arn:aws:iam:::role/ecsTaskExecutionRole",
        "arn:aws:iam:::role/ECSTaskExecution"
    ],
    "Condition": {
        "StringLike": {
            "iam:PassedToService": [
                "ecs-tasks.amazonaws.com"
            ]
        }
    }
}
]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, 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

Essa função dá permissões para o seguinte:

- Leia as 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 .

```

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

```



```

"Statement": [
  {
    "Action": [
      "cloudwatch:DescribeAlarms",
      "lambda:GetAlias",
      "lambda:GetProvisionedConcurrencyConfig"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  }
]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

Criação manual de funções 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 essas funções no IAM.

Para criar uma função de implantação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Criar política.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-deploy-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    }  
  ]  
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-deploy-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

```
codecatalyst-deploy-role
```

- i. Em Descrição da função, insira:

```
CodeCatalyst deploy role
```

- j. Selecione Criar função.

Agora você criou uma função de implantação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN da função de implantação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-deploy-role`).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de implantação com as permissões apropriadas e obteve seu ARN.

Para criar uma função de criação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.

- b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, escolha Políticas.
- d. Escolha Criar política.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Selecione Next (Próximo).
- f. Em Políticas de permissões, pesquise `codecatalyst-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

codecatalyst-build-role

- i. Em Descrição da função, insira:

CodeCatalyst build role

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de confiança e uma política de permissões.

3. Obtenha o ARN da função de construção, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-build-role`).
 - c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de criação com as permissões apropriadas e obteve seu ARN.

Para criar uma função de pilha

Note

Você não precisa criar uma função de pilha, embora isso seja recomendado por motivos de segurança. Se você não criar a função de pilha, precisará adicionar as políticas de permissões descritas mais adiante neste procedimento à função de implantação.

1. Faça login AWS usando a conta na qual você deseja implantar sua pilha.
2. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
3. No painel de navegação, escolha Funções e, em seguida, escolha Criar função.
4. Na parte superior, escolha o AWS serviço.
5. Na lista de serviços, escolha CloudFormation.
6. Selecione Next: Permissions (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, AWS CloudFormation gera erros que mostram quais permissões você precisa adicionar.

8. Escolha Próximo: etiquetas.
9. Selecione Next: Review (Próximo: revisar).
10. Em Nome da função, digite:

```
codecatalyst-stack-role
```

11. Selecione Criar função.
12. Para obter o ARN da função de pilha, faça o seguinte:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-stack-role`).
 - c. Escolha a função na lista.
 - d. Na página Resumo, copie o valor do ARN da função.

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 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 função e uma política 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: '*'

```

Criando a função manualmente para o blueprint do aplicativo 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 a função no IAM.

Para criar uma função de criação

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Create Policy.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em

seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-webapp-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Selecione Next (Próximo).

- f. Anexe a política de permissões à função de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise `codecatalyst-webapp-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

`codecatalyst-webapp-build-role`

- i. Em Descrição da função, insira:

`CodeCatalyst Web app build role`

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de confiança e uma política de permissões.

3. Anexe a política de permissões à função de criação, da seguinte forma:

- a. No painel de navegação, escolha Funções e, em seguida, pesquise por `codecatalyst-webapp-build-role`.
- b. Escolha `codecatalyst-webapp-build-role` exibir seus detalhes.
- c. Na guia Permissões, escolha Adicionar permissões e, em seguida, escolha Anexar políticas.
- d. Pesquise `codecatalyst-webapp-build-policy`, marque sua caixa de seleção e escolha Anexar políticas.

Agora você anexou a política de permissões à função de criação. A função de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN da função de construção, da seguinte forma:

- a. No painel de navegação, escolha Perfis.
- b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-webapp-build-role`).
- c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de criação com as permissões apropriadas e obteve seu ARN.

Criação manual de funções para o blueprint 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 as funções no IAM.

Para criar uma função de construção para CloudFormation

1. Crie uma política para a função, da seguinte forma:
 - a. Faça login em AWS.
 - b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
 - c. No painel de navegação, escolha Políticas.
 - d. Escolha Create Policy.
 - e. Escolha a guia JSON.
 - f. Exclua o código existente.
 - g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:*",
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"
```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-SAM-build-policy
```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    }  
  ]  
}
```

- e. Selecione Next (Próximo).
- f. Anexe a política de permissões à função de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise `codecatalyst-SAM-build-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

`codecatalyst-SAM-build-role`

- i. Em Descrição da função, insira:

`CodeCatalyst SAM build role`

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de confiança e uma política de permissões.

3. Anexe a política de permissões à função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Funções e, em seguida, pesquise por `codecatalyst-SAM-build-role`.
 - b. Escolha `codecatalyst-SAM-build-role` exibir seus detalhes.
 - c. Na guia Permissões, escolha Adicionar permissões e, em seguida, escolha Anexar políticas.
 - d. Pesquise `codecatalyst-SAM-build-policy`, marque sua caixa de seleção e escolha Anexar políticas.

Agora você anexou a política de permissões à função de criação. A função de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN da função de construção, da seguinte forma:
 - a. No painel de navegação, escolha Perfis.
 - b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (`codecatalyst-SAM-build-role`).

- c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de criação com as permissões apropriadas e obteve seu ARN.

Para criar uma função de implantação para o SAM

1. Crie uma política para a função, da seguinte forma:

- a. Faça login em AWS.
- b. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
- c. No painel de navegação, escolha Políticas.
- d. Escolha Create Policy.
- e. Escolha a guia JSON.
- f. Exclua o código existente.
- g. Cole o seguinte código:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```



```

    }
  ]
}

```

Note

Na primeira vez em que a função for usada para executar ações de fluxo de trabalho, use o caractere curinga na declaração de política de recursos e, em seguida, defina o escopo da política com o nome do recurso depois que ele estiver disponível.

```
"Resource": "*"

```

- h. Escolha Próximo: etiquetas.
- i. Selecione Next: Review (Próximo: revisar).
- j. Em Nome, insira:

```
codecatalyst-SAM-deploy-policy

```

- k. Escolha Criar política.

Agora você criou uma política de permissões.

2. Crie a função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Perfis e Criar perfil.
 - b. Escolha a política de confiança personalizada.
 - c. Exclua a política de confiança personalizada existente.
 - d. Adicione a seguinte política de confiança personalizada:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",

```

```
        "codecatalyst.amazonaws.com"
      ],
    },
    "Action": "sts:AssumeRole"
  }
]
```

- e. Selecione Next (Próximo).
- f. Anexe a política de permissões à função de criação. Na página Adicionar permissões, na seção Políticas de permissões, pesquise `codecatalyst-SAM-deploy-policy` e marque sua caixa de seleção.
- g. Selecione Next (Próximo).
- h. Em Nome da função, digite:

`codecatalyst-SAM-deploy-role`

- i. Em Descrição da função, insira:

`CodeCatalyst SAM deploy role`

- j. Selecione Criar função.

Agora você criou uma função de criação com uma política de confiança e uma política de permissões.

3. Anexe a política de permissões à função de criação, da seguinte forma:
 - a. No painel de navegação, escolha Funções e, em seguida, pesquise por `codecatalyst-SAM-deploy-role`.
 - b. Escolha `codecatalyst-SAM-deploy-role` exibir seus detalhes.
 - c. Na guia Permissões, escolha Adicionar permissões e, em seguida, escolha Anexar políticas.
 - d. Pesquise `codecatalyst-SAM-deploy-policy`, marque sua caixa de seleção e escolha Anexar políticas.

Agora você anexou a política de permissões à função de criação. A função de criação agora tem duas políticas: uma política de permissões e uma política de confiança.

4. Obtenha o ARN da função de construção, da seguinte forma:

- a. No painel de navegação, escolha Perfis.
- b. Na caixa de pesquisa, insira o nome da função que você acabou de criar (codecatalyst-SAM-deploy-role).
- c. Escolha a função na lista.

A página de resumo da função é exibida.

- d. Na parte superior, copie o valor do ARN.

Agora você criou a função de criação com as permissões apropriadas e obteve seu ARN.

Validação de conformidade para a Amazon CodeCatalyst

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

Note

Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte [Referência dos Serviços Qualificados pela HIPAA](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços com suporte e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência na 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 datacenter tradicionais.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura global AWS](#). Para saber mais sobre quais CodeCatalyst dados são replicados Regiões da AWS, consulte [Proteção de dados na Amazon CodeCatalyst](#).

Segurança de infraestrutura na Amazon CodeCatalyst

Como um serviço gerenciado, a Amazon CodeCatalyst é protegida pela segurança de rede AWS global. 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 de infraestrutura, consulte [Proteção de infraestrutura](#) em Pilar segurança: AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar CodeCatalyst pela rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Análise de configuração e vulnerabilidade na 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 [modelo de responsabilidade compartilhada da AWS](#).

Seus dados e privacidade na Amazon CodeCatalyst

A 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 AWS de Privacidade](#).

Para solicitar e visualizar seus dados, consulte [Solicitando seus dados](#) no Referência geral da AWS.

Excluindo seu perfil de AWS Builder ID

Excluir seu perfil é uma ação permanente que não pode ser revertida. O processo de exclusão começa imediatamente após você escolher Excluir. A Amazon CodeCatalyst começa a excluir seu perfil e todas as informações pessoais associadas. Esse processo pode levar até 90 dias para ser concluído.

Quando seu perfil é excluído, você não pode acessar ou recuperar seus dados na Amazon CodeCatalyst. Isso inclui tokens de acesso pessoal, funções, associações de usuários e quaisquer CodeCatalyst espaços da Amazon dos quais você seja o único membro. Você não pode mais fazer login na Amazon CodeCatalyst.

Para obter informações sobre como excluir seu perfil de AWS Builder ID, consulte [Excluindo seu AWS Builder ID](#) noReferência geral da AWS.

Melhores práticas para ações de fluxo de trabalho na Amazon CodeCatalyst

Há várias práticas recomendadas de segurança a serem consideradas ao desenvolver seus fluxos de trabalho. CodeCatalyst As diretrizes a seguir são gerais e não representam uma solução de segurança completa. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis em vez de requisitos.

Tópicos

- [Informações confidenciais](#)
- [Termos de licenciamento](#)
- [Código não confiável](#)
- [GitHub Ações](#)

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. CodeCatalyst Os segredos fornecem uma maneira fácil de armazenar e referenciar informações confidenciais de dentro do seu YAML.

Termos de 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 Ações

GitHub As ações são de código aberto, criadas e mantidas pela comunidade. Seguimos o [modelo de responsabilidade compartilhada](#) e consideramos o código-fonte GitHub das ações como dados do cliente pelos quais você é responsável. GitHub As ações podem ter acesso a segredos, tokens de repositório, código-fonte, links de contas e seu tempo de computação. Certifique-se de estar confiante na confiabilidade e na segurança das GitHub ações que planeja executar.

Orientações mais específicas e melhores práticas de segurança para GitHub ações:

- [Fortalecimento da segurança](#)
- [Evitando solicitações próprias](#)
- [Entrada não confiável](#)
- [Como confiar em seus blocos de construção](#)

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 obter 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, os principais de serviço listados no Principal elemento recebem permissões da política baseada em recursos, e o Condition bloco é usado para limitar o acesso ao recurso com escopo reduzido.

```
"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 o ID do espaço. O ARN usa o seguinte formato:

```
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 o máximo possível as permissões na política.

- Você pode usar o caractere curinga (*) na chave de `aws:SourceArn` condição para especificar todos os projetos no espaço com. `project/*`
- Você pode especificar permissões em nível de recurso na chave de `aws:SourceArn` condição para um projeto específico no espaço com. `project/projectId`

Princípios de serviço para CodeCatalyst

Você usa o `Principal` elemento em uma política JSON baseada em recursos para especificar o principal que tem acesso permitido ou negado a um recurso. Os principais que podem ser especificados na política de confiança incluem usuários, funções, contas e serviços. Você não pode usar o `Principal` elemento em uma política baseada em identidade; da mesma forma, você não pode identificar um grupo de usuários como principal em uma política (como uma política baseada em recursos) porque os grupos estão relacionados a permissões, não à autenticação, e os principais são entidades IAM autenticadas.

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. Os princípios do serviço são definidos 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

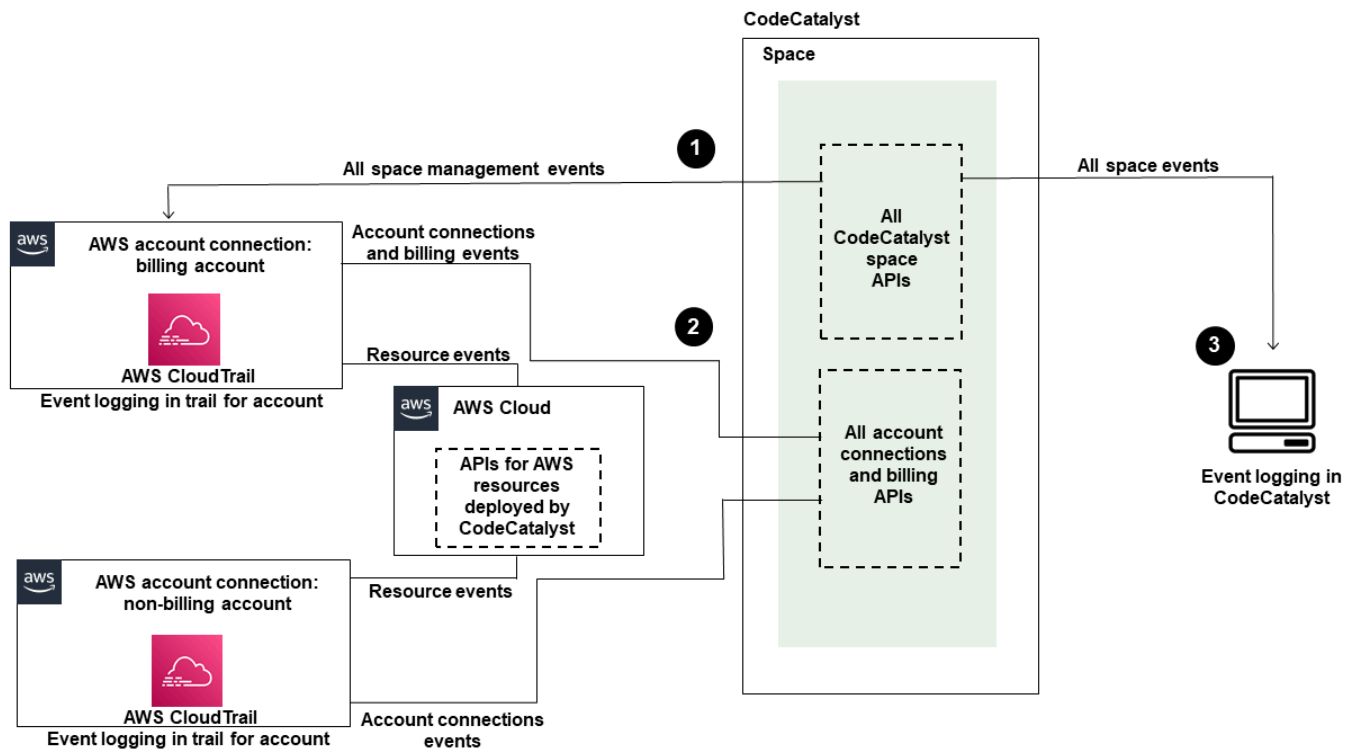
Para obter mais informações, consulte [Elementos de política JSON da AWS : entidade principal](#) no Guia do usuário do IAM.

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 o. Conta da AWS

O diagrama a seguir mostra como todos os eventos de gerenciamento do espaço são registrados na conta CloudTrail de cobrança, enquanto as conexões da conta/eventos de cobrança e os eventos de AWS recursos são registrados na respectiva conta. CloudTrail



O diagrama ilustra as seguintes etapas:

1. Quando um espaço é criado, um Conta da AWS é conectado 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 espaciais serão registrados na trilha e no bucket dessa conta. Para obter mais informações sobre eventos de CodeCatalyst gerenciamento registrados por CloudTrail, consulte [CodeCatalyst informações em 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 [Acessando eventos registrados 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 ter mais informações, consulte [Acessando eventos registrados 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 em log 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, [CodeCatalyst informações em 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 de todas as CodeCatalyst APIs no espaço usando o [list-event-logs](#) comando no AWS CLI.

Tópicos

- [Monitoramento de chamadas de API Contas da AWS usando o AWS CloudTrail registro](#)
- [Acessando eventos registrados usando o registro de eventos](#)

Monitoramento de chamadas de API Contas da AWS usando o AWS CloudTrail registro

CodeCatalyst A Amazon está integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou um AWS service (Serviço da AWS). CloudTrail captura chamadas de API feitas em nome de CodeCatalyst in connected Contas da AWS as events. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos em um bucket do S3, incluindo eventos para CodeCatalyst. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

CodeCatalyst suporta o registro das seguintes ações como eventos em arquivos de CloudTrail log:

- Os eventos de gerenciamento de CodeCatalyst espaços serão registrados na Conta da AWS conta de cobrança designada para o espaço. Para ter mais informações, consulte [CodeCatalyst eventos espaciais](#).

Note

Os eventos de dados para CodeCatalyst espaços podem ser acessados usando a CLI, conforme detalhado em. [Acessando eventos registrados usando o registro de eventos](#)

- Eventos para recursos usados em ações CodeCatalyst de fluxo de trabalho que ocorrem em um Conta da AWS ambiente conectado serão registrados como eventos nesse Conta da AWS. Para ter mais informações, consulte [CodeCatalyst conexões de conta e eventos de cobrança](#).

Important

Embora várias contas possam ser associadas a um espaço, o CloudTrail registro de eventos em CodeCatalyst espaços e projetos se aplica somente à conta de faturamento.

A conta de faturamento de espaço é sua Conta da AWS que é cobrada por CodeCatalyst recursos além do nível AWS gratuito. Várias contas podem ser conectadas a um espaço, enquanto somente uma conta pode ser a conta de cobrança designada. A conta de cobrança ou contas adicionais conectadas ao espaço podem ter funções do IAM que são usadas para implantar AWS recursos e infraestrutura, como um cluster do Amazon ECS ou um bucket do S3, a partir de fluxos de trabalho. CodeCatalyst Você pode usar o fluxo de trabalho YAML para identificar aquele em Conta da AWS que você implantou.

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. CloudTrail o registro deve ser configurado separadamente para cada Conta da AWS local em que os recursos são implantados.

CodeCatalyst o login em contas conectadas inclui as seguintes considerações:

- O acesso aos CloudTrail eventos é gerenciado com o IAM na conta conectada e não na CodeCatalyst.
- Conexões de terceiros, como a vinculação a um GitHub repositório, resultarão na gravação de nomes de recursos de terceiros nos CloudTrail registros.

Note

CloudTrail o registro de CodeCatalyst eventos está no nível do espaço e não isola os eventos pelos limites do projeto.

Para obter mais informações sobre CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

Note

Esta seção descreve o CloudTrail registro de todos os eventos registrados em um CodeCatalyst espaço conectado e aos Contas da AWS quais estão conectados. CodeCatalyst Além disso, para revisar todos os eventos registrados em um CodeCatalyst espaço, você também pode usar o AWS CLI `aws codecatalyst list-event-logs` comando e. Para ter mais informações, consulte [Acessando eventos registrados usando o registro de eventos](#).

CodeCatalyst eventos espaciais

As ações CodeCatalyst para gerenciar recursos no nível do espaço e no nível do projeto são registradas na conta de cobrança do espaço. Para CloudTrail registrar um CodeCatalyst espaço, os eventos são registrados com as seguintes considerações.

- CloudTrail os eventos se aplicam a todo o espaço e não têm como escopo nenhum projeto único.
- Quando você conecta um Conta da AWS a um CodeCatalyst espaço, os eventos registráveis para conexões da conta serão registrados nele. Conta da AWS Depois de habilitar essa conexão, você não poderá desativá-la.
- Quando você conecta um Conta da AWS a um CodeCatalyst espaço e o designa como a conta de cobrança do espaço, os eventos serão registrados nele. Conta da AWS Depois de habilitar essa conexão, você não poderá desativá-la.

Eventos para recursos em nível espacial e em nível de projeto são registrados somente na conta de faturamento. Para alterar a conta de CloudTrail destino, atualize a conta de cobrança em CodeCatalyst. No início do próximo ciclo de cobrança mensal, a alteração entra em vigor para a nova conta de cobrança em. CodeCatalyst Depois disso, a conta de CloudTrail destino é atualizada.

A seguir estão exemplos de eventos relacionados a ações CodeCatalyst para gerenciar recursos em nível espacial e em nível de projeto. AWS As seguintes APIs são lançadas por meio do SDK e da CLI. Os eventos serão registrados na conta de cobrança Conta da AWS especificada para o CodeCatalyst espaço.

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)
- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

CodeCatalyst conexões de conta e eventos de cobrança

Veja a seguir exemplos de eventos relacionados a ações em conexões CodeCatalyst de contas ou faturamento: AWS

- `AcceptConnection`
- `AssociateIAMRoletoConnection`
- `DeleteConnection`
- `DissassociateIAMRolefromConnection`
- `GetBillingAuthorization`

- `GetConnection`
- `GetPendingConnection`
- `ListConnections`
- `ListIAMRolesforConnection`
- `PutBillingAuthorization`
- `RejectConnection`

CodeCatalyst informações em CloudTrail

CloudTrail é ativado em e Conta da AWS quando você cria essa conta. Quando você conecta isso Conta da AWS a um CodeCatalyst espaço, os eventos desse espaço que ocorrem nele Conta da AWS são registrados são registrados nessa CloudTrail conta da AWS. Os eventos registráveis CodeCatalyst são registrados como CloudTrail eventos nos CloudTrail registros na conta conectada e no histórico de eventos no CloudTrail console, junto com outros AWS eventos registráveis nessa conta.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita por um usuário com o ID do AWS construtor.
- Se a solicitação foi feita com credenciais de usuário root ou 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 o elemento [CloudTrail userIdentity](#).

Acessando CloudTrail eventos

Para obter um registro contínuo dos eventos em sua Conta da AWS, incluindo eventos de CodeCatalyst atividades no Conta da AWS, crie uma trilha. Uma trilha permite CloudTrail entregar 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 AWS partição e entrega os arquivos de log ao bucket do S3 que você especificar. Além disso, você pode configurar

outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [CloudTrail serviços e integrações suportados](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [Recebendo arquivos de CloudTrail log de várias contas](#)

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

Exemplo de evento de conexões de CodeCatalyst conta em AWS

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListConnections` ação. Para um Conta da AWS que está conectado ao espaço, `ListConnections` é usado para visualizar todas as conexões da conta CodeCatalyst para isso Conta da AWS. O evento será registrado no local Conta da AWS especificado e o valor do `arn` será o Amazon Resource Name (ARN) da função usada para a ação. `accountId`

```
{
  "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
botocore/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 CodeCatalyst projeto em AWS

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateDevEnvironment` ação. Uma Conta da AWS que esteja conectada ao espaço e seja a conta de cobrança designada para o espaço é usada para eventos em nível de projeto no espaço, como a criação de um ambiente de desenvolvimento.

Abaixo `userIdentity`, no `accountId` campo, está o ID da conta do IAM Identity Center (432677196278) que hospeda o pool de identidades para todas as identidades de ID do AWS Builder. Esse ID da conta contém as seguintes informações sobre o CodeCatalyst usuário do evento.

- O `type` campo indica o tipo de entidade do IAM para a solicitação. Para CodeCatalyst eventos para recursos espaciais e de projetos, esse valor é `IdentityCenterUser`. O `accountId` campo especifica a conta proprietária da entidade que foi usada para obter as credenciais.
- O `userId` campo contém o identificador de ID do AWS construtor para o usuário.

- O `identityStoreArn` campo contém o ARN da função para a conta e o usuário do repositório de identidades.

O `recipientAccountId` campo contém o ID da conta de cobrança do espaço, com um valor de exemplo aqui de 111122223333.

Para obter mais informações, consulte o elemento [CloudTrail userIdentity](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IdentityCenterUser",
    "accountId": "432677196278",
    "onBehalfOf": {
      "userId": "user-ID",
      "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7"
    },
    "credentialId": "ABCDefGhiJKLMn11Lmn_1AbCDEFGHijk-AaBCdEFGHIjKlmnOPqrs11abEXAMPLE"
  },
  "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 certos eventos, o agente do usuário pode não ser conhecido. Nesse caso, CodeCatalyst fornecerá um valor de Unknown no `userAgent` campo no CloudTrail evento.

Consultando suas trilhas de CodeCatalyst eventos

Você pode criar e gerenciar consultas para seus CloudTrail registros usando uma tabela de consultas no Amazon Athena. Para obter mais informações sobre a criação de uma consulta, consulte Como [consultar AWS CloudTrail registros no Guia](#) do usuário do Amazon Athena.

Acessando eventos registrados usando o registro de eventos

Quando os usuários realizam ações na Amazon CodeCatalyst, essas ações são registradas como eventos. Você pode usar o AWS CLI para visualizar registros de eventos em um espaço em um período de tempo 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 CodeCatalyst espaço são registrados na CloudTrail conta de cobrança conectada. Para obter mais informações sobre eventos de CodeCatalyst gerenciamento registrados por CloudTrail, consulte [CodeCatalyst informações em CloudTrail](#).

Para visualizar um registro de eventos de um espaço, você deve ter instalado e configurado o AWS CLI com um perfil para CodeCatalyst e ter a função de administrador do espaço. Para obter

mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#) e [Função de administrador de espaço](#).

Note

Para visualizar o registro de eventos que ocorrem em nome de CodeCatalyst in connected Contas da AWS ou para visualizar o registro de eventos de espaço ou recursos do projeto na conta de faturamento conectada, você pode usar AWS CloudTrail. Para ter mais informações, consulte [Monitoramento de chamadas de API Contas da AWS usando o AWS CloudTrail registro](#).

1. Abra um terminal ou linha de comando e execute o `aws codecatalyst list-event-logs` comando, especificando:
 - O nome do espaço com a `--space-name` opção.
 - 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 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 `--max-results` opção. Se o número de resultados for maior que o número especificado, a resposta incluirá um `nextToken` elemento 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 `--event-name` opção.

Este exemplo retorna eventos registrados no espaço nomeado *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 os eventos ocorrerem nesse período, o comando retornará resultados semelhantes aos seguintes:

```

{
  "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-EXAMPLEEaaaaa",
      "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 `list-event-logs` comando novamente com a `--next-token` opção 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 CodeCatalyst](#).

| Recurso | Informações |
|---|--|
| Aliases em CodeCatalyst | Qualquer combinação de caracteres permitido s entre 3 e 100 caracteres e deve começar com uma letra. Caracteres válidos: A-Z, a-z e 0-9. Os aliases não podem: <ul style="list-style-type: none"> • contêm menos de 3 caracteres • conter espaços ou qualquer um dos seguintes caracteres: ? ^ * [\ ~ : |
| Número máximo de convites enviados por um usuário por dia | 500 |

| Recurso | Informações |
|---|--|
| 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 |
| 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 de caracteres permitidos entre 8 e 64 caracteres. Caracteres válidos: A-Z, a-z e 0-9. Sua senha pode incluir os seguintes caracteres não alfanuméricos: (~ ! @ # \$ % ^ & * _ - + = ` \ { } [] : ; " ' < > , . ? /) |
| Nomes PAT em CodeCatalyst | Qualquer combinação de 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 a verificação do endereço de e-mail expirar | 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 ao se inscrever

Você pode encontrar alguns problemas ao se inscrever. Temos algumas soluções.

Meu endereço de e-mail já está em uso

Se o e-mail que você inseriu já estiver em uso e você o reconhecer como seu, talvez você já tenha um perfil conosco. Faça login com essa identidade existente. Se você não possui 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 seu 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 sua caixa de entrada. Verifique sua pasta de spam, lixo eletrônico e itens excluídos novamente.
3. Se você ainda não vê seu e-mail de verificação, escolha Reenviar código. Se você já saiu dessa página, reinicie seu fluxo de trabalho para se inscrever [na Amazon CodeCatalyst](#).

Minha senha não atende aos requisitos mínimos

Para sua segurança, sua senha deve incluir de 8 a 20 caracteres, letras maiúsculas e minúsculas e números.

Problemas ao fazer login

Esqueci minha senha

Siga as etapas em [Esqueci minha senha](#).

Minha senha não está funcionando

Você deve seguir esses 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.
- Você não pode reutilizar as últimas três 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

Se você achar que não pode adicionar outro dispositivo de MFA, ele pode ter atingido o limite de dispositivos de MFA que você pode registrar. 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 [Excluindo 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 finalizar a sessão

Não consigo encontrar onde sair

No canto superior direito da página, escolha 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 função não existe para um fluxo de trabalho com falha

Problema: depois de criar um projeto a partir do aplicativo web ou do blueprint sem servidor, o fluxo de trabalho falha com o seguinte erro:

CLIENT_ERROR: A função não existe

Solução possível: depois de configurar uma função do IAM com as permissões para executar seu fluxo de trabalho e adicionar a função do IAM ao YAML do seu fluxo de trabalho, o fluxo de trabalho ainda falhará porque talvez seja necessário adicionar a função do IAM à conexão da sua conta. Adicione a função do IAM à conexão da conta do seu espaço, conforme detalhado em [Adicionar funções do IAM às conexões da conta](#).

Eu recebo um erro de função em um fluxo de trabalho com falha

Problema: depois de criar um projeto a partir do aplicativo web ou do blueprint sem servidor, o fluxo de trabalho falha com o seguinte erro:

CLIENT_ERROR: A função não está configurada 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 Conta da AWS conexão ativa, crie e adicione uma função do IAM com permissões para executar ações de fluxo de trabalho. Adicione a função do IAM à conexão da sua conta, conforme detalhado em [Adicionar funções 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 [Adicionar a conexão da conta e as funções do IAM ao seu ambiente de implantação](#).

Preciso atualizar a função do IAM em um 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. Escolha o botão YAML.
2. Selecione a opção Editar.

3. No `ActionRoleArn`: campo, substitua o ARN da função do IAM pelo ARN da função do IAM atualizado. Escolha Validar.
4. Escolha Commit (Confirmar).

O fluxo de trabalho é iniciado automaticamente se estiver na filial principal. Caso contrário, para executar novamente o fluxo de trabalho, escolha 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, escolha Configurações e, em seguida, escolha Aplicativos.
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 escolha Verificar.
4. Aceite as novas permissões, indique os repositórios nos quais você deseja aplicar as permissões e escolha 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](#)
- [Instalando uma extensão em um espaço](#)
- [Desinstalando uma extensão em um espaço](#)
- [Conectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst](#)
- [Desconectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst](#)
- [Vinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#)
- [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#)
- [Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst](#)
- [Iniciando automaticamente a execução de um fluxo de trabalho após eventos de repositórios de terceiros](#)
- [Restringindo o acesso ao IP com o GitHub Enterprise Cloud](#)
- [O bloqueio de pull requests de terceiros é mesclado 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 ter mais informações, consulte [Criação de 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 aplicar 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 status CodeCatalyst de execução do fluxo de trabalho para GitHub repositórios vinculados e bloqueie 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 seu código. A extensão de repositórios Bitbucket permite que você use repositórios Bitbucket vinculados em projetos da Amazon. CodeCatalyst Você também pode vincular repositórios do Bitbucket ao criar um novo CodeCatalyst projeto. Para ter mais informações, consulte [Criação de 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
- Armazene e gerencie arquivos de definição de fluxo de trabalho em seus 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
- Crie um repositório Bitbucket com código gerado por um blueprint ao criar um projeto com um blueprint ou aplicar um blueprint
- 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 problemas do Jira no CodeCatalyst

O Jira é um aplicativo de software que ajuda as equipes de desenvolvimento ágil a planejar, atribuir, monitorar, relatar 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.

Conectando e vinculando

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, em seguida, vincule os repositórios do Bitbucket.
- Software Jira: conecte o site do Jira e, em seguida, vincule os projetos do Jira.

Início rápido: instalação de extensões, conexão de provedores e vinculação de recursos no CodeCatalyst

Este tutorial fornece uma explicação passo a passo das três tarefas a seguir:

1. Instale os GitHub repositórios, os repositórios do Bitbucket 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 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 ou site do Jira a. CodeCatalyst

Important

Para conectar sua GitHub conta, o espaço de trabalho do Bitbucket ou o 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 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.

ℹ 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](#).

3. Vincule seu GitHub repositório, repositório Bitbucket ou projeto Jira ao seu projeto. CodeCatalyst

⚠ 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e 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.

ℹ Note

- Um repositório GitHub ou Bitbucket só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados GitHub ou do Bitbucket com projetos. CodeCatalyst

- Você não pode vincular um repositório GitHub ou um repositório do Bitbucket 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.
- 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 ou a extensão de repositórios do Bitbucket, conectar-se à sua GitHub conta ou ao espaço de trabalho do Bitbucket e vincular repositórios de terceiros ao criar um novo projeto. CodeCatalyst Para ter mais informações, consulte [Criação de 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 e repositórios Bitbucket permitem que você use GitHub nossos repositórios Bitbucket em. 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 ou Jira Software. Você também pode filtrar extensões com base em categorias.

4. (Opcional) Para ver mais detalhes sobre a extensão, como as permissões que a extensão 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 que você instalou, 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 ou a extensão do Jira Software, a próxima etapa é conectar sua GitHub conta, o espaço de trabalho do Bitbucket ou o site do Jira ao seu espaço. CodeCatalyst

Para conectar sua GitHub conta, espaço de trabalho do Bitbucket ou site do Jira ao CodeCatalyst

- Siga um destes procedimentos, 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 um GitHub espaço, ou a página de configuração do CodeCatalyst aplicativo Amazon se você pertencer a apenas um GitHub espaço. Configure o aplicativo para o acesso ao repositório que você deseja permitir e escolha **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-se a um espaço de trabalho do Bitbucket.
 1. Na guia Espaços de trabalho conectados do Bitbucket, escolha 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.
 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.
 - a. Se for a primeira vez que você concede CodeCatalyst acesso a um espaço de trabalho, escolha Ir para configurações para ativar o modo de desenvolvimento do Bitbucket.
 - b. Na página Aplicativos instalados do seu espaço de trabalho, escolha a caixa de seleção Ativar modo de desenvolvimento para permitir a instalação de e CodeCatalyst, em seguida, conecte-se novamente a partir de. CodeCatalyst

Depois de permitir a instalação do CodeCatalyst, ele terá acesso a todos os seus repositórios no espaço de trabalho do Bitbucket.


 Tip

Se você já conectou um espaço de trabalho do Bitbucket 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 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 o aplicativo para o acesso ao espaço de trabalho que você deseja permitir e escolha 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.

- Software Jira: Conecte um site do Jira.
 1. Na guia Sites conectados do Jira, escolha 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. Escolha Começar para ir para a etapa final.

3. Dependendo da sua função, 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 obter mais informações sobre como instalar aplicativos do Jira, consulte [Quem pode instalar aplicativos?](#) .


- 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 realizadas 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 repositórios GitHub ou o Bitbucket 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 ou projeto do Jira a um CodeCatalyst projeto na página de detalhes da extensão

- Siga um destes procedimentos, 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 em cinza, 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, escolha 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 o GitHub repositório em 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

- Repositórios do Bitbucket: vincule um repositório do Bitbucket.
 1. Na guia Repositórios vinculados do Bitbucket, escolha Vincular repositório do Bitbucket.
 2. No menu suspenso do espaço de trabalho do Bitbucket, escolha 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 em cinza, 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 o repositório 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

- Jira Software: vincule um projeto do Jira.
 1. Na guia Projetos vinculados do Jira, escolha Vincular projeto do Jira.
 2. No menu suspenso do 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Você também pode vincular seu repositório GitHub ou o do Bitbucket a um projeto a partir dos repositórios de origem no Code. Para ter mais informações, consulte [Vinculando recursos de fornecedores terceirizados conectados](#).

Próximas etapas

Depois de instalar os GitHub repositórios ou a extensão de repositórios do Bitbucket, conectar seu provedor de recursos e vincular seus repositórios de terceiros aos seus CodeCatalyst projetos, você pode usá-lo em CodeCatalyst fluxos de trabalho e ambientes de desenvolvimento. Você também pode criar repositórios de terceiros na GitHub conta conectada ou no espaço de trabalho do Bitbucket com código gerado a partir de um blueprint. Para obter mais informações, consulte [Iniciando automaticamente a 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 obter mais informações sobre como vincular pull requests 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](#).

Instalando 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 a função de administrador do Space no espaço.

⚠ 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.

Para instalar uma extensão do CodeCatalyst catálogo

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até o seu CodeCatalyst espaço.
3. Navegue até o CodeCatalyst catálogo escolhendo o ícone Catálogo



no menu superior. Você pode pesquisar extensões ou filtrar extensões com base em categorias.

4. (Opcional) Escolha o nome da extensão para ver mais detalhes sobre a extensão, como as permissões que a extensão 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. Navegue pelas guias para obter mais informações sobre a extensão. A página de detalhes também é onde você executará configurações adicionais da extensão, se necessário.


Desinstalando 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 a função de administrador do Space 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. Siga um destes procedimentos para ver uma lista das extensões instaladas em seu espaço:
 - a. Escolha Configurações e, em seguida, escolha Extensões instaladas.
 - b. Escolha o ícone Catálogo  no menu superior.
4. Escolha Configurar na extensão que você deseja desinstalar.
5. Escolha Desinstalar na página de detalhes da extensão.
6. Revise as informações na caixa de diálogo Desinstalar extensão. Siga as instruções e escolha Desinstalar para desinstalar a extensão.

Conectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst

Para usar um repositório GitHub ou Bitbucket 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, o espaço de trabalho do Bitbucket ou o 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 ou site do Jira ao CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Siga um destes procedimentos para ver uma lista das extensões instaladas em seu espaço:
 - a. Escolha Configurações e, em seguida, escolha 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 ou Jira Software.
5. Siga um destes procedimentos, dependendo da extensão de terceiros que você escolheu 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 o aplicativo para o acesso ao repositório que você deseja permitir e escolha 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-se a um espaço de trabalho do Bitbucket.
 1. Na guia Espaços de trabalho conectados do Bitbucket, escolha 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.
 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.
 - a. Se for a primeira vez que você concede CodeCatalyst acesso a um espaço de trabalho, escolha Ir para configurações para ativar o modo de desenvolvimento do Bitbucket.
 - b. Na página Aplicativos instalados do seu espaço de trabalho, escolha a caixa de seleção Ativar modo de desenvolvimento para permitir a instalação de e CodeCatalyst, em seguida, conecte-se novamente a partir de. CodeCatalyst

Depois de permitir a instalação do CodeCatalyst, ele terá acesso a todos os seus repositórios no espaço de trabalho do Bitbucket.


 Tip

Se você já conectou um espaço de trabalho do Bitbucket 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 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 o aplicativo para o acesso ao espaço

de trabalho que você deseja permitir e escolha 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.

- Software Jira: Conecte um site do Jira.
 1. Na guia Sites conectados do Jira, escolha 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. Escolha Começar para ir para a etapa final.

3. Dependendo da sua função, 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 obter mais informações sobre como instalar aplicativos do Jira, consulte [Quem pode instalar aplicativos?](#) .
 - 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 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 problemas do Jira no CodeCatalyst, você pode desconectar sua fonte terceirizada. Quando uma GitHub conta ou espaço de trabalho do Bitbucket 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 ter mais informações, consulte [Desconectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst](#).

Desconectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst

Se você não quiser mais usar repositórios, GitHub repositórios do Bitbucket ou problemas do Jira CodeCatalyst, você pode desconectar sua fonte terceirizada. Depois que uma GitHub conta ou espaço de trabalho do Bitbucket 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, primeiro você deve desvincular todos os repositórios vinculados do Bitbucket desse espaço de trabalho.
- Para desconectar um site do Jira, primeiro você deve desvincular todos os projetos vinculados do Jira dessa conta.

Para ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Para desconectar um GitHub projeto, espaço de trabalho do Bitbucket ou site do Jira

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.
3. Siga um destes procedimentos para ver uma lista das extensões instaladas em seu espaço:
 - a. Escolha Configurações e, em seguida, escolha 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 ou Jira Software.
5. Siga um destes procedimentos, dependendo da extensão de terceiros que você escolheu 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-se de um espaço de trabalho do Bitbucket.

Na guia Espaços de trabalho conectados do Bitbucket, escolha o espaço de trabalho do Bitbucket que você deseja desconectar e, em seguida, escolha Desconectar o espaço de trabalho do Bitbucket.

- Software Jira: desconecte-se de um site Jira.

Na guia Sites conectados do Jira, escolha o site do Jira que você deseja desconectar e, em seguida, escolha 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 escolha Desconectar.

Vinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst

Antes de usar um repositório GitHub ou do Bitbucket ou gerenciar um projeto do Jira, você deve conectar a fonte terceirizada à qual o repositório ou projeto pertence ao seu espaço. CodeCatalyst Para ter mais informações, consulte [Conectando GitHub contas, espaços de trabalho do Bitbucket e sites do Jira CodeCatalyst](#).

Você pode usar repositórios vinculados GitHub ou do Bitbucket 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 do Bitbucket a um CodeCatalyst projeto a partir da página de detalhes dos repositórios ou da extensão dos GitHub repositórios do Bitbucket, ou da visualização de repositórios de origem em Código no próprio projeto.

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

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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Important

Para vincular seu projeto do Jira ao seu CodeCatalyst projeto, você deve ser administrador do CodeCatalyst Space ou administrador CodeCatalyst do projeto.

⚠ 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.

ℹ Note

- Um repositório GitHub ou Bitbucket só pode ser vinculado a um CodeCatalyst projeto em um espaço.
- Você não pode usar repositórios vazios ou arquivados GitHub ou do Bitbucket com projetos. CodeCatalyst
- Você não pode vincular um repositório GitHub ou um repositório do Bitbucket 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.
- Um CodeCatalyst projeto só pode ser vinculado a um projeto do Jira. Um projeto do Jira pode ser vinculado a vários CodeCatalyst projetos.



Tópicos

- [Vinculando recursos de fornecedores terceirizados conectados](#)
- [Vinculando um repositório de terceiros durante CodeCatalyst a criação do projeto](#)

Vinculando recursos de fornecedores terceirizados conectados

Para vincular um GitHub repositório, repositório Bitbucket ou 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. Siga um destes procedimentos para ver uma lista das extensões instaladas em seu espaço espacial:
 - a. Escolha Configurações e, em seguida, escolha Extensões instaladas.
 - b. Escolha o ícone Catálogo  no menu superior.
 4. Escolha Configurar para uma das seguintes extensões: GitHub repositórios, repositórios Bitbucket ou Jira Software.
 5. Siga um destes procedimentos, dependendo da extensão de terceiros que você escolheu 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 em cinza, 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, escolha 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 o GitHub repositório em 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

- Repositórios do Bitbucket: vincule um repositório do Bitbucket.
 1. Na guia Repositórios vinculados do Bitbucket, escolha Vincular repositório do Bitbucket.
 2. No menu suspenso do espaço de trabalho do Bitbucket, escolha 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 em cinza, 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 o repositório 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

- Jira Software: vincule um projeto do Jira.
 1. Na guia Projetos vinculados do Jira, escolha Vincular projeto do Jira.
 2. No menu suspenso do 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Para vincular um repositório GitHub ou Bitbucket a um CodeCatalyst projeto a partir da página de repositórios de origem em um projeto

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.
4. Escolha Adicionar repositório e, em seguida, escolha Vincular repositório.

5. No menu suspenso Provedor de repositório, escolha um dos seguintes provedores de repositório terceirizados: ou Bitbucket. GitHub
6. Siga um destes procedimentos, dependendo do provedor de repositório terceirizado que você escolheu 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, escolha 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 do espaço de trabalho do Bitbucket, escolha 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

7. Escolha Vincular.

Se você não quiser mais usar um repositório GitHub ou 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 ter mais informações, consulte [Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst](#).

Depois de vincular seu repositório GitHub ou o do Bitbucket ao seu 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, blueprints e muito mais. Para obter mais informações, consulte [Iniciando automaticamente a 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 obter mais informações sobre como vincular pull requests 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 repositório GitHub ou um repositório do Bitbucket a um novo CodeCatalyst projeto ao criar o novo CodeCatalyst projeto. Para ter mais informações, consulte [Criação de um projeto com um repositório de terceiros vinculado](#).

Desvinculando GitHub repositórios, repositórios do Bitbucket e projetos do Jira no CodeCatalyst

Se você não quiser mais usar um GitHub repositório ou repositório do Bitbucket ou gerenciar um projeto do Jira no CodeCatalyst, você pode desvincular o repositório ou o projeto do seu projeto. CodeCatalyst

A desvinculação de um repositório GitHub ou do Bitbucket não exclui o repositório nem faz alterações nele. Ele não exclui nenhum arquivo de configuração do fluxo de trabalho armazenado nesse repositório vinculado. No entanto, depois de desvincular um repositório GitHub ou do Bitbucket, os eventos nesse 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 repositório do Bitbucket a um CodeCatalyst projeto a partir da página de detalhes dos repositórios GitHub ou da extensão dos GitHub repositórios do Bitbucket, ou da visualização de repositórios de origem em 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 ou repositório Bitbucket do seu CodeCatalyst projeto, você deve ser o administrador do Space ou o administrador do projeto.

Para desvincular um GitHub repositório, repositório Bitbucket 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 o seu CodeCatalyst espaço.
3. Siga um destes procedimentos para ver uma lista das extensões instaladas em seu espaço:
 - a. Escolha Configurações e, em seguida, escolha 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 ou Jira Software.
5. Siga um destes procedimentos, dependendo da extensão de terceiros que você escolheu 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, em seguida, escolha Desvincular repositório do Bitbucket.

- Jira Software: desvincule um projeto do Jira.

Na guia Projetos do Jira, escolha o projeto do Jira que você deseja desvincular e, em seguida, escolha 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 escolha Desvincular.

Para desvincular um repositório GitHub ou Bitbucket em um 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, em seguida, selecione Repositórios de origem.
4. Escolha o botão de opção do repositório que você deseja desvincular e, em seguida, escolha Desvincular repositório.
5. Revise as informações na caixa de diálogo. Siga as instruções e escolha Desvincular para desvincular o repositório.

Visualizando repositórios de terceiros e pesquisando problemas do Jira no CodeCatalyst

Depois de vincular GitHub nossos repositórios do Bitbucket, 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 ou do Bitbucket na lista de repositórios de origem do seu projeto ou na página de detalhes dos repositórios ou da extensão dos GitHub repositórios do Bitbucket. Escolhê-los na lista de repositórios não os abre em CodeCatalyst. Em vez disso, eles abrem no provedor de repositórios terceirizado, onde você pode visualizar e trabalhar no código no repositório vinculado.

Para visualizar repositórios vinculados GitHub ou do Bitbucket em CodeCatalyst

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst projeto.
3. No painel de navegação, selecione Código e, em seguida, selecione Repositórios de origem.

Para visualizar repositórios vinculados GitHub ou do Bitbucket 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 você deseja visualizar, faça o seguinte:
 - 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

Os repositórios GitHub ou repositórios do Bitbucket vinculados ao seu CodeCatalyst projeto são mostrados na lista. Escolha o repositório GitHub ou o Bitbucket para visualizar e editar arquivos no provedor de repositórios terceirizado.

Note

Se um fluxo de trabalho usar um repositório GitHub ou Bitbucket 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](#)

Para pesquisar 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 pull request.

Iniciando automaticamente a execução de um fluxo de trabalho após eventos de repositórios de terceiros

Você pode usar um repositório vinculado GitHub ou do Bitbucket como fonte para um fluxo de trabalho, onde as alterações em uma ramificação especificada em um repositório vinculado GitHub ou do Bitbucket iniciam automaticamente a execução do fluxo de trabalho.

Um fluxo de trabalho é um procedimento automatizado que descreve como criar, testar e implantar seu 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 executadas 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 você pode usar fluxos de trabalho em um projeto, [crie um projeto com um blueprint](#). Cada blueprint implanta um fluxo de trabalho funcional que você pode revisar, executar e experimentar.

Quando você configura um fluxo de trabalho para usar um repositório vinculado GitHub ou do Bitbucket, o arquivo de configuração do fluxo de trabalho é armazenado nesse repositório GitHub ou no Bitbucket. A configuração do fluxo de trabalho é um arquivo YAML que define o nome, os acionadores, os recursos, os artefatos e as ações do fluxo de trabalho. Para obter mais informações sobre o arquivo de configuração do fluxo de trabalho, consulte [Definição de fluxo de trabalho YAML](#).

O arquivo de configuração do fluxo de trabalho deve estar no `./codecatalyst/workflows/` diretório em seu repositório GitHub ou no Bitbucket.

Você pode usar o editor de fluxo de trabalho para criar e configurar fluxos de trabalho. Para obter mais informações, consulte [Introdução aos fluxos de trabalho](#) e [Conectando um fluxo de trabalho a um repositório de origem](#).

Adicionar 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 `Triggers` seção 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 pull request

O exemplo a seguir mostra um gatilho que inicia a execução de um fluxo de trabalho sempre que uma pull request é criada em qualquer ramificação no seu repositório de origem.

Triggers:

```
- Type: PULLREQUEST
```

Events:

```
- OPEN
```

Para ter mais informações, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).

Restringindo o acesso ao IP com o GitHub Enterprise Cloud

Você pode restringir o acesso aos seus repositórios GitHub ou aos repositórios do Bitbucket com base em endereços IP definindo regras ou configurações. Você pode fazer isso por meio das configurações do provedor terceirizado ou dos recursos de controle de acesso.

Dependendo do provedor de repositório terceirizado que você está usando, consulte um dos seguintes:

- 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.

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 ter 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 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
```

O bloqueio de pull requests de terceiros é mesclado quando os fluxos de trabalho falham

Depois de vincular um GitHub repositório a CodeCatalyst, você pode adicionar CodeCatalyst fluxos de trabalho para pull requests. Uma ou mais execuções de fluxo de trabalho podem ocorrer em uma confirmação específica, e o status de execução de cada fluxo de trabalho também CodeCatalyst é refletido como parte do status de confirmação no GitHub Bitbucket. Quando uma nova confirmação é enviada, os novos [status de execução](#) do fluxo de trabalho são refletidos nessa GitHub 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 GitHub para bloquear uma mesclagem de pull request quando o último commit 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. Para obter mais informações, consulte GitHub a documentação [Sobre verificações de status](#) e [Sobre filiais protegidas](#). Para saber mais sobre fluxos de trabalho, consulte [Executar um fluxo de trabalho](#) e [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#)

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 pull request. Como resultado, eventos de pull request, eventos de fluxo de trabalho e eventos de implantação são enviados para o Jira e adicionados à edição do Jira. As pull requests 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 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, escolha Código e, em seguida, escolha Pull requests.
4. Escolha Criar pull request para inserir os detalhes da pull request.
5. No menu suspenso Repositório de origem, escolha o repositório de origem ao qual você deseja vincular uma pull request.

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 do título da pull request, insira o título das suas pull requests.
9. Escolha Vincular problemas para o campo opcional de problemas do Jira, 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 pull request.
11. Escolha Criar para criar a pull request.

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 pull request vinculada e todos os eventos do fluxo de trabalho relacionados à pull request aparecerão na edição do Jira ao visualizá-la 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 Os eventos de CI/CD 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 a edição 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, escolha o problema do Jira que você deseja visualizar no seu projeto do Jira.
5. No painel Detalhes do projeto Jira, escolha a pull request listada para Development para ver os detalhes da pull request.
6. (Opcional) Para ver as versões mais recentes, escolha a guia Compilações.
7. (Opcional) Para ver o status de desenvolvimento, escolha a guia Implantações.

Pesquise código, problemas, projetos e usuários em CodeCatalyst

Use a barra de pesquisa ou uma janela de resultados de pesquisa dedicada CodeCatalyst para pesquisar códigos, problemas, projetos e usuários 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

- [Refinando 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 o idioma CodeCatalyst da consulta de pesquisa. Para ter mais informações, consulte [Refinando sua consulta de pesquisa](#).
3. Execute um destes procedimentos:
 - Para pesquisar recursos no projeto em que você está atualmente, escolha Este projeto.
 - Para pesquisar recursos em todos os projetos no espaço em que você está atualmente, escolha Este espaço.
4. Visualize os resultados da pesquisa em uma janela dedicada aos resultados da pesquisa fazendo o seguinte:
 - Na parte inferior da janela de resultados da pesquisa rápida, escolha Exibir todos os resultados em nome do projeto | nome do espaço para ver todos os resultados da pesquisa.
 - Pressione Enter para ver todos os resultados da pesquisa.

i Tip

Mencione outros usuários do projeto em um comentário ou descrição de pull request, 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.

Refinando sua consulta de pesquisa

Se você não conseguir encontrar o que está procurando depois de pesquisar, você pode refinar sua pesquisa com a linguagem CodeCatalyst de consulta especializada. Os campos individuais não têm limite de caracteres, mas a consulta geral tem um limite de 1.024 caracteres.

Tópicos

- [Refinação por tipo](#)
- [Refinação por campo](#)
- [Refinando com operadores booleanos](#)
- [Refinação por projeto](#)

Refinação por tipo

Para refinar o escopo de 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 ter mais informações, consulte [Campos de código](#).

- `type:issue AND Bug`— Mostre os resultados do problema em campos relacionados ao problema que contêm “Bug”.

Para ter mais informações, consulte [Campos de problema](#).

- `type:user AND MaryMajor`— Mostre os resultados do usuário em campos relacionados ao usuário que contêm “MaryMajor”.

Para ter mais informações, consulte [Campos de usuário](#).

- `type:project AND Datafeeder`— Mostre os resultados do projeto que contêm “Datafeeder”.

Para ter mais informações, consulte [Campos do projeto](#).

Refinação 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,,, usernameproject`, e assim por diante `description`, e a *consulta* é o texto que você está pesquisando. Para obter uma lista de campos, consulte [Referência de campos pesquisáveis](#). Você pode pesquisar várias consultas usando parênteses.

Exemplos:

- `title:bug`— Mostrar resultados em que o título contém “bug”.
- `username:John`— Mostrar resultados em que o nome do usuário contém “John”.
- `project:DataFeeder`— Mostrar resultados no projeto “DataFeeder”. A consulta não diferencia maiúsculas de minúsculas.
- `description:overview`— Mostrar resultados onde a descrição contém “visão geral”.

Refinando 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, CodeCatalyst as juntará `OR` por padrão. Você pode agrupar frases de pesquisa usando parênteses.

- `exception AND type:code`— Mostrar apenas os resultados do código para “exceção”.
- `path:README.md AND repo:ServerlessAPI`— Mostre resultados para caminhos com “README.md” em que o repositório é denominado “ServerlessAPI”.
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)`— Mostrar resultados para “buildspec.yml” em que o repositório é “ServerlessAPI” ou “ServerlessWebApp”.
- `path:java NOT (path:py OR path:ts)`— Mostra resultados em que o caminho contém “java”, mas não “py” ou “ts”.

Refinação por projeto

Para refinar o escopo de 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*—Mostrar 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.

Escapando caracteres especiais - 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 escapar de um caractere, adicione uma barra invertida (\) na frente dele. Por exemplo, a consulta de pesquisa [Feature] deve ser Feature ou \ [Feature].

Limitando a pesquisa — a pesquisa não diferencia maiúsculas de minúsculas. Pesquisar em letras minúsculas evita que suas consultas dividam palavras na mudança de maiúsculas e minúsculas. Por exemplo, para consultar por MyService e somenteMyService, considere consultar myservice para evitar resultados que contenham somente my ouservice.

A pesquisa une palavras e partes de palavras com a conjunção OR-wise por padrão. Por exemplo, new function poderia retornar resultados contendo ambos new function e também resultados com apenas new oufunction. Para evitar o último, combine várias palavras comAND. Por exemplo, você pode pesquisarnew 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 commits, considere [clonar o repositório localmente](#), [abrir a ramificação em um ambiente de desenvolvimento](#) ou [visualizar as ramificações e os detalhes na](#) interface do usuário. CodeCatalyst A alteração da ramificação padrão resulta em atualizações nos arquivos que podem

ser descobertos pela pesquisa. Para ter mais informações, consulte [Gerenciando a ramificação padrão de um repositório](#).

Referência de campos pesquisáveis

CodeCatalyst pesquisa os seguintes campos 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 |
|-------------------------|-------|---|
| Nome da filial | ramo | Nome da filial 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. |
| ID de confirmação | 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 do commit na ponta do nome da ramificação especificado em. <code>branchName</code> |
| Mensagem de confirmação | 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 |

| Campo | Alias | Descrição |
|----------------------------|---------|--|
| | | de confirmação for fornecida , esse valor será uma string vazia. |
| filePath | caminho | Caminho do arquivo desse arquivo de código. |
| lastUpdatedBy | N/D | CodeCatalyst usuário 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. |
| lastUpdatedByIdentificação | N/D | ID exclusiva gerada pelo sistema do usuário que atualizou o arquivo de código pela última vez. Se a ID do usuário não estiver disponível, esse valor pode ser o endereço de e-mail do usuário. |
| lastUpdatedTime | N/D | Hora em que os dados da pesquisa foram atualizados pela última vez com o commit que continha o arquivo de código (em horário universal coordenado (UTC)). |
| projectId | N/D | ID exclusiva do projeto gerada pelo sistema. |

| Campo | Alias | Descrição |
|-------------------|---------------------------|---|
| projectName | Nomes do projeto, projeto | Nome de exibição do projeto que contém o repositório de origem em que o arquivo de código foi confirmado. |
| ID do repositório | RepoID | ID exclusiva gerada pelo sistema do repositório de origem. |
| repositoryName | repositório, repositório | Nome de exibição do repositório de origem em que o arquivo de código foi confirmado. |

Campos de problema

| Campo | Alias | Descrição |
|-----------------------|--------------------|--|
| IDs dos destinatários | ID do destinatário | IDs exclusivos gerados pelo sistema dos usuários atribuídos ao problema. |
| cessionários | destinatário | Nomes de usuário dos usuários atribuídos ao problema. |
| Criado por | N/D | Nome de exibição do usuário que criou o problema. |
| createdById | N/D | ID exclusiva gerada pelo sistema do usuário que criou o problema. |
| Hora de criação | N/D | Hora em que o problema foi criado (em horário universal coordenado (UTC)). |

| Campo | Alias | Descrição |
|----------------------------|---------------------------|--|
| description | N/D | Descrição do problema. |
| É arquivado | archived | Valor booleano que indica se a ocorrência deve ser criada em um estado arquivado. |
| Está bloqueado | blocked | Valor booleano que indica se o problema está marcado como bloqueado. |
| IDs de etiquetas | ID da etiqueta | 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. |
| lastUpdatedByIdentificação | N/D | ID exclusiva gerada pelo sistema do usuário que atualizou o problema pela última vez. |
| lastUpdatedTime | N/D | Hora em que o problema foi atualizado pela última vez (em horário universal coordenado (UTC)). |
| priority | N/D | Prioridade do problema, se um tiver sido atribuído. |
| projectId | N/D | ID exclusiva do projeto gerada pelo sistema. |
| projectName | Nomes do projeto, projeto | Projeto no qual esse problema pode ser encontrado. |

| Campo | Alias | Descrição |
|--------------|-------|---|
| ID curta | N/D | Identificador abreviado e de incremento automático para o problema. |
| status | N/D | Status do problema que indica se o problema está na lista de pendências ou na coluna a bordo. |
| ID de status | N/D | Identificador do status do sistema. |
| title | N/D | Título da edição. |

Campos do projeto

| Campo | Alias | Descrição |
|--------------------|---------|--|
| description | N/D | Descrição do projeto. |
| lastUpdatedTime | N/D | Hora em que os metadados do projeto foram atualizados pela última vez (em horário universal coordenado (UTC)). |
| projectName | project | Nome do projeto no espaço. |
| Caminho do projeto | 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 em CodeCatalyst. Os nomes de exibição não são exclusivos. |
| email | N/D | Endereço de e-mail do usuário. |
| lastUpdatedTime | N/D | Hora em que os metadados do usuário foram atualizados pela última vez (em horário universal coordenado (UTC)). |
| userName | username | Nome de usuário escolhido pelo usuário quando ele se inscreveu CodeCatalyst. Diferentemente dos nomes de exibição, os nomes de usuário não podem ser alterados. |

Solução de problemas da Amazon CodeCatalyst

As informações a seguir podem ajudá-lo a solucionar problemas comuns no CodeCatalyst. Você também pode usar o relatório de CodeCatalyst saúde da Amazon para determinar se há problemas de serviço que possam estar afetando sua experiência.

Tópicos

- [Solução de problemas gerais de acesso](#)
- [Solução de problemas de suporte](#)
- [Parte ou toda a Amazon CodeCatalyst não está disponível](#)
- [Não consigo criar um projeto no CodeCatalyst](#)
- [Quero enviar feedback em CodeCatalyst](#)
- [Solução de problemas com repositórios de origem](#)
- [Solução de problemas de projetos e plantas](#)
- [Solução de problemas com fluxos de trabalho](#)
- [Solução de problemas com a pesquisa em CodeCatalyst](#)
- [Solução de problemas com contas associadas ao seu espaço](#)
- [Solução de problemas com ambientes de desenvolvimento](#)
- [Solução de problemas com problemas](#)
- [Solução de problemas entre a Amazon CodeCatalyst e os AWS SDKs ou o AWS CLI](#)

Solução de problemas gerais de acesso

Esqueci minha senha

Problema: esqueci a senha que uso para minha ID do AWS construtor e da Amazon CodeCatalyst.

Possíveis correções: A maneira mais fácil de corrigir esse problema é redefinir sua senha.

1. Abra a [Amazon CodeCatalyst](#) e insira seu endereço de e-mail. Depois, escolha Continue (Continuar).
2. Escolha Esqueci a senha
3. Enviaremos um e-mail com um link para você alterar sua senha. Se você não vê o e-mail na sua caixa de entrada, verifique sua pasta de spam.

Parte ou toda a Amazon CodeCatalyst não está disponível

Problema: acessei ou segui um link para o CodeCatalyst console, mas vejo um erro.

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. Verifique o [relatório Health](#) 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 uma ID de AWS construtor que não tem a função de administrador do Space. Você deve ter essa função 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 seu navegador e tente novamente.

Solução de problemas de suporte

Recebo uma mensagem de erro ao acessar AWS Support a Amazon CodeCatalyst

Problema: Quando escolho a CodeCatalyst opção AWS Support para a Amazon, 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 a Conta da AWS função necessária à conta de cobrança do espaço. A conta designada como conta de cobrança do espaço usa a `AWSRoleForCodeCatalystSupport`

função e a política `AmazonCodeCatalystSupportAccess` gerenciada. Para obter mais informações, consulte [Criando a `AWSRoleForCodeCatalystSupport` função para sua conta e espaço](#).

Note

Um AWS Builder ID só pode obter suporte para o alias com o qual está autenticado e somente para recursos com base nas permissões em `CodeCatalyst`. O suporte de conta e cobrança está disponível para todos os usuários no espaço. No entanto, os criadores só podem obter suporte para recursos e informações para os quais tenham permissão `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 espacial para que os usuários do espaço criem casos de suporte técnico. Peça ao administrador do espaço que adicione um AWS Support plano à sua conta de faturamento do espaço ou acesse <https://repost.aws/> para perguntar à AWS comunidade.

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 em CodeCatalyst.

Correções: se um usuário com a função de administrador do Space mudar de conta de faturamento do espaço, isso desconectará o AWS Support plano e todos os casos associados do espaço. Os AWS Support casos associados à antiga conta de cobrança espacial não estarão mais visíveis na AWS Support Amazon CodeCatalyst. O usuário raiz dessa conta de faturamento pode visualizar e resolver casos antigos a partir do AWS Management Console e pode configurar permissões do IAM AWS Support para que outros usuários visualizem e resolvam casos antigos. Você não poderá continuar recebendo suporte técnico da antiga conta CodeCatalyst de faturamento espacial por meio do AWS Management Console, mas poderá receber suporte técnico para outros serviços até que seu AWS Support plano seja cancelado.

Para obter mais informações, consulte [Atualizando, resolvendo e reabrindo seu caso](#) no Guia do AWS Support usuário.

Não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) na AWS Support Amazon CodeCatalyst

Problema: Não consigo abrir um caso de suporte para outro AWS service (Serviço da AWS) em AWS Support for CodeCatalyst.

Possíveis correções: você só pode abrir casos de CodeCatalyst suporte a partir de AWS Support for CodeCatalyst. Se você precisar de suporte para serviços ou recursos implantados de outro serviço AWS, da CodeCatalyst Amazon ou de outro serviço terceirizado, precisará criar um caso por meio do canal de suporte de serviços terceirizado AWS Management Console ou do canal de suporte de serviços terceirizado. Para obter mais informações, consulte [Criação de casos de suporte e gerenciamento de casos](#) no Guia AWS Support do usuário.

Parte ou toda a Amazon CodeCatalyst não está disponível

Problema: acessei ou segui um link para o CodeCatalyst console, mas vejo um erro.

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. Verifique o [relatório Health](#) 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 uma ID de AWS construtor que não tem a função de administrador do Space. Você deve ter essa função 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 seu navegador e tente novamente.

Quero enviar feedback em CodeCatalyst

Problema: Encontrei um bug CodeCatalyst e quero enviar um feedback.

Possíveis correções: você pode enviar feedback diretamente em CodeCatalyst.

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. No painel de navegação, escolha Enviar feedback.
3. Escolha o tipo de feedback no menu suspenso e insira seu feedback.

Solução de problemas com repositórios de origem

As informações a seguir podem ajudá-lo a solucionar problemas comuns com repositórios de origem em CodeCatalyst

Tópicos

- [Atingi o máximo de armazenamento do meu espaço e vejo avisos ou erros](#)
- [Eu recebo uma mensagem de erro ao tentar clonar ou enviar para um repositório de CodeCatalyst origem da Amazon](#)
- [Eu recebo uma mensagem de erro ao tentar confirmar ou enviar para um repositório de CodeCatalyst origem da Amazon](#)
- [Preciso de um repositório de origem para meu projeto](#)
- [Meu repositório de origem é totalmente novo, mas contém um commit](#)
- [Quero uma ramificação diferente como minha ramificação padrão](#)
- [Estou recebendo e-mails sobre atividades em pull requests](#)
- [Esqueci meu token de acesso pessoal \(PAT\)](#)
- [Um pull request não exibe as alterações que eu espero](#)
- [Uma pull request mostra o status Não mesclável](#)

Atingi o máximo de armazenamento do meu espaço e vejo avisos ou erros

Problema: quero enviar o código para um ou mais repositórios de origem em CodeCatalyst, mas vejo um erro. No console, vejo uma mensagem na página do repositório de origem informando que atingi o limite de armazenamento do espaço.

Possíveis correções: dependendo da sua função no projeto ou no espaço, você pode reduzir o tamanho de um ou mais repositórios de origem, excluir repositórios de origem não utilizados ou alterar seu nível de cobrança para um que tenha mais armazenamento.

- Para reduzir o tamanho de um repositório de origem em um projeto, você pode excluir ramificações não utilizadas. Para ter mais informações, consulte [Excluindo uma ramificação](#) e [Função de colaborador](#).
- Para reduzir o armazenamento geral de um espaço, você pode excluir repositórios de origem não utilizados. Para ter mais informações, consulte [Excluindo um repositório de origem](#) e [Função de administrador do projeto](#).
- Para aumentar a quantidade de armazenamento disponível para seu espaço, você pode alterar seu nível de cobrança para um com mais espaço de armazenamento. Para obter mais informações, consulte [Alterar seu nível CodeCatalyst de cobrança](#) no Amazon CodeCatalyst Administrator Guide.

Eu recebo uma mensagem de erro ao tentar clonar ou enviar para um repositório de CodeCatalyst origem da Amazon

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 seu AWS Builder ID, talvez não tenha configurado seu sistema de gerenciamento de credenciais com seu PAT ou seu PAT pode ter expirado. Experimente uma ou mais das seguintes soluções:

- Crie um token de acesso pessoal (PAT). Para ter mais informações, consulte [Conceda 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, você não é membro desse projeto ou não aceitou um convite para esse projeto. Para ter mais informações, consulte [Aceitando um convite e criando seu AWS Builder ID](#).
- Verifique se o comando clone está formatado corretamente e inclui sua ID do AWS Builder. 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 à sua ID do AWS Builder e que ele não esteja expirado. Se você não tiver um ou se o PAT estiver expirado, crie um. Para ter mais informações, consulte [Conceda 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 ter mais informações, consulte [Criar um ambiente de desenvolvimento](#).

Eu recebo uma mensagem de erro ao tentar confirmar ou enviar para um repositório de CodeCatalyst origem da Amazon

Problema: quando tento enviar para um repositório de origem, recebo um erro de permissões.

Possíveis correções: talvez você não tenha uma função no projeto que permita confirmar e enviar alterações de código para o projeto. Visualize sua função no projeto em que você está tentando enviar alterações para um repositório de origem. Para ter mais informações, consulte [Obter uma lista dos membros e suas funções no projeto](#) e [Concedendo acesso com funções de usuário](#).

Se você tem uma função 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 que você envie alterações de código para essa ramificação. Em vez disso, tente criar uma ramificação e enviar seu código para essa ramificação. Para ter mais informações, consulte [Gerencie 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 em CodeCatalyst. Se alguém com a função de administrador do Space instalar os GitHub repositórios e conectá-los a uma GitHub conta, você poderá vinculá-los aos GitHub repositórios disponíveis para adicioná-los ao seu projeto se tiver a função de administrador do projeto. Para obter 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 um commit

Problema: acabei de criar um repositório de origem. Ele deve estar vazio, mas contém um commit, uma ramificação e um README .md arquivo.

Possíveis correções: Esse é o comportamento esperado. Todos os repositórios de origem 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 markdown 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 eu quero uma ramificação diferente como minha ramificação padrão.

Possíveis correções: você não pode alterar ou excluir a ramificação padrão nos repositórios de origem em CodeCatalyst. Você pode criar ramificações adicionais e usar essas ramificações nas ações de origem nos fluxos de trabalho. Você também pode optar por vincular GitHub repositórios e usá-los como repositórios para seu projeto.

Estou recebendo e-mails sobre atividades em pull requests

Problema: não me inscrevi nem configurei notificações por e-mail sobre a atividade de pull request, mas as estou recebendo mesmo assim.

Possíveis correções: as notificações por e-mail são enviadas automaticamente sobre a atividade do pull request. Para ter 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. 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 ter mais informações, consulte [Conceda aos usuários acesso ao repositório com tokens de acesso pessoal](#).

Um pull request não exibe as alterações que eu espero

Problema: criei uma pull request, mas não vejo as mudanças que espero ver entre as ramificações de origem e de destino.

Possíveis correções: Isso pode ser causado por vários problemas. Experimente 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 pull request 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 pull request. Algumas diferenças podem ser muito grandes para serem exibidas. Para ter mais informações, consulte [Cotas para repositórios de origem em CodeCatalyst](#) e [Visualizando um arquivo](#).
- As pull requests exibem as diferenças entre a base de mesclagem e qualquer revisão que você escolher. Quando você cria uma pull request, a diferença exibida para você é a diferença entre a ponta da ramificação de origem e a ponta da ramificação de destino. Depois que a pull request 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 obter mais informações sobre diferenças e bases de mesclagem no Git, [git-merge-base](#) consulte a documentação do Git.

Uma pull request mostra o status Não mesclável

Problema: quero mesclar uma pull request, 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 pull request devem aprovar uma pull request 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 pull request.

Note

Se um revisor obrigatório tiver sido removido do seu projeto antes de aprovar a pull request, você não poderá mesclar a pull request. Feche a pull request e crie uma nova pull request.

- Pode haver um conflito de mesclagem entre a ramificação de origem e a ramificação de destino. CodeCatalyst não suporta 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 ter mais informações, consulte [Mesclando uma pull request](#).

Solução de problemas de projetos e plantas

Esta seção pode ajudá-lo a solucionar alguns problemas comuns que você pode encontrar ao trabalhar com projetos e plantas na Amazon. CodeCatalyst

API Java com dependências ausentes do AWS Fargate blueprint para apache-maven-3.8.6

Problema: para um projeto criado a partir da API Java com AWS Fargate blueprint, o fluxo de trabalho falha com um erro por apache-maven-3.8.6 dependências ausentes. O fluxo de trabalho falha com uma saída semelhante ao exemplo a seguir:

```
Step 8/25 : RUN wget https://d1cdn.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://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving d1cdn.apache.org (d1cdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.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://d1cdn.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 blueprint 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 blueprint. AWS Fargate
2. Atualize o Dockerfile (`/static-assets/app/Dockerfile`) para usá-lo `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.

Abaixo está 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. Usando as etapas a seguir, encontre sua função e atualize as permissões da política de função para permitir 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 suas políticas designadas é armazenado em seu repositório de origem no diretório `/.codecatalyst/workflows`. Não é necessário modificar o fluxo de trabalho YAML, a menos que você esteja adicionando uma nova ID de função ao fluxo de trabalho existente. Para obter informações sobre elementos e formatação do modelo YAML, consulte [Definição de fluxo de trabalho YAML](#).

Essas são as etapas de alto nível a serem seguidas para editar sua política de função e verificar o fluxo de trabalho YAML.

Para referenciar o nome da sua função no fluxo de trabalho YAML e atualizar a política

1. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço. Navegue até seu projeto.
3. Escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
4. Escolha o fluxo de trabalho intitulado `OnPullRequest`. Escolha a guia Definição.
5. No fluxo de trabalho YAML, no `Role`: campo abaixo da ação `codeguru_codereview`, anote o nome da função. Essa é a função da política que você modificará no IAM. O exemplo a seguir mostra o nome da função.

The screenshot displays the Amazon CodeCatalyst interface for a workflow named 'OnPullRequest'. The workflow is associated with the repository 'mysfitscvb63' and is currently in a 'Valid' state. The workflow definition is shown in a code editor, detailing the following configuration:

```

1 Name: OnPullRequest
2 SchemaVersion: "1.0"
3 Triggers:
4   - Type: PULLREQUEST
5     Branches:
6       - main
7     Events:
8       - OPEN
9       - REVISION
10 Actions:
11   codeguru_codereview:
12     Identifier: aws/build@v1
13     Inputs:
14       Sources:
15         - WorkflowSource
16     Variables:
17       - Name: AWS_DEFAULT_REGION
18         Value: us-west-2
19     Outputs:
20       Artifacts:
21         - Name: codereview
22           Files:
23             - ./code-guru/*
24     Configuration:
25       Steps:
26         - Run: curl -OL https://github.com/aws/aws-codeguru-c
27         - Run: unzip aws-codeguru-cli.zip
28         - Run: export PATH=$PATH:./aws-codeguru-cli/bin
29         - Run: aws-codeguru-cli --root-dir ./src --no-prompt -
30     Environment:
31       Name: development
32     Connections:
33       - Name: connection-11-30
34         Role: CodeCatalystPreviewDevelopmentAdministrator-j

```

6. Faça um dos seguintes procedimentos:

- (Recomendado) Atualize a função de serviço conectada ao seu projeto com as permissões necessárias para a Amazon CodeGuru e o Amazon CodeGuru Reviewer. A função terá um nome `CodeCatalystWorkflowDevelopmentRole-spaceName` com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName*de serviço](#). 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 Amazon CodeGuru e Amazon CodeGuru Reviewer ou crie uma nova função com as permissões necessárias.
7. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

No console do IAM, encontre a função na etapa 5, como `CodeCatalystPreviewDevelopmentRole`.

8. Na função da etapa 5, altere a política de permissão para incluir as `codeguru:*` permissões `codeguru-reviewer:*` e. Depois de adicionar essas permissões, a política de permissões deve ser semelhante à seguinte:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "apigateway:*",
        "ecr:*",
        "ecs:*",
        "ssm:*",
        "codedeploy:*",
        "s3:*",
        "iam:DeleteRole",
        "iam:UpdateRole",
        "iam:Get*",
        "iam:TagRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:CreatePolicy",
        "iam:DeletePolicy",
        "iam:CreatePolicyVersion",
        "iam:DeletePolicyVersion",
        "iam:PutRolePermissionsBoundary",
        "iam:DeleteRolePermissionsBoundary",
        "sts:AssumeRole",

```

```
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:DescribeRules",
        "elasticloadbalancing:ModifyRule",
        "cloudwatch:DescribeAlarms",
        "sns:Publish",
        "sns:ListTopics",
        "codeguru-reviewer:*",
        "codeguru:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
```

9. Depois de fazer as correções de política, retorne CodeCatalyst e inicie a execução do fluxo de trabalho novamente.

Ainda está procurando resolver seu problema?

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.

Solução de problemas com fluxos de trabalho

Consulte as seções a seguir para solucionar problemas relacionados aos fluxos de trabalho na Amazon. CodeCatalyst Para obter mais informações sobre fluxos de trabalho, consulte [Crie, teste e implante com fluxos de trabalho em CodeCatalyst](#).

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 não tem erros”?](#)
- [Como faço para corrigir os erros “Não é possível localizar credenciais” e ExpiredToken ""?](#)
- [Como faço para corrigir os erros “Não é possível conectar-se ao servidor”?](#)
- [Por que os CodeDeploy campos estão ausentes no editor visual?](#)

- [Como faço para corrigir erros de recursos do IAM?](#)
- [Como faço para corrigir erros de “instalação do 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 tempo de execução?](#)
- [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 muitos 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 CodeCatalyst console, em CI/CD, Fluxos de trabalho, seu fluxo de trabalho aparece com a seguinte mensagem:

```
Workflow is inactive.
```

Essa mensagem indica que o arquivo de definição do fluxo de trabalho contém um acionador que não se aplica à ramificação em que você está atualmente. Por exemplo, seu arquivo de definição de fluxo de trabalho pode conter um PUSH gatilho que faz referência à sua main ramificação, mas você está em uma ramificação de recursos. Como as alterações que você está fazendo em sua ramificação de recursos não se aplicam e não iniciarão a main execução do fluxo de trabalho, CodeCatalyst desativa o fluxo de trabalho na ramificação e o marca como `Inactive`.

Correções possíveis:

Se você quiser iniciar um fluxo de trabalho em sua ramificação de recursos, você pode fazer o seguinte:

- Em sua ramificação de recursos, no arquivo de definição do fluxo de trabalho, remova a `Branches` propriedade da `Triggers` seção 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 estiver ativado, 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 `Triggers` seção e execute o fluxo de trabalho manualmente.
- Em sua ramificação de recursos, no arquivo de definição do fluxo de trabalho, altere a `PUSH` seção para que ela faça referência à sua ramificação de recursos em vez de outra ramificação (como `main`, por exemplo).

Important

Tenha cuidado para não cometer essas alterações se você não pretende mesclá-las de volta à sua `main` filial.

Para obter mais informações sobre a edição do arquivo de definição do fluxo de trabalho, consulte [Criação de um fluxo de trabalho](#).

Para obter mais informações sobre gatilhos, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).

Como faço para corrigir os erros “A definição do fluxo de trabalho **não** tem erros”?

Problema: Você vê qualquer uma das seguintes mensagens de erro:

Erro 1:

Na página CI/CD, Fluxos de trabalho, abaixo do nome do seu fluxo de trabalho, você vê:

```
Workflow definition has n errors
```

Erro 2:

Ao editar um fluxo de trabalho, você escolhe o botão Validar e a seguinte mensagem aparece na parte superior do CodeCatalyst console:

```
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, você verá o seguinte erro no campo Definição do fluxo de trabalho:

```
n errors
```

Correções possíveis:

- Escolha CI/CD, escolha Fluxos de trabalho e escolha o nome do fluxo de trabalho que tem o erro. No campo Definição do fluxo de trabalho próximo à parte superior, escolha o link para o erro. Os detalhes sobre o erro aparecem na parte inferior da página. Siga as dicas de solução de problemas do erro para corrigir o problema.
- Verifique se o arquivo de definição do fluxo de trabalho é um arquivo YAML.
- Certifique-se de que as propriedades YAML no arquivo de definição do fluxo de trabalho estejam aninhadas no nível correto. Para ver como as propriedades devem ser aninhadas no arquivo de definição do fluxo de trabalho [Definição de fluxo de trabalho YAML](#), consulte ou consulte a documentação da ação, que está vinculada a [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#).
- Certifique-se de que os asteriscos (*) e outros caracteres especiais tenham escapado corretamente. Para escapar deles, adicione aspas simples ou duplas. Por exemplo: .

```
Outputs:  
  Artifacts:  
    - Name: myartifact  
      Files:  
        - "**/*"
```

Para obter mais informações sobre caracteres especiais no arquivo de definição do fluxo de trabalho, consulte [Diretrizes e convenções de sintaxe](#).

- Certifique-se de que as propriedades YAML no arquivo de definição do fluxo de trabalho usem a capitalização correta. Para obter mais informações sobre regras de maiúsculas e minúsculas, consulte [Diretrizes e convenções de sintaxe](#). Para determinar o revestimento correto de cada

propriedade, consulte ou consulte a [Definição de fluxo de trabalho YAML](#) documentação de sua ação, que está vinculada ao formulário. [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#)

- Verifique se a `SchemaVersion` propriedade está presente e definida para a versão correta no arquivo de definição do fluxo de trabalho. Para ter mais informações, consulte [SchemaVersion](#).
- Certifique-se de que a `Triggers` seção no arquivo de definição do fluxo de trabalho inclua todas as propriedades necessárias. Para determinar as propriedades necessárias, escolha o gatilho no [editor visual](#) e procure os campos com informações ausentes ou consulte a documentação de referência do acionador em [Triggers](#).
- Certifique-se de que a `DependsOn` propriedade no arquivo de definição do fluxo de trabalho esteja configurada corretamente e não introduza dependências circulares. Para ter mais informações, consulte [Configurando ações para depender de outras ações](#).
- Certifique-se de que a `Actions` seção no arquivo de definição do fluxo de trabalho inclua pelo menos uma ação. Para ter mais informações, consulte [Ações](#).
- Certifique-se de que cada ação inclua todas as propriedades necessárias. Para determinar as propriedades necessárias, escolha a ação no [editor visual](#) e procure campos com informações ausentes ou consulte a documentação da ação, que está vinculada a partir de [Adicionar uma ação a um CodeCatalyst fluxo de trabalho](#).
- Certifique-se de que todos os artefatos de entrada tenham artefatos de saída correspondentes. Para ter mais informações, consulte [Definindo um artefato de saída](#).
- Certifique-se de que as variáveis definidas em uma ação sejam exportadas para que possam ser usadas em outras ações. Para ter mais informações, consulte [Exportar 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 [Tutorial: Implantar um aplicativo 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 que você está usando para acessar AWS os serviços expiraram. Nesse caso, não execute o `aws configure` comando. Em vez disso, use as instruções a seguir para atualizar sua chave de AWS acesso e token de sessão.

Para atualizar sua chave de AWS acesso e token de sessão

1. Verifique se você tem o URL do portal de AWS acesso, o nome de usuário e a senha do usuário 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](#) o tutorial.

 Note

Se você não tiver essas informações, acesse a página de `codecatalyst-eks-user` detalhes no IAM Identity Center, escolha Redefinir senha, Gerar uma senha de uso único [...] e redefina a senha novamente para exibir as informações na tela.

2. Execute um destes procedimentos:
 - Cole o URL do portal de AWS acesso na barra de endereço do seu navegador.

Ou

 - Atualize a página do portal de AWS acesso se ela já estiver carregada.
3. Faça login com o nome `codecatalyst-eks-user` de usuário e a senha, se você ainda não estiver conectado.
4. Escolha e Conta da AWS, em seguida, escolha o nome do Conta da AWS ao qual você atribuiu o `codecatalyst-eks-user` usuário e o conjunto de permissões.
5. Ao lado do nome do conjunto de permissões (`codecatalyst-eks-permission-set`), escolha Linha de comando ou acesso programático.
6. Copie os comandos no meio da página. Eles são parecidos com os seguintes:

```
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 sequência 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.

Agora você atualizou suas credenciais. Os `kubectl` comandos `AWS CLI``eksctl`, e agora devem funcionar.

Como faço para corrigir os erros “Não é possível conectar-se ao servidor”?

Problema: Ao trabalhar no tutorial descrito em [Tutorial: Implantar um aplicativo 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 `kubectl` utilitário está usando para se conectar ao seu cluster Amazon EKS expiraram. Para resolver o problema, atualize as credenciais digitando o seguinte comando 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 em que seu AWS cluster está implantado.

Por que os CodeDeploy campos estão ausentes no editor visual?

Problema: você está usando uma ação [Deploy to Amazon ECS](#) e não está vendo os CodeDeploy campos, como CodeDeploy AppSpec no editor visual do fluxo de trabalho. Esse problema pode ocorrer porque o serviço Amazon ECS que você especificou no campo Serviço não está configurado para realizar implantações em azul/verde.

Correções possíveis:

- Escolha um serviço diferente do Amazon ECS na guia Configuração da ação Deploy to Amazon ECS. Para ter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Container Service \(ECS\) com um fluxo de trabalho](#).
- Configure o serviço Amazon ECS selecionado para realizar implantações em azul/verde. Para obter mais informações sobre a configuração de implantações azul/verde, consulte [Implantação azul/verde no Amazon Elastic Container Service](#) CodeDeploy Developer Guide.

Como faço para corrigir erros de recursos do IAM?

Problema: você está usando uma ação [Deploy AWS CloudFormation stack](#) e vê `##[error] requires capabilities: [capability-name]` nos registros da ação Deploy AWS CloudFormation stack.

Possíveis correções: conclua o procedimento a seguir para adicionar o recurso ao arquivo de definição do fluxo de trabalho. Para obter mais informações sobre os recursos do IAM, consulte Como [reconhecer recursos do IAM em AWS CloudFormation modelos](#) no Guia do usuário do IAM.

Visual

Para adicionar um recurso do IAM 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha Visual.
7. No diagrama do fluxo de trabalho, escolha sua ação Deploy AWS CloudFormation stack.
8. Escolha a guia Configuração.
9. Na parte inferior, escolha Avançado - opcional.
10. Na lista suspensa Capacidades, marque a caixa de seleção ao lado da capacidade mencionada na mensagem de erro. Se o recurso não estiver disponível na lista, use o editor YAML para adicioná-lo.

11. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
12. Escolha 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 obter mais informações sobre como executar um fluxo de trabalho manualmente, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

YAML

Para adicionar um recurso do IAM 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, em seguida, escolha Fluxos de trabalho.
4. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
5. Selecione a opção Editar.
6. Escolha YAML.
7. Na ação Deploy AWS CloudFormation stack, adicione uma `capabilities` propriedade, como esta:

```
DeployCloudFormationStack:
  Configuration:
    capabilities: capability-name
```

Substitua *capability-name* pelo nome do recurso do IAM mostrado na mensagem de erro. Use vírgulas e nenhum espaço para listar vários recursos. Para obter mais informações, consulte a descrição da `capabilities` propriedade no [Definição YAML da ação “Implantar AWS CloudFormation pilha”](#).

8. (Opcional) Escolha Validar para validar o código YAML do fluxo de trabalho antes de confirmar.
9. Escolha 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 obter mais informações sobre como executar um fluxo de trabalho manualmente, consulte [Iniciando um fluxo de trabalho executado manualmente](#).

Como faço para corrigir erros de “instalação do npm”?

Problema: sua [ação de AWS CDK implantação ou ação](#) de [AWS CDK bootstrap](#) falha com um `npm install` erro. Esse erro pode ocorrer porque você está armazenando as dependências do seu AWS CDK aplicativo 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 `cdk.json` arquivo do seu AWS CDK aplicativo com informações adicionais de registro e autenticação.

Antes de começar

1. Crie segredos para suas informações de autenticação. Você referenciará esses segredos no `cdk.json` arquivo em vez de fornecer os equivalentes em texto não criptografado. Para criar os segredos:
 - a. Abra o CodeCatalyst console em <https://codecatalyst.aws/>.
 - b. Selecione o projeto.
 - c. No painel de navegação, escolha CI/CD e, em seguida, escolha Segredos.
 - d. Crie dois segredos com as seguintes propriedades:

| Primeiro segredo | Segundo segredo |
|---|--|
| Nome: <code>npmUsername</code> | Nome: <code>npmAuthToken</code> |
| Valor: <i>npm-username</i> , em que <i>npm-username</i> é o nome de usuário usado para se autenticar em seu registro npm privado. | Valor: <i>npm-auth-token</i> , onde <i>npm-auth-token</i> está o token de acesso usado para autenticar seu registro npm privado. Para obter mais informações sobre tokens de acesso npm, consulte Sobre tokens de acesso na documentação do npm. |
| (Opcional) Descrição: <code>The username used to authenticate to the private npm registry.</code> | |

| Primeiro segredo | Segundo segredo |
|------------------|--|
| | (Opcional) Descrição: The access token used to authenticate to the private npm registry. |

Para obter mais informações sobre segredos, consulte [Configurando e usando segredos em um fluxo de trabalho](#).

2. Adicione os segredos como variáveis de ambiente à sua AWS CDK ação. A ação substituirá as variáveis por valores reais quando for executada. Para adicionar os segredos:
 - a. No painel de navegação, escolha CI/CD e, em seguida, escolha Fluxos de trabalho.
 - b. Escolha o nome do seu fluxo de trabalho. Você pode filtrar pelo nome do repositório ou ramificação de origem em que o fluxo de trabalho está definido, ou filtrar pelo nome do fluxo de trabalho.
 - c. Selecione a opção Editar.
 - d. Escolha Visual.
 - e. No diagrama do fluxo de trabalho, escolha sua AWS CDK ação.
 - f. Escolha a guia Entradas.
 - g. Adicione duas variáveis com as seguintes propriedades:

| Primeira variável | Segunda variável |
|---|--|
| Nome: NPMUSER | Nome: NPMTOKEN |
| Valor: <code>\${Secrets.npmUsername}</code> | Valor: <code>\${Secrets.npmAuthToken}</code> |

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 CDK bootstrap; uma ação de AWS CDK implantação será semelhante.

```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
  CDKBootstrapAction:
    Identifier: aws/cdk-bootstrap@v1
    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 NPMTOKEN variáveis NPMUSER e em seu `cdk.json` arquivo. Vá para o próximo procedimento.

Para atualizar seu arquivo `cdk.json`

1. Vá para o diretório raiz do seu AWS CDK projeto e abra o `cdk.json` arquivo.
2. Encontre a "app": propriedade e altere-a para incluir o código mostrado em *itálico vermelho*:

Note

O código de exemplo a seguir é de um TypeScript projeto. Se você estiver usando um JavaScript projeto, 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/ com o caminho* para as dependências do seu projeto em seu registro privado. AWS CDK
- *hello-cdk.ts|.js* com o nome do seu arquivo de ponto de entrada. Isso pode ser um arquivo .ts (TypeScript) ou .js (JavaScript), dependendo do idioma que você está usando.

Note

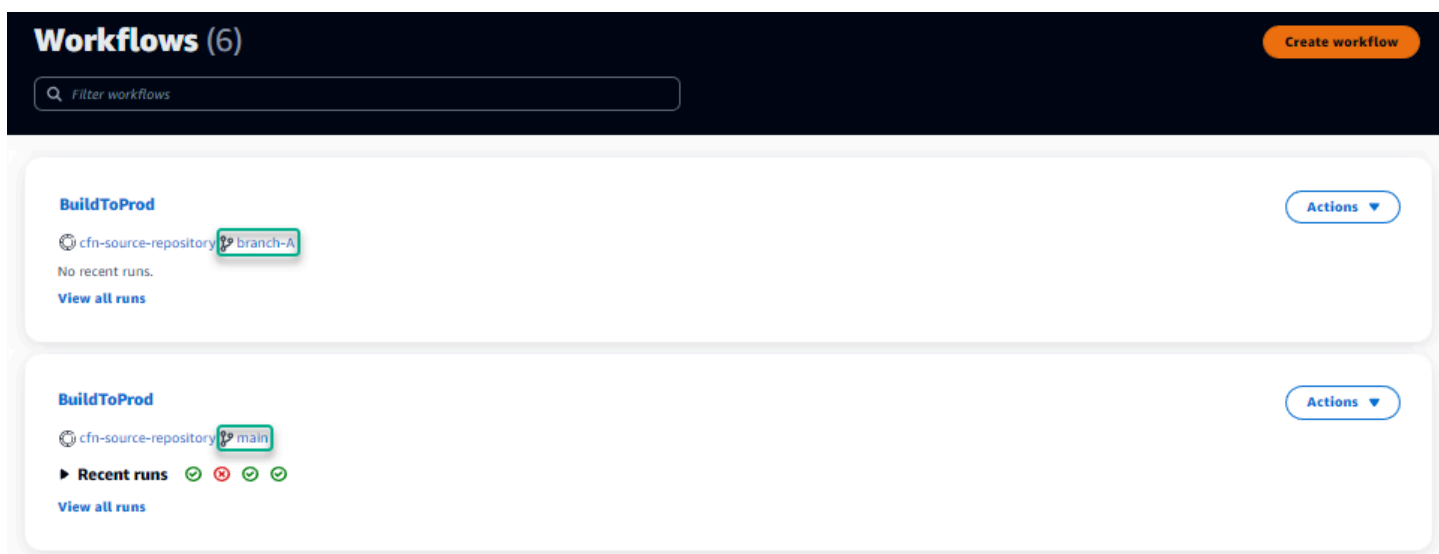
A ação substituirá as variáveis ***NPMUSER e NPMTOKEN pelo nome de usuário e token*** de acesso npm que você especificou em Segredos.

4. Salve o arquivo `cdk.json`.

5. Execute novamente a ação manualmente para ver se as alterações corrigem o erro. Para obter mais informações sobre como executar ações manualmente, consulte [Iniciando 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 filiais diferentes. Na página Fluxos de trabalho, você pode diferenciar fluxos de trabalho com o mesmo nome observando o nome da ramificação. Para ter mais informações, consulte [Organizando seu trabalho de código-fonte com filiais na Amazon CodeCatalyst](#).



Posso armazenar meus arquivos de definição de fluxo de trabalho em outra pasta?

Não, você deve armazenar todos os arquivos de definição do fluxo de trabalho na `.codecatalyst/workflows` pasta. Se você estiver usando um repositório mono com vários projetos lógicos, coloque todos os seus arquivos de definição de fluxo de trabalho na `.codecatalyst/workflows` pasta e, em seguida, use a `FilesChanged` propriedade dentro de um acionador para acionar o fluxo de trabalho em um caminho de projeto especificado. Para ter mais informações, consulte [Iniciando um fluxo de trabalho executado automaticamente com gatilhos](#).

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, você pode definir uma dependência em outra ação definindo o `DependsOn` campo. Você também pode configurar uma ação para consumir artefatos ou variáveis que são saídas de outras ações. Para ter mais informações, consulte [Configurando ações para depender de outras ações](#).

Por que meu fluxo de trabalho é validado com sucesso, mas falha em tempo de execução?

Se você validou seu fluxo de trabalho usando o `Validate` botão, mas seu fluxo de trabalho falhou de qualquer maneira, isso pode ser devido a uma limitação no validador.

Qualquer erro relacionado a um CodeCatalyst recurso, 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 ter 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 por CodeCatalyst.

Possíveis correções: Isso pode ser causado por vários problemas. Experimente uma ou mais das seguintes soluções:

- Certifique-se de que a ferramenta usada para executar testes produza saídas em um dos formatos que CodeCatalyst compreende. Por exemplo, se você quiser permitir `pytest` CodeCatalyst a descoberta de relatórios de cobertura de teste e código, inclua os seguintes argumentos:

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Para ter 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 em `JUnitXML` formato, verifique se a extensão do arquivo é `.xml`. Para ter 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 `ExcludePaths` não excluir 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 ter 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` exclua 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 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 ter mais informações, consulte [Configurando 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, CodeCatalyst gera relatórios de cobertura de código para arquivos incluídos nas dependências do meu aplicativo armazenadas em `node_modules`.

Possíveis correções: você pode ajustar a `ExcludePaths` configuração para excluir arquivos indesejados. Por exemplo, para excluir `node_modules`, adicione `node_modules/**/*`. Para ter mais informações, consulte [Incluir/excluir caminhos](#).

A descoberta automática gera muitos 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 Maven Surefire Plugin, você pode usar `npx junit-merge` para agregar os arquivos manualmente. A expressão completa pode ter a seguinte aparência:

```
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 coincidem com os da ~/ .codecatalyst/workflows/ pasta no seu repositório de origem](#). Você pode ver as seguintes incompatibilidades:

- Um fluxo de trabalho aparece 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 em seu repositório de origem, mas um fluxo de trabalho correspondente não aparece na página Fluxos de trabalho.
- Existe um fluxo de trabalho no repositório de origem e na página Fluxos de trabalho, mas os dois são diferentes.

Esse problema pode ocorrer se a página Fluxos de trabalho não tiver tido tempo de atualizar ou se uma cota de fluxo de trabalho tiver sido excedida.

Correções possíveis:

- Espere. Normalmente, você precisa esperar dois ou três segundos após uma confirmação na fonte antes de ver a alteração na página Fluxos de trabalho.
- Se você excedeu uma cota de fluxo de trabalho, faça o seguinte:

Note

Para determinar se uma cota de fluxo de trabalho foi excedida [Cotas para fluxos de trabalho](#), revise 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 o número máximo de fluxos de trabalho por cota de espaço, exclua alguns fluxos de trabalho e, em seguida, realize um teste de confirmação no arquivo de definição do fluxo de trabalho. Um exemplo de confirmação de teste pode ser adicionar um espaço ao arquivo.
- Se você excedeu a cota máxima de tamanho do arquivo de definição de fluxo de trabalho, altere o arquivo de definição de fluxo de trabalho para reduzir seu tamanho.
- Se você excedeu o número máximo de arquivos de fluxo de trabalho processados em uma única cota de eventos de origem, realize várias confirmações de teste. Modifique menos do que o número máximo de fluxos de trabalho em cada confirmação.
- Aumente as cotas do fluxo de trabalho ativando o faturamento por níveis pagos. Para obter mais informações, consulte [Gerenciando o faturamento](#) no Guia do CodeCatalyst Administrador da Amazon.

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 da sua função 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 ter mais informações, consulte [Arquivos de definição de fluxo de trabalho](#). A função de administrador de espaço, a função de administrador de projeto e a função de colaborador têm permissão para confirmar e enviar código para repositórios em um projeto.

Se você tem a função 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 os usuários com essa função enviem código para essa ramificação específica. Tente criar um fluxo de trabalho em uma ramificação diferente ou confirmar suas alterações em uma ramificação diferente. Para ter mais informações, consulte [Gerencie as ações permitidas para uma ramificação com regras de ramificação](#).

Solução de problemas com a pesquisa em CodeCatalyst

Consulte as seções a seguir para solucionar problemas relacionados à pesquisa. CodeCatalyst Para obter mais informações sobre fluxos de trabalho, consulte [Pesquise código, problemas, projetos e usuários em 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 ver os detalhes de um usuário, não vejo as informações dele no projeto.

Possíveis correções: no momento, a pesquisa não oferece suporte à pesquisa de usuários em um projeto. Para pesquisar usuários com acesso ao seu espaço, alterne para Este espaço em QuickSearch 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 vários 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 obter mais informações sobre como refinar suas consultas, consulte [Refinando 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 à medida que você navega pelas páginas.

Ao navegar pelos resultados, você pode ver a seguinte mensagem: Nenhum resultado para “teste”. Você receberá a mensagem se não tiver acesso aos resultados restantes.

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 contas associadas ao seu espaço

Em CodeCatalyst, você pode adicionar um Conta da AWS ao seu espaço para conceder permissões aos recursos e para fins de cobrança. As informações a seguir podem ajudá-lo a solucionar problemas comuns com contas associadas no CodeCatalyst.

Tópicos

- [Minha solicitação de Conta da AWS conexão recebe um erro de token inválido](#)

- [O fluxo de trabalho do meu CodeCatalyst projeto da Amazon falha com um erro na conta configurada, no ambiente ou na função do IAM](#)
- [Preciso de uma conta, função e ambiente associados para criar um projeto](#)
- [Não consigo acessar a página do Amazon CodeCatalyst Spaces no AWS Management Console](#)
- [Quero uma conta diferente como minha conta de cobrança](#)

Minha solicitação de Conta da AWS conexão 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: certifique-se de fornecer o ID da conta que você deseja adicionar ao seu espaço. Você deve ter permissões administrativas para sua conta 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 AWS Management Console. É necessário que a mesma conta esteja logada no lado do console. Tente novamente depois de verificar o seguinte:

- Você está logado no AWS Management Console com o mesmo Conta da AWS que deseja adicionar ao seu espaço.
- Você está logado no AWS Management Console com o Oeste dos EUA (Oregon) Região da AWS (us-west-2) selecionado.
- Se você chegou da página de cobrança e deseja adicioná-la Conta da AWS como uma conta de cobrança especificada ao seu espaço, verifique se a conta ainda não é uma conta de cobrança de outro espaço.

O fluxo de trabalho do meu CodeCatalyst projeto da Amazon falha com um erro na conta configurada, no ambiente ou na função do IAM

Problema: quando o fluxo de trabalho é executado e não encontra uma conta configurada ou funções do IAM associadas ao seu espaço, você deve preencher os campos de função, 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:

- A função não está disponível para uso com a conexão associada ao ambiente.
- A ação não teve sucesso. 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 teve sucesso. Status: FALHA; O valor fornecido para a função do IAM não é válido. Verifique se o nome existe, se a função do IAM foi adicionada à conexão da sua conta e se a conexão já está associada ao seu CodeCatalyst espaço na Amazon

Possíveis correções: verifique se os campos YAML do fluxo de trabalho têm valores precisos para [Ambiente](#), [Conexões](#) e [Função](#). As ações do CodeCatalyst fluxo de trabalho que exigem um ambiente são ações de criação ou implantação que executam AWS recursos ou geram pilhas AWS de recursos.

Escolha o bloco de ação do fluxo de trabalho com falha e, em seguida, escolha Visual. Escolha a guia Configuração. Se os campos Ambiente, Nome da conexão e Nome da função não estiverem preenchidos, você precisará atualizar manualmente o fluxo de trabalho. Use as etapas a seguir para editar seu fluxo de trabalho YAML:

- Expanda o `/.codecatalyst` diretório e, em seguida, expanda o `/workflows` diretório. Abra o arquivo YAML do fluxo de trabalho. Certifique-se de que as funções e as informações da conta do IAM estejam especificadas no YAML que você configurou 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 Environment, Connection e Role são necessárias para executar ações CodeCatalyst de criação e implantação de fluxo de trabalho com AWS recursos. Para ver um exemplo, consulte os parâmetros YAML de referência da ação de CodeCatalyst construção para [Ambiente](#), [Conexões](#) e [Função](#).

- Certifique-se de que seu espaço tenha uma conta adicionada a ele e certifique-se de que a conta tenha a função ou funções apropriadas do IAM adicionadas à conta. Você pode ajustar ou adicionar contas se tiver a função de administrador do Space. Para ter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Preciso de uma conta, função 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 autorizações para adicioná-las ao seu projeto se tiver a função de administrador do Space. Você também deve ter um Conta da AWS local onde tenha permissões administrativas ou possa trabalhar com seu AWS administrador.

Para garantir que uma conta e uma função estejam disponíveis na tela de criação do projeto, você deve primeiro adicionar a conta e as funções. Para ter mais informações, consulte [Permitindo acesso a AWS recursos com conexão Contas da AWS](#).

Você tem a opção de escolher criar uma função de serviço com uma política de função chamada política de CodeCatalystWorkflowDevelopmentRole-*spaceName*função. A função terá um nome CodeCatalystWorkflowDevelopmentRole-*spaceName* com um identificador exclusivo anexado. Para obter mais informações sobre a função e a política de funções, consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName*de serviço](#). Para obter as etapas para criar a função, consulte [Criando a CodeCatalystWorkflowDevelopmentRole-*spaceName*função para sua conta e espaço](#). A função é adicionada à sua conta e está disponível nas páginas de criação do projeto em CodeCatalyst.

Não consigo acessar a página do Amazon CodeCatalyst Spaces no AWS Management Console

Problema: Quando tento acessar a CodeCatalyst página da Amazon no AWS Management Console para adicionar uma conta ao meu CodeCatalyst espaço ou adicionar funções a uma conta emAWS, recebo um erro de permissão.

Correções possíveis:

Para o seu espaço, você pode adicionar Contas da AWS autorizações para adicioná-las ao seu projeto se tiver a função de administrador do espaço. Você também deve ter um Conta da AWS local

onde tenha permissões administrativas ou possa trabalhar com seu AWS administrador. Primeiro, você deve se certificar de que está conectado AWS Management Console com a mesma conta que deseja gerenciar. Depois de fazer login no AWS Management Console, você pode abrir o console e tentar novamente.

Abra a CodeCatalyst página da Amazon AWS Management Console em <https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#/>.

Quero uma conta diferente como minha conta de cobrança

Problema: Ao configurar meu CodeCatalyst login, concluí várias etapas para configurar meu espaço e associar um autorizado Conta da AWS. Agora, quero autorizar uma conta diferente para cobrança.

Possíveis correções: para seu espaço, você pode autorizar contas de cobrança se tiver a função de administrador do Space. Você também deve ter um Conta da AWS local onde tenha permissões administrativas ou possa trabalhar com seu AWS administrador.

Para obter mais informações, consulte [Gerenciando o faturamento](#) no Guia do CodeCatalyst Administrador da Amazon.

Solução de problemas com ambientes de desenvolvimento

Consulte as seções a seguir para solucionar problemas relacionados aos ambientes de desenvolvimento. Para obter 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 desconectado](#)
- [Meu ambiente de desenvolvimento conectado ao VPC falhou](#)
- [Não consigo encontrar em qual diretório meu projeto está](#)
- [Não consigo me conectar ao meu ambiente de desenvolvimento via SSH](#)

- [Não consigo me conectar ao meu ambiente de desenvolvimento via SSH porque minha configuração SSH local está ausente](#)
- [Não consigo me conectar ao meu ambiente de desenvolvimento via SSH porque estou tendo problemas com AWS Config o codecatalyst meu perfil](#)
- [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 Dev Environments informando que atingi o limite de armazenamento do espaço.

Possíveis correções: dependendo da sua função no projeto ou no espaço, você pode excluir um ou mais dos seus próprios ambientes de desenvolvimento ou, se tiver a função de administrador do espaço, poderá excluir ambientes de desenvolvimento não utilizados criados por outros usuários. Você também pode decidir alterar o nível de cobrança para um que inclua mais armazenamento.

- Para ver os limites de armazenamento, consulte a guia Faturamento do CodeCatalyst espaço da Amazon para ver se as cotas de uso atingiram o máximo permitido. Se as cotas tiverem atingido o máximo, entre em contato com alguém com a função de administrador do Space para remover ambientes de desenvolvimento desnecessários ou considerar a alteração do nível de cobrança.
- 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 seu IDE, verifique se você tem uma CodeCatalyst função que permite criar um ambiente de desenvolvimento. A função de administrador do espaço, a função de administrador do projeto e a função de colaborador têm permissão para criar ambientes de desenvolvimento. Para obter mais informações, consulte [Concedendo acesso com funções 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 da sua função no projeto ou no espaço, talvez você não tenha permissões para enviar o código para os repositórios de origem do projeto. A função de administrador do espaço, a função de administrador do projeto e a função de colaborador têm permissão para enviar código aos repositórios do projeto.

Se você tem a função de Colaborador, mas não consegue enviar o código para uma ramificação específica, pode haver uma regra de ramificação configurada para a ramificação específica que impede que os usuários com essa função enviem o código para essa ramificação específica. Tente enviar suas alterações para uma ramificação diferente ou criar uma ramificação e, em seguida, enviar seu código para essa ramificação. Para obter mais informações, consulte [Gerencie 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 CodeCatalyst espaço da Amazon 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 Space para aumentar o nível de cobrança.

Meu ambiente de desenvolvimento desconectado

Problema: Meu ambiente de desenvolvimento foi desconectado enquanto eu o estava usando.

Possíveis correções: Para corrigir o problema, verifique sua conexão com a Internet. Se você não estiver conectado à Internet, conecte-se e continue trabalhando em seu ambiente de desenvolvimento.

Meu ambiente de desenvolvimento conectado ao 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 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 endereços CIDR IPv4, configure um bloco CIDR diferente de 172.17.0.0/16

Note

Você não pode 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 meu ambiente de desenvolvimento via SSH

Para solucionar problemas de conexão com seu ambiente de desenvolvimento via SSH, você pode executar o `ssh` comando com a `-vvv` opção 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 meu 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 `Host codecatalyst-dev-env*` seção estiver desatualizado, você não conseguirá se conectar ao seu ambiente de desenvolvimento via SSH. Para solucionar esse problema, exclua a `Host codecatalyst-dev-env*` seção e execute o primeiro comando do modal SSH Access novamente. Para obter mais informações, consulte [Conectando-se a um ambiente de desenvolvimento usando SSH](#).

Não consigo me conectar ao meu ambiente de desenvolvimento via SSH porque estou tendo problemas com AWS Config o **codecatalyst** meu perfil

Certifique-se de que seu AWS Config (~/.aws/config) para o codecatalyst perfil corresponda ao descrito em [Configurando para usar o AWS CLI com CodeCatalyst](#). Caso contrário, exclua o perfil codecatalyst e execute o primeiro comando do modal SSH Access novamente. Para obter mais informações, consulte [Conectando-se a um ambiente de desenvolvimento usando SSH](#).

Solução de problemas com IDEs

Consulte as seções a seguir para solucionar problemas relacionados aos IDEs em CodeCatalyst. Para obter mais informações sobre IDEs, consulte [Criando um ambiente de desenvolvimento em um IDE](#).

Tópicos

- [Eu tenho versões de imagem de tempo de execução incompatíveis em AWS Cloud9](#)
- [Não consigo acessar meus arquivos no /projects/projects in AWS Cloud9](#)
- [Não consigo iniciar meu ambiente de desenvolvimento AWS Cloud9 usando um devfile personalizado](#)
- [Estou tendo problemas em AWS Cloud9](#)
- [Em JetBrains, não consigo me conectar aos meus ambientes de desenvolvimento por meio de CodeCatalyst](#)
- [Não consigo instalar AWS Toolkit para meu IDE](#)
- [No meu IDE, não consigo iniciar meus ambientes de desenvolvimento](#)

Eu tenho versões de imagem de tempo de execução incompatíveis em AWS Cloud9

AWS Cloud9 está usando versões diferentes do ativo de front-end e da imagem de tempo de execução do back-end. Usar versões diferentes pode fazer com que a extensão Git e o funcionamento da extensão Git sejam AWS Toolkit incorretos. Para corrigir o problema, navegue até o painel do Dev Environment, interrompa o Dev Environment e inicie-o novamente. Para corrigir o problema usando APIs, use a UpdateDevEnvironment API para atualizar o tempo de execução. Para obter mais informações, consulte [UpdateDevEnvironment](#) referência da CodeCatalyst API da Amazon.

Não consigo acessar meus arquivos no `/projects/projects` in AWS Cloud9

O AWS Cloud9 editor não consegue acessar os arquivos no diretório `/projects/projects`. Para corrigir o problema, use o AWS Cloud9 terminal para acessar seus arquivos ou movê-los para um diretório diferente.

Não consigo iniciar meu ambiente de desenvolvimento AWS Cloud9 usando um devfile personalizado

Sua imagem 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 um novo para continuar.

Estou tendo problemas em AWS Cloud9

Para outros problemas, consulte a seção de solução de problemas no [Guia AWS Cloud9 do usuário](#).

Em JetBrains, não consigo me conectar aos meus ambientes de desenvolvimento por meio de CodeCatalyst

Para corrigir o problema, verifique se você tem somente a versão mais recente do JetBrains instalado. 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 JetBrains e registre o manipulador de protocolo novamente.

Não consigo instalar AWS Toolkit para meu IDE

Para corrigir esse problema no VS Code, instale manualmente AWS Toolkit for Visual Studio Code a partir de [GitHub](#).

Para corrigir esse problema JetBrains, instale manualmente a AWS Toolkit for JetBrains a partir de [GitHub](#).

No meu IDE, não consigo iniciar meus ambientes de desenvolvimento

Para corrigir esse problema no VS Code, verifique se você tem a versão mais recente do VS Code e AWS Toolkit for Visual Studio Code está instalada. Se você não tiver a versão mais recente, atualize e inicie seu ambiente de desenvolvimento. Para obter mais informações, consulte [Amazon CodeCatalyst for VS Code](#).

Para corrigir esse problema JetBrains, verifique se você tem a versão mais recente do JetBrains e AWS Toolkit for JetBrains está instalada. Se você não tiver a versão mais recente, atualize e inicie seu ambiente de desenvolvimento. Para obter mais informações, consulte [Amazon CodeCatalyst for JetBrains](#).

Solução de problemas com devfiles

Consulte as seções a seguir para solucionar problemas relacionados aos devfiles em CodeCatalyst. Para obter mais informações sobre devfiles, consulte [Configurando 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 construído em meu ambiente de desenvolvimento com o devfile universal padrão](#)
- [Quero mover um arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento](#)
- [Estou tendo 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 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 registros abaixo `/aws/mde/logs/devfile.log`. Você também pode verificar se `postStart` a execução foi bem-sucedida em seus registros: `/aws/mde/logs/devfileCommand.log`.

Meu projeto não está sendo construído 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 `devfile.yaml` arquivo no repositório de origem do projeto para localizar e corrigir quaisquer erros.

Quero mover um arquivo de desenvolvimento do repositório para um ambiente de desenvolvimento

Você pode mover o devfile padrão para o seu repositório `/projects/devfile.yaml` de código-fonte. Para atualizar a localização do devfile, use o seguinte comando: `/aws/mde/mde start --location repository-name/devfile.yaml`.

Estou tendo 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 seu ambiente e corrigir seu devfile. No modo de recuperação, a execução `/aws/mde/mde status` não conterá a localização do seu devfile.

```
{
  "status": "STABLE"
}
```

Você pode verificar o erro nos registros abaixo `/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 poderá ver um dos seguintes:

- `{"status": "STABLE", "location": "devfile.yaml" }`

Isso indica que seu devfile está correto.

- `{"status": "STABLE" }`

Isso indica que seu devfile não pôde ser iniciado e entrou no modo de recuperação.

Você pode verificar o erro exato nos registros abaixo `/aws/mde/logs/devfile.log`.

Você também pode verificar se `postStart` a execução foi bem-sucedida em seus registros: `/aws/mde/logs/devfileCommand.log`.

Para obter mais informações, consulte [Especificando imagens de devfile universais para um ambiente de desenvolvimento](#).

Meu devfile não é compatível com as ferramentas fornecidas na imagem mais recente

Em seu ambiente de desenvolvimento, `devfile` ou `devfile postStart` pode falhar se as `latest` ferramentas não tiverem as ferramentas necessárias para um projeto específico. Para corrigir o problema, faça o seguinte:

1. Navegue até seu `devfile`.
2. Em seu `devfile`, atualize para uma versão de imagem granular em vez de `latest`. Pode ser semelhante ao seguinte:

```
components:  
  - container:  
      image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Crie um novo ambiente de desenvolvimento usando o `devfile` atualizado.

Solução de problemas com problemas

As informações a seguir podem ajudá-lo a solucionar problemas comuns com problemas em CodeCatalyst.

Tópicos

- [Não consigo escolher um responsável para o meu problema](#)

Não consigo escolher um responsável para o meu problema

Problema: ao criar um problema, a lista de responsáveis fica vazia.

Possíveis correções: a lista de responsáveis está diretamente vinculada aos CodeCatalyst usuários listados como membros do projeto. Para verificar se o acesso ao perfil do usuário está funcionando corretamente, escolha o ícone do perfil e, em seguida, escolha Perfil do usuário. Se as informações do perfil do usuário não forem preenchidas, verifique se há incidentes no relatório de saúde. Se ele for preenchido, registre um tíquete de serviço.

Solução de problemas entre a Amazon CodeCatalyst e os AWS SDKs ou o AWS CLI

As informações a seguir podem ajudá-lo a solucionar problemas comuns ao trabalhar com CodeCatalyst e com os AWS CLI AWS SDKs.

Tópicos

- [Eu recebo uma mensagem de erro quando entro 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 aws codecatalyst em uma linha de comando ou terminal dizendo que é uma escolha inválida

Problema: Quando tento usar o AWS CLI with CodeCatalyst, um ou mais dos aws codecatalyst comandos 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 serviços e comandos mais recentes. Atualize sua instalação do AWS CLI e tente novamente. Para ter 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 CodeCatalyst, recebo uma mensagem informando `You can configure credentials by running "aws configure". ouUnable to locate authorization token.`

Solução: você deve configurar um AWS CLI perfil para trabalhar com CodeCatalyst comandos. Para mais informações, consulte [Configurando para usar o AWS CLI com CodeCatalyst](#).

Entendendo o status atual do serviço com o relatório CodeCatalyst de saúde

O relatório de CodeCatalyst saúde da Amazon é um painel público que fornece aos usuários uma lista agregada de up-to-the-minute notificações sobre desempenho de recursos e disponibilidade de serviços CodeCatalyst que têm um impacto generalizado. Você pode ver quais recursos estão com problemas e podem afetar os aplicativos em CodeCatalyst. Isso permite rastrear interrupções e outros períodos de inatividade de recursos em todo o sistema. Quando ocorre um incidente, um indicador azul aparece no ícone do relatório de saúde. Além disso, envia CodeCatalyst automaticamente um alerta e uma notificação por e-mail a todos os usuários com a função de administrador do Space 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 obter atualizações de até o minuto.

Aqui está um possível fluxo de trabalho para usar o relatório de CodeCatalyst saúde:

Mateo Jackson é desenvolvedor no Budding Space com permissões de administrador do Space. Ao tentar criar uma pull request, ele continua recebendo uma mensagem de erro. Ele verifica seu e-mail e descobre que recebeu um e-mail de incidente do sistema gerado automaticamente CodeCatalyst ao fornecer um histórico detalhado sobre o problema do sistema que afeta seu espaço. Ele escolhe Exibir atualização e é levado ao relatório de CodeCatalyst integridade, onde pode ver todos os incidentes relatados pelo sistema. Ele escolhe o incidente na lista para obter mais informações. É aberta uma tela dividida que fornece um registro de data e hora da 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 ele não tenha acesso ao seu e-mail, ele pode escolher o ícone de sino no painel superior para acessar o relatório de CodeCatalyst saúde.

CodeCatalyst conceitos de relatório de saúde

Aprender os conceitos a seguir ajudará você a entender o relatório de CodeCatalyst integridade e como eles permitem que você acompanhe a integridade de seus aplicativos, serviços e recursos.

O incidente

O incidente é o evento do sistema que está afetando os aplicativos e os recursos internos 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.

Capacidades afetadas

Os recursos afetados são os recursos ou aplicativos afetados pelo incidente. Um único incidente pode afetar várias áreas do sistema, incluindo pull requests, problemas, fluxos de trabalho, teste, implantação e origem.

Atualizado em

Atualizado em fornece um registro de data e hora da última atualização do incidente.

AWS Support para Amazon CodeCatalyst

Ao criar um espaço, você deve conectar uma Conta da AWS e designá-la como a conta de cobrança do seu espaço. A conta da AWS que você designa como sua conta de cobrança também é onde você acessa seu AWS Support plano da Amazon CodeCatalyst. Se precisar de suporte, você pode criar casos de suporte a partir dessa opção designada Conta da AWS.

CodeCatalyst usuários em um espaço usam a CodeCatalyst página AWS Support for Amazon CodeCatalyst para gerenciar casos de suporte. Você pode fazer o upgrade para um AWS Support plano, como Business Support ou Enterprise Support, para criar e gerenciar casos de suporte CodeCatalyst técnico no CodeCatalyst. O suporte está disponível por telefone, web ou chat para casos de suporte.

Somente casos específicos do CodeCatalyst serviço e dos recursos podem ser suportados AWS Support pela Amazon CodeCatalyst. CodeCatalyst os recursos incluem recursos implantados dentro CodeCatalyst e por usuários em CodeCatalyst, mas não incluem recursos implantados para outros serviços AWS ou serviços de terceiros. Se precisar de suporte para qualquer outro AWS serviço, você deve abri-lo por meio do AWS Management Console.

Para alterar seu plano de suporte, consulte [Alteração de planos de suporte](#).

Note

Os planos do Developer Support não foram projetados para ambientes de produção. Se uma conta de cobrança de espaço tiver um plano de Developer Support, esse plano não será aplicado em cascata a todos os administradores de espaço e membros do espaço dentro dela. AWS Support CodeCatalyst


Cobrança AWS Support para a Amazon CodeCatalyst

Quando você cria um espaço no CodeCatalyst, os usuários no espaço podem criar e gerenciar casos de AWS Support suporte da Amazon CodeCatalyst. Você pode criar dois tipos de casos de clientes:

- Os casos de suporte de conta e cobrança estão disponíveis para todos os CodeCatalyst usuários no espaço. Você pode obter ajuda com perguntas sobre faturamento e conta com base nas suas permissões em CodeCatalyst.


- Os casos de suporte técnico conectam você a um engenheiro de suporte técnico para obter ajuda com problemas técnicos relacionados ao serviço e extensões de aplicativos de terceiros. Se você tiver o suporte Básico, não poderá criar um caso de suporte técnico.

A conta Conta da AWS designada como conta de cobrança do espaço deve ter um plano de Business Support ou Enterprise Support para que o espaço seja usado AWS Support em casos técnicos. CodeCatalyst

 Note

Se seu espaço for usado AWS Support para a Amazon CodeCatalyst a partir de uma conta que não tem um plano Business Support ou Enterprise Support, você ainda pode usar AWS Support para a Amazon CodeCatalyst para casos de conta e cobrança.

Para obter suporte técnico, você deve abrir todas as caixas pelo CodeCatalyst console. Você não pode criar casos de suporte técnico para a CodeCatalyst partir [AWS Support](#) do AWS Management Console.

 Note

As solicitações de aumento do limite de AWS Support serviço não estão disponíveis na Amazon CodeCatalyst. Essas solicitações só podem ser enviadas pelo usuário raiz da conta de faturamento espacial no AWS Support Center Console.

AWS Support for Amazon CodeCatalyst tem os mesmos contratos de suporte que AWS Support, com as seguintes considerações:

- Listas de severidade, tempos de resposta e SLAs AWS Support se aplicam aos casos de suporte em AWS Support For CodeCatalyst, conforme detalhado em [Escolha de uma severidade](#).
- Administradores e membros do espaço não podem usar as AWS Support APIs, o AWS SDK ou o AWS Support aplicativo no Slack para criar casos para CodeCatalyst. CodeCatalyst casos de suporte só podem ser enviados de CodeCatalyst.

Note

CodeCatalyst não está totalmente integrado com AWS Trusted Advisor nossa Detecção e Resposta a AWS Incidentes. Valide como CodeCatalyst está integrado para garantir que suas práticas comerciais estejam alinhadas com a integração atual.

Você deve ser um usuário no espaço em que deseja solicitar suporte.

Note

Se você tiver mais de um construtor em seu espaço, recomendamos que você compre 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 cobrança do espaço usa a `AWSRoleForCodeCatalystSupport` função e a política [AmazonCodeCatalystSupportAccess](#) gerenciada. Isso permite que CodeCatalyst os usuários em um espaço acessem AWS Support a CodeCatalyst página da Amazon. Para obter mais informações sobre essa função e política, consulte [AmazonCodeCatalystSupportAccess](#). Para outras considerações sobre faturamento, consulte [Gerenciando o faturamento no Guia do Administrador](#) da Amazon CodeCatalyst .

Aqui está um fluxo possível para um construtor criar um caso de suporte em CodeCatalyst:

Mateo Jackson é desenvolvedor de um projeto em CodeCatalyst. Depois de se inscrever no plano Conta da AWS que gerencia o faturamento da Amazon CodeCatalyst e fazer o upgrade para um plano de Business Support, todos os criadores do setor podem criar casos de suporte técnico. AWS Support Mateo envia um caso de suporte técnico para uma falha no fluxo de trabalho em seu projeto. Mateo usa a CodeCatalyst página AWS Support para a Amazon para preencher o formulário e criar um caso, fornecendo a 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 Conta da AWS designada como conta de cobrança e associada ao plano de suporte do espaço.

Embora todos os criadores possam criar casos de suporte no AWS Support for CodeCatalyst, você não será cobrado por cada caso criado. Você pode abrir casos e contatos praticamente ilimitados com base no plano AWS Support Premium adquirido em sua conta de cobrança de espaço.

Note

A conta de faturamento de espaço é aquela em Conta da AWS que você é cobrado por CodeCatalyst usuários e recursos. Se você implantou outras Contas da AWS, entre em contato AWS Support por meio do AWS Management Console para obter ajuda com os recursos implantados em outros serviços.

Você pode identificar o Conta da AWS que você implantou a partir do fluxo de trabalho.

Configurando seu espaço AWS Support para a Amazon CodeCatalyst

AWS Support for Amazon CodeCatalyst gerencia casos de suporte como parte da integração da AWS Support API com CodeCatalyst.

A `AWSRoleForCodeCatalystSupport` função é uma função de serviço usada para casos de suporte em seu espaço. A função deve ser adicionada à conta de cobrança designada para o espaço. Para obter mais informações ou para criar a função, consulte [Criando a AWSRoleForCodeCatalystSupport função para sua conta e espaço](#).

Note

Para um espaço criado antes de 20 de abril de 2023, você deve criar a função para que o suporte funcione em seu espaço. CodeCatalyst Se criar um espaço após 20 de abril de 2023, você poderá criar a função durante a criação do espaço, na página de detalhes de cobrança ou clicando no link do banner de suporte em CodeCatalyst. CodeCatalyst

Para configurar o suporte para seu espaço

1. Ao criar um CodeCatalyst espaço, você é instruído a conectar uma conta de cobrança. A conta de cobrança designada para o espaço será cobrada por. AWS Para obter mais informações sobre como criar um espaço, consulte [Criando sua primeira função espacial e de desenvolvimento \(começando sem um convite\)](#).
2. Quando você cria um CodeCatalyst espaço, a opção está disponível para criar a função de `AWSRoleForCodeCatalystSupport` serviço que permite que CodeCatalyst os usuários acessem o suporte. A função usa a política

gerenciada `AmazonCodeCatalystSupportAccess`. A função deve ser adicionada à conta de cobrança Conta da AWS designada como a do espaço. Para obter mais informações sobre a criação da função, consulte [Criando a `AWSRoleForCodeCatalystSupport` função para sua conta e espaço](#).

3. Para a conta de cobrança designada para o espaço, recomenda-se que o administrador do espaço adquira um plano Business Support ou Enterprise Support para o. Conta da AWS Todos os membros do espaço poderão gerenciar casos de suporte da Amazon CodeCatalyst, e AWS Support os canais de suporte serão alinhados ao AWS Support plano que você adquiriu quando as integrações forem concluídas.
4. Para criar e gerenciar casos de suporte em CodeCatalyst, consulte [Criando um caso de CodeCatalyst suporte em CodeCatalyst](#).

Acessando o suporte para CodeCatalyst no AWS Management Console

Se a conta de cobrança habilitada para suporte de um espaço for desconectada, AWS Support os casos associados à conta de cobrança de espaço anterior e ao plano de suporte associado não estarão mais visíveis na Amazon. AWS Support CodeCatalyst O usuário raiz dessa conta de faturamento pode visualizar e resolver casos antigos a partir do AWS Management Console e pode configurar permissões do IAM AWS Support para que outros usuários visualizem e resolvam casos antigos. Você ainda poderá aproveitar os benefícios do seu plano de suporte AWS Management Console para todos os outros Serviços da AWS e concluir quaisquer casos de CodeCatalyst suporte que não tenham sido resolvidos anteriormente.

Para obter mais informações, consulte [Atualizar, resolver e reabrir seu caso no Guia](#) do AWS Support usuário.

Support cases com informações gerais sobre como fazer também CodeCatalyst podem ser abertos no AWS Management Console, mas nenhum suporte técnico pode ser recebido por meio desse canal para. CodeCatalyst Para obter mais informações, consulte [Criação de casos de suporte e gerenciamento de casos](#) no Guia AWS Support do usuário.

Aqui está um fluxo possível para um usuário resolver um caso de suporte CodeCatalyst no AWS Management Console:

Embora todos os criadores possam criar casos de suporte com AWS Support a Amazon CodeCatalyst, as solicitações de suporte são cobradas da conta designada como a conta de

cobrança do espaço. Mateo Jackson é desenvolvedor de um projeto em CodeCatalyst que abriu um caso de suporte técnico para uma falha no fluxo de trabalho em seu projeto. No entanto, a conta de cobrança do espaço que foi cadastrado na Amazon CodeCatalyst e comprou um plano Business Support foi desconectada do espaço. AWS Support A única maneira de Mateo visualizar a comunicação mais recente e resolver os casos abertos CodeCatalyst é gerenciar a ID do caso a partir do AWS Support Centro noAWS Management Console. Para fazer isso, Mateo recebe permissões de IAM do usuário raiz da conta de faturamento de espaço anterior anexada ao caso de suporte e resolve o AWS Support caso por meio do console.

Important

Se você alterar a conta de cobrança designada para seu espaço, seu AWS Support plano ainda estará acessível até o final do mês por meio do AWS Management Console único. Você precisará recomprar AWS Support na conta de cobrança atualizada para continuar acessando seus casos de suporte criados anteriormente em. CodeCatalyst Recomendamos esperar até que você tenha resolvido todos os seus casos de suporte para alterar as contas de faturamento de espaço para evitar qualquer impacto no acesso aos seus casos de suporte AWS Support por meio da Amazon CodeCatalyst.

Criando um caso de CodeCatalyst suporte em CodeCatalyst

Você pode criar um caso de suporte na CodeCatalyst página AWS Support para a Amazon.

Um AWS Builder ID só pode obter suporte para o alias com o qual está autenticado e somente para recursos com base em suas permissões. As opções de conta e cobrança estão disponíveis para todos os administradores e membros do espaço. No entanto, os usuários só podem obter suporte para os recursos aos quais têm acesso CodeCatalyst e não em relação ao gerenciamento do faturamento da conta.

Você pode criar um caso de conta e cobrança ou um caso de suporte técnico para seus CodeCatalyst recursos usando a CodeCatalyst página AWS Support for do seu espaço.

Note

Somente casos específicos do CodeCatalyst serviço e dos recursos podem ser suportados AWS Support pela Amazon CodeCatalyst. CodeCatalyst os recursos incluem recursos implantados dentro CodeCatalyst e por usuários em CodeCatalyst, mas não incluem

recursos implantados para outros serviços AWS ou serviços de terceiros. Se precisar de suporte para qualquer outro AWS serviço, você deve abri-lo por meio do AWS Management Console.


Para criar um caso de suporte em CodeCatalyst

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

 Tip

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

3. Na parte superior da página, escolha a opção ? ícone e, em seguida, escolha Support.
4. Escolha Criar caso.
5. Escolha uma das seguintes opções:
 - Conta e faturamento
 - Técnico


 Note

No AWS Support caso da Amazon CodeCatalyst, se um plano Business Support ou Enterprise Support for adicionado à conta de cobrança do espaço, o suporte CodeCatalyst técnico do caso 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](#).

AWS Supportos planos não se estendem por espaços. Se você for membro de vários espaços, o administrador do espaço precisará comprar um plano AWS Support 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 obter informações sobre como escolher uma severidade, consulte [Escolha de uma severidade](#).
 - Orientações gerais

- Sistema prejudicado
 - Sistema de produção prejudicado
 - Sistema de produção desativado
 - Sistema crítico para os negócios inativo
7. Selecione Next step: Additional information (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 CodeCatalyst, como ID do fluxo de trabalho, registros ou capturas de tela
 - Mensagens de erro recebidas
 - Etapas de resolução de problemas que você seguiu

 Note

Não compartilhe nenhuma informação confidencial em caso de correspondência, como credenciais, cartões 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 à sua ID do AWS Construtor é preenchido automaticamente.
13. (Opcional) Escolha o projeto em Nome do projeto (se aplicável).

 Note

Você só verá projetos para os quais você tem permissão. Se você precisar acessar outro projeto, peça ao administrador do projeto que lhe forneça acesso antes de criar um caso de suporte.

14. Escolha a próxima etapa: Entre em contato conosco.

15. Em Idioma de contato preferencial, escolha o padrão. Somente o inglês está disponível no momento.
16. Escolha a opção Web, Telefone ou Chat para o método de contato.
17. Revise os detalhes do seu 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 em seu caso de suporte. Não há como omitir um caso de suporte de usuários individuais no momento.

Resolvendo um caso de suporte em CodeCatalyst

Você pode resolver casos de suporte abertos na CodeCatalyst página AWS Support para a Amazon.

Você deve ter uma função de administrador do Space ou membro do Space no espaço em que deseja resolver um caso de suporte. Se você não tiver a função de administrador do Space 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 CodeCatalyst console em <https://codecatalyst.aws/>.
2. Navegue até seu CodeCatalyst espaço.

Tip

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

3. Na parte superior da página, escolha a opção? ícone e, em seguida, escolha AWS Support para Amazon CodeCatalyst.
4. Escolha o link para o caso de suporte que você deseja gerenciar. Escolha Resolve case (Resolver caso).

Reabrindo um caso de suporte em CodeCatalyst

Você pode usar para reabrir um caso de suporte resolvido na CodeCatalyst página AWS Support para a Amazon.

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

Tip

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

3. Na parte superior da página, escolha a opção? ícone e, em seguida, escolha AWS Support for CodeCatalyst.
4. Escolha o link para o caso de suporte que você deseja gerenciar. Selecione Reopen (Reabrir). Escolha OK na tela de confirmação e, em seguida, escolha Enviar.
5. Preencha a Descrição com as informações mais recentes sobre o mesmo problema. Não compartilhe nenhuma informação confidencial em caso de correspondência, como credenciais, cartões de crédito, URLs assinados ou informações de identificação pessoal.

Cotas para CodeCatalyst

A tabela a seguir descreve as cotas e os limites da Amazon CodeCatalyst. Você pode encontrar informações adicionais sobre aspectos específicos CodeCatalyst nos seguintes tópicos:

- [Cotas para repositórios de origem em CodeCatalyst](#)
- [Cotas para identidade, permissão e acesso em CodeCatalyst](#)
- [Cotas para fluxos de trabalho](#)
- [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 espaços 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 cobrança de um espaço | 1 |
| 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

As descrições do espaço são opcionais. Se especificados, eles devem ter entre 0 e 200 caracteres. Eles podem conter qualquer combinação de letras, números, espaços, pontos, sublinhados, vírgulas, traços e os seguintes caracteres especiais:

? & \$ % + = / \ ; : \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 devem ter entre 3 e 63 caracteres. Eles também devem começar com um caractere alfanumérico. Os nomes de espaço podem conter qualquer combinação de letras, números, pontos, sublinhados e traços. Eles não podem conter nenhum dos seguintes caracteres:

! ? @ # \$ % ^ & * () + = { } []
| \ > < ~ ` ' " ; :

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 |
|--|---|--------------------|
| Novo conteúdo: etapas adicionadas para trabalhar com conexões pessoais | 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 |
| Conteúdo atualizado: use repositórios de terceiros com plantas | Documentação atualizada para a capacidade de criar um repositório GitHub ou do Bitbucket ao Criando um projeto com um blueprint criar um blueprint personalizado . Aplicando um blueprint em um projeto para adicionar recursos | 5 de junho de 2024 |
| Novo conteúdo: extensão de repositórios Bitbucket | Novo conteúdo adicionado para usar a extensão de repositórios Bitbucket em. CodeCatalyst | 5 de junho de 2024 |
| Novo conteúdo: Tipos de ação | O Ações de terceiros tópico foi atualizado para mencionar a ação SonarCloud Escanear. | 29 de maio de 2024 |
| Conteúdo atualizado: Definição YAML da ação "AWS CDK implantar" | Corrigido o CdkRootRotPath exemplo. | 28 de maio de 2024 |

| | | |
|--|--|---------------------|
| Conteúdo atualizado | Títulos de tópicos atualizados e conteúdo reorganizado para melhorar a legibilidade e a descoberta. Se você quiser fornecer feedback sobre essas mudanças, use este link Forneça feedback . | 17 de maio de 2024 |
| Novo conteúdo: visualização do histórico de alterações em um arquivo | 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 |
| Conteúdo atualizado: Tutorial: usando recursos generativos de IA | O tutorial foi atualizado para refletir a integração com o Amazon Q Developer. | 29 de abril de 2024 |
| Conteúdo atualizado: Tutorial: usando recursos generativos de IA | O tutorial foi atualizado para refletir a possibilidade de a Amazon Q analisar problemas em busca de complexidade, sugerir e criar tarefas e trabalhar em tarefas em um problema. | 22 de abril de 2024 |
| Novo conteúdo: Bloqueando a execução de um fluxo de trabalho | Foram adicionados Bloqueando a execução de um fluxo de trabalho , Exigindo aprovações em execuções de fluxo de trabalho , e vários outros tópicos relacionados às aprovações do fluxo de trabalho. | 22 de abril de 2024 |

[Novo conteúdo: Tutorial: Criação de um aplicativo completo com plantas de PDK compostas](#)

Foi adicionado um novo tutorial para usar os blueprints do AWS Project Development Kit (AWS PDK) em um projeto da Amazon CodeCatalyst .

9 de abril de 2024

[Novo conteúdo: usando tarefas para dividir problemas em objetivos menores](#)

Conteúdo adicionado para apoiar o lançamento de tarefas em edições. 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: Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#)

O [Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos](#) tópico 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: Definição YAML da ação “AWS CDK implantar” | Adicionou uma nova <code>CloudAssemblyRootPath</code> propriedade ao Definição YAML da ação “AWS CDK implantar” . | 1º de abril de 2024 |
| Conteúdo atualizado: Especificando imagens do Docker do ambiente de execução | O Especificando imagens do Docker do ambiente de execução tópico foi atualizado para incluir informações sobre a nova imagem do ambiente de execução de março de 2024. | 26 de março de 2024 |
| Conteúdo atualizado: Trabalhando com funções | Informações consolidadas de permissão de função em uma única tabela. A tabela está em um novo Visualizando as permissões disponíveis para cada função tópico. | 18 de março de 2024 |
| Novo conteúdo: Veja todos os espaços e projetos de um usuário | 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 Veja todos os espaços e projetos de um usuário . | 18 de março de 2024 |

| | | |
|--|--|-------------------------|
| Novo conteúdo: Exemplo: um gatilho com um puxão e galhos | Foi adicionado um exemplo de gatilho de pull request. Fiz pequenas correções em todo o Iniciando um fluxo de trabalho executado automaticamente com gatilhos tópico. | 11 de março de 2024 |
| Conteúdo atualizado: Trabalhando com funções | A documentação das funções foi atualizada para incluir permissões para criar, excluir e visualizar ambientes. | 4 de março de 2024 |
| Conteúdo atualizado: Tutorial: usando recursos generativos de IA | O tutorial foi atualizado para refletir as mudanças ao criar e atribuir problemas ao Amazon Q. | 4 de março de 2024 |
| Novo conteúdo: componentes de problemas | Foi adicionado novo conteúdo sobre como trabalhar com componentes de problemas como desenvolvedor de esquemas personalizados. | 27 de fevereiro de 2024 |
| Conteúdo atualizado: Tipos de ação | O CodeCatalyst Ações do Labs tópico foi atualizado para incluir uma lista de ações do CodeCatalyst Labs. | 21 de fevereiro de 2024 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação foi atualizada para refletir a nova funcionalidade com regras de aprovação e requisitos primordiais para mesclar uma pull request. | 15 de fevereiro de 2024 |

| | | |
|--|--|-------------------------|
| Conteúdo atualizado: <u>mesclagem de pull requests</u> | Foi adicionada documentação para pull requests para incluir informações sobre a substituição dos requisitos de mesclagem para mesclar uma pull request que ainda não recebeu aprovações dos revisores necessários ou atendeu às regras de aprovação. | 15 de fevereiro de 2024 |
| Novo conteúdo: <u>Gerenciar regras de aprovação</u> | Foi adicionada documentação para pull requests para incluir informações sobre a criação e o gerenciamento de regras de aprovação. | 15 de fevereiro de 2024 |
| Conteúdo atualizado: <u>Trabalhando com funções</u> | A documentação das funções foi atualizada para incluir permissões para trabalhar com regras de aprovação e pull requests. | 14 de fevereiro de 2024 |
| Conteúdo atualizado: <u>Como faço para corrigir os erros “A definição do fluxo de trabalho não tem erros”?</u> | A Como faço para corrigir os erros “A definição do fluxo de trabalho não tem erros”? seção foi atualizada para incluir mais dicas de solução de problemas. | 9 de fevereiro de 2024 |
| Novo conteúdo: <u>Visualizando o status do fluxo de trabalho</u> | Foi adicionada uma seção que descreve os estados do fluxo de trabalho. | 9 de fevereiro de 2024 |
| Novo conteúdo: <u>Visualizando o status do fluxo de trabalho</u> | Foi adicionada uma seção que descreve os estados do fluxo de trabalho. | 9 de fevereiro de 2024 |

| | | |
|---|---|------------------------|
| Atualize o conteúdo: Cotas para fluxos de trabalho | O Cotas para fluxos de trabalho tópico foi atualizado com o Número máximo de ações por fluxo de trabalho e o Número máximo de ambientes associados a cotas Conta da AWS por espaço. | 7 de fevereiro de 2024 |
| Conteúdo atualizado: Criar um ambiente | A Criar um ambiente seção foi atualizada para indicar que você pode usar no máximo uma conexão de conta por ambiente. | 31 de janeiro de 2024 |
| Novo conteúdo: repositório de plantas GitHub personalizadas | Novo conteúdo adicionado ao GitHub repositório que é disponibilizado publicamente. | 10 de janeiro de 2024 |
| Conteúdo atualizado: Configurando o npm com CodeCatalyst | 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 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação foi atualizada para refletir a nova funcionalidade com os recursos generativos de IA em CodeCatalyst. | 28 de novembro de 2023 |
| Conteúdo atualizado: Criação de um problema | A documentação foi atualizada para refletir a nova funcionalidade com os recursos generativos de IA em CodeCatalyst. | 28 de novembro de 2023 |

| | | |
|---|---|------------------------|
| Novo conteúdo: Tutorial: Usando recursos generativos de IA | Foi adicionado um tutorial para usar os recursos generativos de IA na Amazon CodeCatalyst. | 28 de novembro de 2023 |
| Novo conteúdo: esquemas personalizados e gerenciamento do ciclo de vida | 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 |
| Conteúdo atualizado: Tutorial: Criando um projeto com o blueprint de aplicativo web moderno de três camadas | Atualizou o tutorial com correções e informações sobre solução de problemas. | 22 de novembro de 2023 |
| Conteúdo atualizado: Iniciando um fluxo de trabalho executado automaticamente com gatilhos | Alguns exemplos e descrições relacionados aos acionadores de pull request foram corrigidos. Adicionou uma Ação considerações ao ramificar seção. | 22 de novembro de 2023 |
| Novo conteúdo: Faça 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 Configure e faça login em CodeCatalyst e Faça login com SSO . | 17 de novembro de 2023 |

| | | |
|---|---|------------------------|
| Conteúdo atualizado: Trabalhando com funções | A documentação das funções foi atualizada para incluir permissões para trabalhar com equipes, conexões de VPC, login único e recursos de máquina. | 16 de novembro de 2023 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação foi atualizada para refletir as mudanças na forma como as alterações de uma pull request são exibidas. | 16 de novembro de 2023 |
| Conteúdo atualizado: Cotas para CodeCatalyst | O Cotas para CodeCatalyst tópico foi atualizado com o Número máximo de conexões VPC para uma cota de espaço. | 16 de novembro de 2023 |
| Novo conteúdo: Gerenciamento de equipes para um espaço e para CodeCatalyst projetos | Foram adicionadas informações sobre o uso de equipes com espaços. Consulte Permitindo o acesso ao espaço usando equipes e Permitindo o acesso ao projeto usando equipes . | 16 de novembro de 2023 |
| Novo conteúdo: Gerenciamento dos recursos de máquina para plantas e fluxos de trabalho em um espaço | Foram adicionadas informações sobre o uso de recursos da máquina com espaços. Consulte Permitindo o acesso ao espaço para os recursos da máquina . | 16 de novembro de 2023 |

| | | |
|---|---|------------------------|
| Novo conteúdo: Gerenciamento de recursos de máquina para esquemas e fluxos de trabalho em um projeto CodeCatalyst | Foram adicionadas informações sobre o uso de recursos da máquina com CodeCatalyst projetos. Consulte Permitindo o acesso ao projeto para recursos da máquina . | 16 de novembro de 2023 |
| Novo conteúdo: associando uma conexão VPC a um ambiente | Foi adicionada documentação para associar uma conexão VPC a um ambiente, que pode ser usada em um fluxo de trabalho. | 16 de novembro de 2023 |
| Novo conteúdo: associando uma conexão VPC a um ambiente de desenvolvimento | Foi adicionada documentação para usar ambientes de desenvolvimento com uma conexão VPC. | 16 de novembro de 2023 |
| Novo conteúdo | Publicação inicial do Amazon CodeCatalyst Administrator Guide . | 16 de novembro de 2023 |
| Novo conteúdo: Definição YAML da ação “AWS CDK implantar” | Adicionou uma nova propriedade <code>CdkCliVersion</code> de ao Definição YAML da ação “AWS CDK implantar” e a definição YAML da ação “AWS CDK bootstrap” . | 14 de novembro de 2023 |
| Conteúdo atualizado: Trabalhando com funções | A documentação das funções foi atualizada para incluir permissões para trabalhar com regras de ramificação. | 13 de novembro de 2023 |

| | | |
|---|--|------------------------|
| Conteúdo atualizado: Solução de problemas com repositórios de origem, fluxos de trabalho e ambientes de desenvolvimento | Os tópicos de solução de problemas foram atualizados para incluir informações sobre como trabalhar com regras de ramificação. | 13 de novembro de 2023 |
| Conteúdo atualizado: Crie e teste a definição YAML da ação | Atualizou a documentação da <code>Environment</code> propriedade. Agora é um campo opcional para ações de criação e teste. | 13 de novembro de 2023 |
| Novo conteúdo: Gerenciar regras de ramificação | Foi adicionada documentação para ramificações para incluir informações sobre a visualização de quaisquer regras para ramificações em um repositório de origem e a criação e gerenciamento de regras de ramificações. | 13 de novembro de 2023 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação foi atualizada para refletir as mudanças na forma como as informações sobre uma pull request são exibidas. | 10 de novembro de 2023 |
| Conteúdo atualizado: Armazenando arquivos em cache entre execuções de fluxo de trabalho | A documentação foi atualizada para incluir limitações de armazenamento em cache de arquivos. | 10 de novembro de 2023 |
| Conteúdo atualizado: Tutorial: Implantar um aplicativo no Amazon EKS | A documentação foi atualizada para mencionar o plano de implantação de aplicativos EKS. | 9 de novembro de 2023 |

| | | |
|--|---|-------------------------|
| Novo conteúdo: Pacotes em CodeCatalyst | Documentação adicionada para usar pacotes em CodeCatalyst. | 1.º de novembro de 2023 |
| Conteúdo novo e atualizado: Trabalhando com funções | Atualizou a documentação para quatro novas funções em CodeCatalyst: usuário avançado, acesso limitado, revisor e somente leitura. | 1.º de novembro de 2023 |
| Conteúdo atualizado: Exportar um parâmetro GitHub de saída para que outras ações possam usá-lo | Os exemplos foram atualizados para usar o arquivo de GITHUB_OUTPUT ambiente em vez do set-output comando. Usar arquivos de ambiente GitHub é o método recomendado para definir parâmetros de saída. | 24 de outubro de 2023 |
| Novo conteúdo: Iniciando um fluxo de trabalho executado automaticamente com gatilhos | Documentação adicionada para acionadores de agendamento. | 16 de outubro de 2023 |
| Conteúdo atualizado: Definição YAML da ação “Implantar no cluster Kubernetes” | Foram adicionadas informações sobre o uso da CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> função aos Tutorial: Implantar um aplicativo no Amazon EKS tópicos Definição YAML da ação “Implantar no cluster Kubernetes” e. | 22 de setembro de 2023 |

[Conteúdo atualizado: novo nome da função e política para a CodeCatalystWorkflowDevelopmentRole-*spaceName* função](#)

As etapas e as descrições das funções foram atualizadas para a alteração do nome da função de desenvolvedor para CodeCatalystWorkflowDevelopmentRole-*spaceName* . A função de desenvolvedor agora usa a política AdministratorAccess AWS gerenciada. Consulte [Entendendo a função CodeCatalystWorkflowDevelopmentRole-*spaceName* de serviço e Criando sua primeira função espacial e de desenvolvimento \(começando sem um convite\)](#).

20 de setembro de 2023

[Conteúdo atualizado: Configurando e usando variáveis em um fluxo de trabalho](#)

Introduziu dois novos conceitos: variáveis definidas pelo usuário e variáveis predefinidas. Esses conceitos devem facilitar a leitura e a compreensão da [Configurando e usando variáveis em um fluxo de trabalho](#) seção.

19 de setembro de 2023

[Conteúdo atualizado: Trabalhando com commits](#)

A documentação foi atualizada para refletir a alteração nas informações exibidas e fornecer detalhes sobre a visualização de compromissos com vários pais.

7 de setembro de 2023

| | | |
|--|---|-----------------------|
| Novo conteúdo: Iniciando um fluxo de trabalho executado automaticamente com gatilhos | O exemplo a seguir foi adicionado ao Iniciando um fluxo de trabalho executado automaticamente com gatilhos tópico: Exemplo: um simples gatilho “push to main” | 6 de setembro de 2023 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação foi atualizada para refletir a alteração na ordem de exibição da ramificação de origem e da ramificação de destino ao criar uma pull request. | 30 de agosto de 2023 |
| Novo conteúdo: visualize e altere a ramificação padrão | Foi adicionada documentação para ramificações para incluir informações sobre como visualizar e alterar a ramificação padrão de um repositório de origem. | 30 de agosto de 2023 |
| Conteúdo atualizado: Definição YAML da ação “Implantar no cluster Kubernetes” | Foi adicionada uma nota sobre Helm e Kustomize à descrição da Manifests propriedade no. Definição YAML da ação “Implantar no cluster Kubernetes” | 15 de agosto de 2023 |
| Novo conteúdo: Gerenciamento de anexos de problemas | Foi adicionada documentação para trabalhar e gerenciar anexos sobre problemas. | 15 de agosto de 2023 |
| Conteúdo atualizado: Iniciando um fluxo de trabalho executado automaticamente com gatilhos | Melhorou e expandiu a documentação relacionada aos acionadores do fluxo de trabalho. | 11 de agosto de 2023 |

[Novo conteúdo: Solução de problemas de permissões de função](#)

Foram adicionadas informações sobre a atualização das permissões da função para executar um fluxo de trabalho que requer acesso à Amazon CodeGuru. Consulte [O fluxo de trabalho moderno de três camadas do blueprint de aplicativos web OnPullRequest falha com erro de permissões para a Amazon CodeGuru](#).

11 de agosto de 2023

[Novo conteúdo: Como fazer para corrigir as mensagens “O fluxo de trabalho está inativo”?](#)

Foi adicionado o seguinte tópico de solução de problemas: [Como fazer para corrigir as mensagens “O fluxo de trabalho está inativo”?](#)

11 de agosto de 2023

[Novo conteúdo: Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#)

Foi adicionada documentação para a ação de cluster Deploy to Kubernetes. Para obter mais informações, consulte [Implantação de um aplicativo no Amazon Elastic Kubernetes Service com um fluxo de trabalho](#) e [Tutorial: Implantar um aplicativo 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-event-logs` comando. Consulte [Monitoramento de eventos e chamadas de API usando o registro](#).

20 de julho de 2023

[Conteúdo atualizado: Iniciando um fluxo de trabalho executado automaticamente com 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

[Conteúdo atualizado: Cotas para fluxos de trabalho](#)

Atualizou o [Cotas para fluxos de trabalho](#) tópico com a cota de tempo máximo em que uma ação pode ser executada.

27 de junho de 2023

[Conteúdo atualizado: Definição de fluxo de trabalho YAML](#)

Corrigido um erro de formatação no bloco `Compute` de código.

27 de junho de 2023

| | | |
|--|---|---------------------|
| Conteúdo atualizado: <u>Proteção de dados</u> | A documentação foi atualizada para incluir informações adicionais sobre a replicação de dados. | 26 de junho de 2023 |
| Novo conteúdo: <u>Especificar a versão principal, secundária ou de patch de uma ação</u> | Adicionou um Especificar a versão principal, secundária ou de patch de uma ação tópico. | 21 de junho de 2023 |
| Conteúdo atualizado: <u>Implantação em Contas da AWS e VPCs com ambientes CodeCatalyst</u> | Esclareceu a Quais ações apoiam os ambientes? seção. | 14 de junho de 2023 |
| Conteúdo atualizado: <u>documentação de problemas reorganizada</u> | Reorganizou a 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 |
| Conteúdo atualizado: <u>Alternador de visualização de problemas</u> | Vários fluxos de usuários foram atualizados para se alinharem ao alternador de visualização de problemas atualizado. | 31 de maio de 2023 |
| Conteúdo atualizado: <u>gerenciamento de notificações</u> | A documentação das notificações foi atualizada para incluir informações sobre a configuração de notificações pessoais do Slack. | 30 de maio de 2023 |
| Conteúdo atualizado: <u>gerenciamento de notificações</u> | A documentação das notificações foi atualizada para incluir informações sobre a configuração de notificações pessoais do Slack. | 30 de maio de 2023 |

| | | |
|---|---|--------------------|
| Novo conteúdo: modelo de CodeCatalyst confiança | 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 Entendendo o modelo de CodeCatalyst confiança . | 20 de maio de 2023 |
| Conteúdo atualizado: Conectando um fluxo de trabalho a um repositório de origem | Simplificou as instruções em Fazendo referência a arquivos em um repositório de origem . | 10 de maio de 2023 |
| Conteúdo atualizado: Cotas para fluxos de trabalho | O Cotas para fluxos de trabalho tópico foi atualizado com o tamanho máximo de uma cota de valor variável de saída. | 10 de maio de 2023 |
| Novo conteúdo: Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos | Foram adicionados dois exemplos: Exemplo: referenciando um arquivo em um único artefato Exemplo: referenciar um arquivo em um artefato quando um está presente WorkflowSource e. | 10 de maio de 2023 |

| | | |
|--|--|---------------------|
| Conteúdo atualizado: trabalhando com pull requests | A documentação de pull requests foi atualizada para incluir informações sobre a configuração de preferências de e-mail para eventos de pull request. | 21 de abril de 2023 |
| Conteúdo atualizado: gerenciamento de notificações | A documentação das notificações foi atualizada para incluir informações sobre a configuração das preferências de e-mail para eventos de pull request. | 21 de abril de 2023 |
| Atualizações das políticas gerenciadas | Foram adicionadas as políticas AWSPolítica gerenciada da : AmazonCodeCatalystFullAccessAWSPolítica gerenciada da : AmazonCodeCatalystReadOnlyAccess , e AWSPolítica gerenciada da : AmazonCodeCatalystSupportAccess gerenciadas. Consulte Atualizações do CodeCatalyst para políticas gerenciadas pela AWS . | 20 de abril de 2023 |
| Novo conteúdo: Removendo um alvo de implantação | Adicionou um Removendo um alvo de implantação tópico. | 20 de abril de 2023 |
| Novo conteúdo: Tipos de ação | Adicionou um CodeCatalyst ações tópico. | 20 de abril de 2023 |

| | | |
|--|--|---------------------|
| Atualizações para gerenciar um usuário com a função de administrador de espaço em um espaço | Foram adicionadas informações sobre como remover ou alterar a função de um usuário com a função de administrador de espaço em um espaço. Consulte Removendo ou alterando a função de um usuário com a função de administrador do Space . | 19 de abril de 2023 |
| Atualizações para administrar ambientes de desenvolvimento | Foram adicionadas informações sobre a administração de ambientes de desenvolvimento como administrador do Space. Consulte Administrando ambientes de desenvolvimento para um espaço . | 19 de abril de 2023 |
| Conteúdo atualizado: problemas de localização e visualização | Reorganizou o Encontrando e visualizando problemas tópico e os subtópicos. | 19 de abril de 2023 |
| Conteúdo atualizado: Configurando as imagens do Docker do ambiente de computação e tempo de execução para um fluxo de trabalho | Foi adicionado suporte para a arquitetura Arm64 no Amazon Linux 2. | 19 de abril de 2023 |
| Novo conteúdo: Movendo questões dentro de grupos | Foi adicionada documentação para mover questões dentro de grupos no Quadro e nas visualizações de Todas as questões. | 19 de abril de 2023 |

| | | |
|--|---|---------------------|
| Conteúdo atualizado: Cotas para fluxos de trabalho | Atualizou o Cotas para fluxos de trabalho tópico com cotas ausentes e atualizou o tamanho total máximo da cota de variáveis de saída de uma única ação para 120 KB (de 2 KB). | 18 de abril de 2023 |
| Novo conteúdo: Visualizando o código-fonte de uma ação | Adicionou um Visualizando o código-fonte de uma ação tópico. | 18 de abril de 2023 |
| Novo conteúdo: Tentando novamente os casos de teste de um relatório | Adicionou um Tentando novamente os casos de teste de um relatório tópico. | 11 de abril de 2023 |
| Novo conteúdo: Interromper uma execução de fluxo de trabalho | Adicionou um Interromper uma execução de fluxo de trabalho tópico. | 10 de abril de 2023 |
| Novo conteúdo: seções adicionadas para marcar recursos para conexões de contas entre AWS e a Amazon CodeCatalyst | Foram adicionadas informações para marcar recursos de conexão da conta e gerenciar políticas do IAM para recursos de conexão. Consulte Usando tags para controlar o acesso aos recursos de conexão da conta e CodeCatalyst referência de permissões . | 6 de abril de 2023 |
| Novo conteúdo: Tipos de ação | Adicionou um Tipos de ação tópico. | 6 de abril de 2023 |
| Conteúdo atualizado: Definição YAML da ação “Implantar AWS CloudFormation pilha” | Atualizou a descrição da <code>parameter-override</code> propriedade. Agora ele oferece suporte a arquivos JSON. | 5 de abril de 2023 |

| | | |
|---|---|---------------------|
| Novo conteúdo: Criação de um projeto CodeCatalyst com um GitHub repositório vinculado | Foi adicionada uma nova seção Como criar um projeto intitulado Criação de um projeto com um repositório de terceiros vinculado com instruções para criar um projeto vinculado ao seu GitHub repositório. | 5 de abril de 2023 |
| Conteúdo atualizado: trabalhando com notificações | A documentação das notificações foi atualizada para incluir informações sobre a configuração de e-mails sobre eventos do projeto. | 31 de março de 2023 |
| Novo conteúdo | Publicação inicial do guia do Amazon CodeCatalyst Action Development Kit . | 31 de março de 2023 |
| Conteúdo atualizado: reestruturou a seção Spaces na Amazon CodeCatalyst | A seção Espaços foi atualizada removendo páginas de destino e consolidando tópicos. | 29 de março de 2023 |
| Conteúdo atualizado: Tutorial: Implantar um aplicativo no Amazon ECS | Alterado Etapa 1: configurar um AWS usuário e AWS CloudShell para descrever como criar um usuário em AWS IAM Identity Center em vez de AWS Identity and Access Management. A criação de usuários do IAM não é mais recomendada. | 23 de março de 2023 |

| | | |
|---|---|---------------------|
| Conteúdo atualizado: Trabalhando com funções | A documentação das funções de administrador do espaço, administrador do projeto e colaborador foi atualizada para incluir permissões para vincular problemas a pull requests. | 13 de março de 2023 |
| Conteúdo atualizado: trabalhando com pull requests | A documentação de pull requests foi atualizada para incluir informações sobre problemas de vinculação a pull requests. | 13 de março de 2023 |
| Conteúdo atualizado: Trabalhando com problemas | A documentação de problemas foi atualizada para incluir informações sobre como vincular problemas a pull requests. | 13 de março de 2023 |
| Novo conteúdo: Visualizando o status e os detalhes de todas as execuções em seu projeto | Foi adicionada uma seção que descreve a nova página de execução do fluxo de trabalho agregado. | 8 de março de 2023 |
| Novo conteúdo: Como faço para corrigir os erros “A definição do fluxo de trabalho não tem erros”? | Foi adicionada uma seção sobre como solucionar os erros “A definição do fluxo de trabalho tem erros”. | 7 de março de 2023 |
| Conteúdo atualizado: Criação de um fluxo de trabalho | As instruções foram atualizadas para refletir a nova interface do usuário. | 3 de março de 2023 |
| Novo conteúdo: Integração universal-test-runner em uma ação de teste | Adicionou um Integração universal-test-runner em uma ação de teste tópico. | 3 de março de 2023 |

| | | |
|---|--|-------------------------|
| Conteúdo atualizado: Crie, teste e implante com fluxos de trabalho em CodeCatalyst | Várias seções foram atualizadas para refletir os novos filtros de nome de repositório, ramificação e fluxo de trabalho de origem na página de resumo dos fluxos de trabalho. | 2 de março de 2023 |
| Novo conteúdo: Rastreamento do status de implantação por confirmação | Foi adicionada uma seção sobre visualização da qualidade do código e do status de implantação por confirmação. | 27 de fevereiro de 2023 |
| Novo conteúdo: Variáveis produzidas pela fonte ("BranchName" e "CommId") | Foi adicionada uma nova variável BranchName predefinida. | 16 de fevereiro de 2023 |
| Conteúdo atualizado: Gerenciamento de membros espaciais na Amazon CodeCatalyst | 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 |
| Conteúdo atualizado: etapas adicionadas para o gerenciamento de PAT no console da Amazon CodeCatalyst | Foram adicionadas etapas para visualizar, criar e excluir PATs no console. | 15 de fevereiro de 2023 |
| Conteúdo atualizado: Especificando imagens do Docker do ambiente de execução | Foram adicionadas mais ferramentas à tabela de versões da ferramenta de imagem padrão. | 10 de janeiro de 2023 |

| | | |
|---|--|------------------------|
| Conteúdo atualizado: Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos | Corrigido um caminho de artefato. | 3 de janeiro de 2023 |
| Conteúdo atualizado: Definição YAML da GitHub ação "Ações" | Corrigido o trecho de código na Steps seção. | 3 de janeiro de 2023 |
| Conteúdo atualizado: Conectando um fluxo de trabalho a um repositório de origem | Corrigido um caminho de origem. | 3 de janeiro de 2023 |
| Conteúdo atualizado: atualização de uma pull request | A documentação foi atualizada para incluir informações sobre a atualização de revisores obrigatórios ou opcionais para uma pull request. | 23 de dezembro de 2022 |
| Novo conteúdo: Armazenar do arquivos em cache entre execuções de fluxo de trabalho | Foi adicionada uma página para armazenamento em cache de arquivos em um fluxo de trabalho. | 20 de dezembro de 2022 |
| Conteúdo atualizado: trabalhando com pull requests | Atualizou a documentação das pull requests para incluir informações sobre notificações. | 16 de dezembro de 2022 |
| Novo conteúdo: Definição YAML da ação "AWS CDK implantar" | Adicionou uma nova CdkRootPath propriedade. | 16 de dezembro de 2022 |
| Novo conteúdo: Compartilhamento de computação entre ações | Adicionou um Compartilhamento de computação entre ações tópico. | 14 de dezembro de 2022 |

| | | |
|---|--|------------------------|
| Conteúdo atualizado: Compartilhamento de dados entre ações em um fluxo de trabalho usando artefatos | Exemplos fixos mostrando como especificar artefatos de entrada. | 13 de dezembro de 2022 |
| Novo conteúdo: Definição YAML da GitHub ação “Ações” | Foi adicionada uma página de referência dedicada para a ação GitHub Ações. | 13 de dezembro de 2022 |
| Conteúdo atualizado: Cotas para projetos em CodeCatalyst | Atualizou a documentação com um máximo de 100 projetos em um espaço. | 2 de dezembro de 2022 |
| Novo conteúdo | Publicação inicial do Guia CodeCatalyst do usuário da Amazon. | 1º de dezembro de 2022 |

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.

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.