



Guia do usuário

AWS CloudFormation



Versão da API 2010-05-15

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

AWS CloudFormation: Guia do usuário

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

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

Table of Contents

O que é o CloudFormation?	1
Simplificar o gerenciamento de infraestrutura	1
Replicar sua infraestrutura com rapidez	1
Controlar e rastrear alterações em sua infraestrutura com facilidade	2
Conceitos básicos do CloudFormation	2
Informações relacionadas	2
Começar	4
Como o CloudFormation funciona	4
Principais conceitos	4
Como o CloudFormation funciona	8
Maneiras de começar a usar o CloudFormation	12
Cadastrar-se em uma Conta da AWS	12
Inscrever-se para uma Conta da AWS	12
Criar um usuário com acesso administrativo	12
Criar sua primeira pilha	14
Pré-requisitos	15
Criar uma pilha do CloudFormation com o console	15
Monitorar a criação da pilha	19
Testar o servidor Web	20
Solução de problemas	20
Limpeza	22
Próximas etapas	23
Práticas recomendadas	24
Encurtar o ciclo de feedback para melhorar a velocidade de desenvolvimento	25
Organizar pilhas por ciclo de vida e propriedade	26
Use referências entre pilhas para retornar o valor de uma saída exportada por outra pilha	27
Use os StackSets do CloudFormation para implantações em várias contas e em várias regiões	27
Verificar cotas para todos os tipos de recursos	28
Reutilizar modelos para replicar pilhas em vários ambientes	28
Usar módulos para reutilizar configurações de recursos	28
Adote a infraestrutura como práticas de código	29
Não incorporar credenciais em seus modelos	29
Usar tipos de parâmetros específicos da AWS	30

Usar restrições de parâmetros	30
Use pseudoparâmetros para promover a portabilidade	30
Usar <code>AWS::CloudFormation::Init</code> para implantar aplicações de software em instâncias do Amazon EC2	31
Usar os scripts auxiliares mais recentes	32
Validar modelos antes de usá-los	32
Validar modelos para a conformidade com a política da organização	32
Usando YAML ou JSON para criação de modelos	33
Implemente uma estratégia abrangente de marcação	34
Aproveite as macros de modelo para transformações avançadas	34
Gerenciar todos os recursos de pilha via CloudFormation	34
Criar conjuntos de alteração antes de atualizar as pilhas	35
Use as políticas de pilha para proteger os recursos	35
Use o AWS CloudTrail para registrar chamadas do CloudFormation	35
Usar revisões de código e controles de revisão para gerenciar seus modelos	36
Atualizar suas instâncias do Amazon EC2 regularmente	36
Use a detecção de desvios regularmente	36
Configurar gatilhos de reversão para recuperação automática	36
Implemente estratégias eficazes de refatoração de pilhas	37
Use os hooks do CloudFormation para gerenciamento do ciclo de vida	37
Use o Gerador de IaC para criar modelos com recursos existentes	38
Use AWS Infrastructure Composer para design de modelo visual	38
Considere o uso do AWS Cloud Development Kit (AWS CDK) para uma infraestrutura complexa	39
Usar o IAM para controlar o acesso	39
Aplicação do princípio de privilégio mínimo	40
Parâmetros confidenciais seguros	40
Implemente a política como código com o AWS CloudFormation Guard	41
Trabalhar com modelos	42
Onde os modelos são armazenados	43
Validar modelos	43
Conceitos básicos de modelos	2
Planejar o uso da referência do modelo do CloudFormation	44
Exemplos de modelo	45
Formato de modelo	45
Estrutura do modelo	46

Comentários	48
Especificações	49
Saiba mais	49
Usar expressões regulares	49
Seções do modelo	51
Resources	52
Parameters	61
Outputs	74
Mappings	80
Metadata	88
Rules	94
Conditions	114
Transform	126
Versão do formato	127
Description	128
Compositor de Infraestrutura da AWS	129
Por que usar o Infrastructure Composer no modo de console do CloudFormation?	129
Em que esse modo difere do console padrão do Infrastructure Composer?	129
Servidor de linguagens do AWS CloudFormation	130
IDEs integrados ao servidor de linguagens do AWS CloudFormation	130
Introdução	131
Inicialização de um projeto do CloudFormation no IDE	137
Código aberto	138
Precisa de ajuda?	139
Gerador de IaC	139
Considerações	140
Permissões	141
Comandos comumente usados	141
Migrar um modelo para o AWS CDK	142
Iniciar uma verificação de recursos	142
Visualizar resumo da verificação	147
Criar um modelo com base nos recursos verificados	148
Criar uma pilha com base nos recursos verificados	153
Resolver propriedades somente gravação	154
Obter valores armazenados em outros serviços	160
Considerações gerais	160

Obter valor em texto simples do Systems Manager	161
Obter string segura do Systems Manager	165
Obter valor do Secrets Manager	168
Obtenha valores da AWS	173
Sintaxe	173
Pseudoparâmetros disponíveis	174
Exemplos	176
Obter saídas de pilhas	180
Exportar valores de saída da pilha vs usar pilhas aninhadas	181
Considerações	181
Listar valores de saída exportados	182
Listar pilhas que importam um valor de saída exportado	183
Especificar recursos existentes no runtime	184
Visão geral	184
Exemplo	185
Considerações	189
Tipos de parâmetros específicos da AWS compatíveis	189
Tipos de parâmetros compatíveis do Systems Manager	191
Tipos de parâmetros incompatíveis do Systems Manager	192
Instruções	192
Consultar saídas de recurso em outra pilha do CloudFormation	193
Implantar aplicações no Amazon EC2	198
Atualizar um stack	208
Criar uma aplicação dimensionada e com balanceamento de carga	225
Emparelhar com uma VPC em outra conta	239
Executar implantações azul/verde do ECS	249
Trechos de modelo	274
Geral	275
Ajuste de escala automático	285
Console de faturamento da AWS	318
CloudFormation	322
CloudFront	329
CloudWatch	341
CloudWatch Logs	347
DynamoDB	381
Amazon EC2	385

Amazon ECS	423
Amazon EFS	446
Elastic Beanstalk	472
Elastic Load Balancing	476
IAM	481
AWS Lambda	499
Amazon Redshift	506
Amazon RDS	516
Route 53	524
Amazon S3	533
Amazon SNS	542
Amazon SQS	542
Amazon Timestream	543
Pilhas baseadas em Windows	572
Fazer o bootstrap de pilhas do Windows	573
Usar tipos de recursos fornecidos pelo CloudFormation	581
Recursos personalizados	582
Macros de modelo	628
Pilhas aninhadas	646
Condições de espera	653
Criar configurações de recursos reutilizáveis com módulos	659
Considerações sobre o uso de módulos	660
Entender o versionamento de módulos	661
Usar módulos em um modelo	663
Usar parâmetros para especificar valores de módulo	664
Recursos do módulo de referência	667
Criar e gerenciar pilhas	669
Interfaces para gerenciar pilhas	671
Criar uma pilha	672
Criar uma pilha	672
Configurar opções da pilha	675
Visualizar a configuração da sua pilha	678
Visualizar informações da pilha	679
Atualizar seu modelo de pilha	681
Compreender atualização de comportamentos de recursos da pilha	684
Atualizar pilhas usando conjuntos de alterações	685

Criar um conjunto de alterações	687
Visualizar um conjunto de alterações	694
Conjuntos de alterações sensíveis a desvios	702
Executar um conjunto de alterações	709
Excluir um conjunto de alterações	711
Exemplos de conjuntos de alterações	713
Conjuntos de alteração para pilhas aninhadas	725
Validar implantações de pilhas	732
Como funciona a validação pré-implantação	733
Considerações	733
Pré-requisitos	734
Validar a implantação de uma pilha (console)	734
Validar uma implantação de pilha (AWS CLI)	735
Tipos de validação	737
Limitações do recurso	737
Atualizar pilhas diretamente	751
Cancelar uma atualização de pilha	755
Para cancelar uma atualização de pilha (console)	755
Para cancelar uma atualização de pilha (AWS CLI)	756
Excluir uma pilha	756
Recursos relacionados	758
Visualizar pilhas excluídas	758
Monitorar o progresso da pilha	759
Visualizar eventos de pilha	759
Visualizar eventos das pilhas por operação	768
Visualizar um cronograma de implantação da pilha	772
Sobre eventos de criação de pilhas	775
Monitorar atualizações de pilhas	777
Continuar revertendo uma atualização	779
Determinar a causa de uma falha na pilha	783
Stack failure options (Opções de falha da pilha)	785
Reduza sua pilha em caso de violação de alarme	795
Adicionar triggers de reversão durante a criação ou atualização da pilha	796
Adicionar gatilhos de reversão a um conjunto de alterações	797
Visualizar gatilhos de reversão para uma pilha	797
Visualizar gatilhos de reversão de um conjunto de alterações	798

Detectar alterações de configuração não gerenciadas em pilhas e recursos	798
O que é desvio?	799
Códigos de status de detecção de desvio	800
Considerações ao detectar o desvio	802
Detectar desvio em uma pilha inteira do CloudFormation	805
Detectar desvios em recursos de pilha individuais	811
Resolver o desvio com uma operação de importação	815
Importar recursos da AWS	825
Importar recursos da AWS manualmente	826
Importar recursos da AWS automaticamente	865
Reverter uma operação de importação	867
Refatoração de pilhas	872
Como a refatoração de pilhas funciona	873
Considerações sobre refatoração de pilhas	873
Pré-requisitos	874
Refatorar pilhas (console)	874
Refatorar pilhas (AWS CLI)	876
Limitações do recurso	880
Suporte a tipos de recursos	885
Use links de criação rápida para criar pilhas do CloudFormation	1072
Formato do URL	1073
Exemplo	1074
Criando uma pilha usando um link de criação rápida	1075
Exemplos de comandos para o AWS CLI e o PowerShell	1076
CancelUpdateStack	1077
ContinueUpdateRollback	1078
CreateStack	1079
Implantar	1083
DeleteStack	1084
DescribeStackEvents	1085
DescribeStackResource	1087
DescribeStackResources	1088
DescribeStacks	1092
GetTemplate	1095
ListStackResources	1097
ListStacks	1099

UpdateStack	1101
ValidateTemplate	1105
Fazer upload de artefatos locais em um bucket do S3	1107
Gerenciar pilhas com StackSets	1111
Conceitos do StackSets	1112
Contas de administrador e de destino	1113
CloudFormation StackSets	1113
Modelos de permissões para StackSets	1113
Instâncias da pilha	1114
Operações de StackSets	1115
Opções de operação de StackSets	1116
Tags	1118
Códigos de status do StackSets	1119
Códigos de status da instância de pilha	1120
Pré-requisitos	1121
Habilitar Regiões da AWS desabilitadas por padrão	1122
Conceder permissões autogerenciadas	1123
Ativar o acesso confiável	1140
Começar a usar um modelo de exemplo	1145
Pré-requisitos	1146
Criar um StackSet com um modelo de exemplo do console	1146
Monitorar a criação de um StackSet	1149
Visualizar os resultados do StackSet	1150
Atualizar seu StackSet	1150
Adicionar pilhas ao seu StackSet	1151
Limpeza	1151
Próximas etapas	1153
Criar StackSets (permissões autogerenciadas)	1154
Criar um StackSet com permissões autogerenciadas (console)	1154
Criar um StackSet com permissões autogerenciadas (AWS CLI)	1156
Criar um conjunto de pilhas (permissões gerenciadas pelo serviço)	1158
Considerações	1158
Criar um StackSet com permissões gerenciadas pelo serviço (console)	1159
Criar um StackSet com permissões gerenciadas pelo serviço (AWS CLI)	1162
Habilitar-desabilitar implantações automáticas	1166
Como as implantações automáticas funcionam	1166

Considerações	1166
Habilitar ou desabilitar implantações automáticas (console)	1167
Habilitar ou desabilitar implantações automáticas (AWS CLI)	1168
Atualizar StackSets	1168
Atualizar seu StackSet (console)	1169
Atualizar seu StackSet (AWS CLI)	1171
Adicionar pilhas a StackSets	1172
Adicionar pilhas a um StackSet (console)	1173
Adicionar pilhas a um StackSet (AWS CLI)	1175
Substituir parâmetros em pilhas	1176
Substituir parâmetros em pilhas (console)	1176
Substituir parâmetros em pilhas (AWS CLI)	1178
Excluir pilhas de StackSets	1179
Excluir pilhas do seu StackSet (console)	1180
Excluir pilhas do StackSet (AWS CLI)	1182
Excluir StackSets	1183
Excluir um StackSet (console)	1184
Excluir um StackSet (AWS CLI)	1184
Excluir funções de serviço (opcional)	1185
Portões de conta de destino	1186
Requisitos	1186
Modelos do CloudFormation para a criação de funções do Lambda	1187
Escolher do modo de simultaneidade	1187
Como cada Modo de Simultaneidade funciona	1188
Escolha entre Alta tolerância a falhas e Tolerância moderada a falhas com base na velocidade de implantação	1191
Escolher o modo de simultaneidade (console)	1193
Escolher o modo de simultaneidade (AWS CLI)	1193
Detectar desvios em StackSets	1194
Como o CloudFormation executa a detecção de desvios em um StackSet	1195
Detectar desvios em um StackSet (console)	1196
Detectar desvios em um StackSet (AWS CLI)	1197
Como interromper a detecção de desvios em um StackSet	1204
Importar pilhas para o StackSets	1205
Importação de pilhas autogerenciadas	1206
Importação de pilhas gerenciadas por serviço	1209

Reverter importações de pilhas	1215
Práticas recomendadas	1215
Definir o modelo	1215
Como criar ou adicionar pilhas ao StackSet	1216
Como atualizar pilhas em um StackSet	1216
Exemplos de modelo	1217
Solução de problemas	1218
Motivos comuns de falha na operação da pilha	1219
Realizar novamente operações de criação ou atualização de pilha com falha	1219
Falha de exclusão de instância de pilha	1220
Falha na operação de importação de pilhas	1220
Contagem de falhas de instâncias de pilha para operações do StackSets	1221
Sincronizar pilhas com o código-fonte do Git	1228
Como a sincronização do Git funciona	1229
Como a sincronização do Git funciona	1229
Comentários sobre solicitações pull	1230
Arquivo de implantação da pilha	1231
Arquivo de modelo do CloudFormation	1232
Repositório de definição de modelo	1232
Pré-requisitos	1232
Repositório Git	1233
Modelo do CloudFormation	1233
Perfil de serviço de sincronização do Git	1233
Permissões do IAM para usuários do console	1237
Criar uma pilha com base no código-fonte do repositório	1240
Criar uma pilha com base no código-fonte do repositório	1240
Atualize sua pilha a partir do seu repositório Git	1243
Habilitar comentários em solicitações pull	1243
Painel de status	1244
Status do Git Sync	1244
Eventos de sincronização mais recentes	1245
Estados de pilhas com suporte	1245
Gerenciar extensões com o registro do CloudFormation	1250
Documentação relacionada	1250
Conceitos	1251
Tipos de extensão	1251

Extensões públicas	1252
Extensões ativadas	1252
Visualizar as extensões disponíveis e ativadas	1253
Extensões públicas	1254
perfil do IAM	1255
Usar novas versões de extensões automaticamente	1257
Usar aliases para fazer referência a extensões	1258
Comandos da AWS CLI comumente usados para trabalhar com extensões públicas	1259
Ativar uma extensão pública	1259
Atualizar uma extensão pública	1263
Desativar extensões públicas	1264
Extensões privadas	1266
permissões do IAM	1266
Comandos da AWS CLI comumente usados para trabalhar com extensões privadas	1266
Registrar uma extensão privada	1267
Cancelar o registro de extensões privadas	1269
Editar dados de configuração para extensões	1271
Permissões necessárias para usar referências dinâmicas	1272
Editar dados de configuração para uma extensão (console)	1272
Editar dados de configuração para uma extensão (AWS CLI)	1273
Registrar tipos de recursos em AWS Config	1274
Impedir que propriedades confidenciais sejam registradas em um item de configuração ...	1275
Entrega contínua	1276
Demonstração: Criar um pipeline para pilhas de produção e teste	1276
Pré-requisitos	1277
Visão geral da demonstração	1277
Etapa 1: Editar o artefato e fazer upload dele para um bucket do S3	1278
Etapa 2: Criar a pilha do pipeline	1280
Etapa 3: Visualizar a pilha do WordPress	1285
Etapa 4: Limpar os recursos	1286
Consulte também	1287
Referência de propriedades de configuração	1287
Propriedades de configuração (console)	1288
Propriedades de configuração (objeto JSON)	1291
Consulte também	1296
CloudFormationArtefatos do	1296

Arquivo de modelo de pilha	1297
Arquivo de configuração de modelo	1297
Consulte também	1298
Usar funções de substituição de parâmetros com pipelines CodePipeline	1298
Fn::GetArtifactAtt	1299
Fn::GetParam	1300
Consulte também	1303
Guia de referência de modelos	1304
Segurança	1305
Proteger pilhas contra a exclusão	1306
Controlar quem pode alterar a proteção contra encerramento em pilhas	1308
Impedir atualizações nos recursos de pilha	1308
Exemplo de política de pilha	1309
Definir uma política de pilha	1310
Configurar uma política de pilha	1314
Atualizar recursos protegidos	1316
Modificar uma política de pilha	1319
Mais exemplos de políticas de pilha	1320
Proteção de dados	1323
Criptografia em repouso	1324
Criptografia em trânsito	1325
Privacidade do tráfego entre redes	1325
Controlar o acesso com o IAM	1325
Definir políticas baseadas em identidade do IAM para o CloudFormation	1325
Confirmar recursos do IAM em modelos do CloudFormation	1335
Gerenciar credenciais para aplicações executadas em instâncias do Amazon EC2	1336
Conceder acesso temporário (acesso federado)	1337
Exemplos de políticas baseadas em identidade	1338
AWS políticas gerenciadas da	1346
Função de serviço do CloudFormation	1349
Prevenção do problema "confused deputy" entre serviços	1350
Solicitações de FAS e avaliação de permissões	1352
Registrar em log chamadas de API	1355
Informações do CloudFormation no CloudTrail	1355
Noções básicas sobre entradas de arquivos de log do CloudFormation	1356
Segurança da infraestrutura	1360

Resiliência	1361
Validação de conformidade	1361
Análise de configuração e vulnerabilidade	1362
Práticas recomendadas de segurança	1362
Usar o IAM para controlar o acesso	1362
Não incorporar credenciais em seus modelos	1363
Use o AWS CloudTrail para registrar chamadas do CloudFormation	1363
AWS PrivateLink	1363
Considerações sobre endpoints da VPC do CloudFormation	1364
Criar um endpoint da VPC de interface para o CloudFormation	1365
Criando uma política de endpoint da VPC para o CloudFormation	1365
Monitoramento com o EventBridge	1367
Visão geral dos eventos de sincronização do CloudFormation e do Git	1368
Permissões	1370
Criar um padrão de eventos personalizado	1370
Referência detalhada de eventos	1371
Alteração de status de recurso	1373
Alteração de status de desvios	1375
Alteração de status de detecção de desvios	1377
Alteração de status do StackSet	1380
Alteração de status de instância de pilha do StackSet	1381
Alteração de status de operação do StackSet	1384
Alteração do status de sincronização do repositório	1386
Alteração do status de sincronização de recursos	1389
Cotas	1392
Disponibilidade de recursos	1400
StackSets e macros	1400
Solução de problemas	1401
Guia de solução de problemas	1401
Solucionar de problemas de erros	1402
Falha na exclusão da pilha	1403
Erro de dependência	1404
Conflitos do AWS Config e do AWS Systems Manager	1404
Erro ao analisar parâmetros ao passar uma lista	1405
Permissões insuficientes do IAM	1405
Valor inválido ou propriedade de recurso sem suporte	1405

Cota excedida	1406
Falha na reversão de pilhas aninhadas	1406
Nenhuma atualização a ser executada	1407
Falha ao estabilizar recurso durante uma operação de criação, atualização ou exclusão de pilha	1407
O Grupo de segurança não existe na VPC	1408
Falha na reversão da atualização	1408
A condição de espera não recebeu o número necessário de sinais de uma instância do Amazon EC2	1410
Recurso removido da pilha, mas não excluído	1410
Entrar em contato com o suporte	1411
Solução de problemas com o Amazon Q	1413
Recursos	1413
Como funciona	1414
Preços	1414
Disponibilidade regional	1414
Limitações	1414
Histórico do documento	1415
Atualizações arquivadas	1444

O que é o CloudFormation?

O CloudFormation é um serviço que ajuda você a modelar e configurar seus recursos da AWS para despendar menos tempo gerenciando esses recursos e mais tempo se concentrando em seus aplicativos executados AWS. Você cria um modelo que descreve todos os recursos da AWS desejados (como funções do Amazon EC2 e tabelas do Amazon RDS), e o CloudFormation cuida do provisionamento e da configuração desses recursos para você. Não é necessário criar e configurar individualmente os recursos da AWS e descobrir o que depende do que: o CloudFormation lida com tudo isso. Os cenários a seguir demonstram como o CloudFormation pode ajudar.

Simplificar o gerenciamento de infraestrutura

Para uma aplicação Web escalável que também inclui um banco de dados de backend, é possível usar um grupo do Auto Scaling, um balanceador de carga do Elastic Load Balancing e uma instância de banco de dados do Amazon Relational Database Service. É possível utilizar cada serviço individual para provisionar esses recursos e, depois de criar os recursos, você precisa configurá-los para trabalharem em conjunto. Todas essas tarefas podem adicionar complexidade e tempo antes que você consiga configurar e executar seu aplicativo.

Em vez disso, é possível criar ou modificar um modelo de CloudFormation existente. Um modelo descreve todos os seus recursos e suas propriedades. Ao usar esse modelo para criar uma pilha do CloudFormation, o CloudFormation provisionará o grupo do Auto Scaling, o balanceador de carga e o banco de dados para você. Depois que a pilha tiver sido criada com êxito, seus recursos da AWS estão em funcionamento. É possível excluir a pilha com facilidade, o que exclui todos os recursos na pilha. Ao usar o CloudFormation, você gerencia facilmente um conjunto de recursos como uma unidade.

Replicar sua infraestrutura com rapidez

Se o seu aplicativo requer mais disponibilidade, é possível replicá-lo em várias regiões, de forma que se uma região se tornar indisponível, os usuários ainda podem usar o aplicativo em outras regiões. O desafio da replicação de seu aplicativo é que ele também exige que você replique seus recursos. Além da necessidade de registrar todos os recursos que o seu aplicativo exige, também é necessário provisionar e configurar esses recursos em cada região.

Reutilize seu modelo do CloudFormation para criar seus recursos de modo consistente e repetível. Para reutilizar seu modelo, descreva seus recursos uma vez e, em seguida, provisione os mesmos recursos cada vez mais em diversas regiões.

Controlar e rastrear alterações em sua infraestrutura com facilidade

Em alguns casos, você pode ter recursos subjacentes que deseja atualizar de forma incremental. Por exemplo, é possível mudar para um tipo de instância com maior desempenho em sua configuração de execução do Auto Scaling, de modo que seja possível reduzir o número máximo de instâncias no seu grupo do Auto Scaling. Se ocorrerem problemas após a conclusão da atualização, pode ser necessário reverter sua infraestrutura para as configurações originais. Para fazer isso manualmente, além da necessidade de lembrar quais recursos foram alterados, também é necessário saber quais eram as configurações originais.

Quando você provisiona sua infraestrutura com o CloudFormation, o modelo do CloudFormation descreve exatamente quais recursos são provisionados e as configurações deles. Como esses modelos são arquivos de texto, basta rastrear as diferenças em seus modelos para rastrear alterações em sua infraestrutura, semelhante à forma como os desenvolvedores controlam revisões de código-fonte. Por exemplo, é possível usar um sistema de controle de versão com seus modelos para que você saiba exatamente quais alterações foram feitas, quem as fez e quando elas foram feitas. Se, a qualquer momento, for necessário reverter alterações em sua infraestrutura, use uma versão anterior do seu modelo.

Conceitos básicos do CloudFormation

O CloudFormation está disponível por meio do [console](#) do CloudFormation, da [API](#), da [AWS CLI](#), dos [AWS SDKs](#) e por meio de várias integrações.

Consulte uma introdução ao CloudFormation em [Como o CloudFormation funciona](#).

Saiba como começar a usar o CloudFormation em [Criar sua primeira pilha](#).

Informações relacionadas

É possível saber mais sobre o CloudFormation neste guia do usuário, bem como nos seguintes recursos:

- Consulte detalhes do produto e perguntas frequentes na [página do produto do AWS CloudFormation](#).

- Para obter informações sobre preços, consulte [AWS CloudFormation preços](#).

Conceitos básicos do CloudFormation

Você pode começar a usar o CloudFormation por meio do Console de gerenciamento da AWS criando uma pilha a partir de um modelo de exemplo, o que ajudará você a aprender os fundamentos da criação de pilhas. Um modelo é um arquivo de texto que define todos os recursos em uma pilha. Uma pilha é a implantação de um modelo do CloudFormation. Com um único modelo, é possível criar várias pilhas. Cada pilha contém uma coleção de recursos da AWS que podem ser gerenciados como uma unidade.

O CloudFormation é um serviço gratuito, porém os recursos da AWS que você incluir nas pilhas serão cobrados segundo as tarifas correntes de cada um deles. Para obter mais informações sobre o preço da AWS, acesse a página de detalhes de cada produto em <http://aws.amazon.com>.

Tópicos

- [Como o CloudFormation funciona](#)
- [Cadastrar-se em uma Conta da AWS](#)
- [Criar sua primeira pilha](#)

Como o CloudFormation funciona

Este tópico descreve como o CloudFormation funciona e apresenta os principais conceitos que você precisa conhecer ao usá-lo.

Tópicos

- [Principais conceitos](#)
- [Como o CloudFormation funciona](#)
- [Maneiras de começar a usar o CloudFormation](#)

Principais conceitos

Ao usar o CloudFormation, você trabalha com modelos e pilhas. Você cria modelos para descrever os recursos da AWS e suas respectivas propriedades. Sempre que você cria uma pilha, o CloudFormation provisiona os recursos descritos no modelo.

Tópicos

- [Modelos](#)
- [Pilhas](#)
- [Conjuntos de alterações](#)

Modelos

Um modelo do CloudFormation é um arquivo de texto formatado em YAML ou JSON. Você pode salvar esses arquivos com qualquer extensão, por exemplo, `.yaml`, `.json`, `.template` ou `.txt`. O CloudFormation usa esses modelos como esquemas para a criação dos recursos da AWS. Por exemplo, em um modelo, é possível descrever uma instância do Amazon EC2, como o tipo de instância, o ID da AMI, os mapeamentos de dispositivos de blocos e o nome do par de chaves do Amazon EC2. Sempre que cria uma pilha, você também deve especificar um modelo usado pelo CloudFormation para criar o que você descreveu no modelo.

Por exemplo, caso você tenha criado uma pilha usando o modelo a seguir, o CloudFormation provisiona uma instância com uma ID da AMI `ami-0ff8a91507f77f867`, um tipo de instância `t2.micro`, um nome do par de chaves `testkey` e um volume do Amazon EBS.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

JSON

```
{
```

```

"AWSTemplateFormatVersion": "2010-09-09",
"Description": "A sample template",
"Resources": {
  "MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "ami-0ff8a91507f77f867",
      "InstanceType": "t2.micro",
      "KeyName": "testkey",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sdm",
          "Ebs": {
            "VolumeType": "io1",
            "Iops": 200,
            "DeleteOnTermination": false,
            "VolumeSize": 20
          }
        }
      ]
    }
  }
}

```

Também é possível especificar vários recursos em um único modelo e configurar esses recursos para que funcionem juntos. Por exemplo, é possível modificar o modelo anterior para incluir um endereço IP elástico (EIP) e associá-lo à instância Amazon EC2, conforme mostrado no seguinte exemplo:

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm

```

```
Ebs:
  VolumeType: io1
  Iops: 200
  DeleteOnTermination: false
  VolumeSize: 20
MyEIP:
  Type: 'AWS::EC2::EIP'
  Properties:
    InstanceId: !Ref MyEC2Instance
```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A sample template",
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "t2.micro",
        "KeyName": "testkey",
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/sdm",
            "Ebs": {
              "VolumeType": "io1",
              "Iops": 200,
              "DeleteOnTermination": false,
              "VolumeSize": 20
            }
          }
        ]
      }
    },
    "MyEIP": {
      "Type": "AWS::EC2::EIP",
      "Properties": {
        "InstanceId": {
          "Ref": "MyEC2Instance"
        }
      }
    }
  }
}
```

```
}  
}
```

Os modelos anteriores se baseiam em uma instância do Amazon EC2 única. No entanto, os modelos do CloudFormation têm recursos adicionais que é possível usar para criar conjuntos complexos de recursos e reutilizar esses modelos em vários contextos. Por exemplo, é possível adicionar parâmetros de entrada cujos valores são especificados quando você cria uma pilha do CloudFormation. Em outras palavras, você pode especificar um valor como o tipo de instância ao criar uma pilha, em vez de quando cria o modelo, o que facilita a reutilização do modelo em situações diferentes.

Pilhas

Ao usar o CloudFormation, você gerencia recursos relacionados como uma única unidade, chamada pilha. Você pode criar, atualizar e excluir um conjunto de recursos criando, atualizando e excluindo pilhas. Todos os recursos em uma pilha são definidos pelo modelo CloudFormation dela. Suponhamos que você tenha criado um modelo que inclui um grupo do Auto Scaling, um load balancer do Elastic Load Balancing e uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS). Para criar esses recursos, você cria uma pilha enviando o modelo que criou, e o CloudFormation provisiona todos esses recursos para você.

Conjuntos de alterações

Caso precise fazer alterações nos recursos em execução em uma pilha, você atualiza a pilha. Antes de fazer alterações nos recursos, você pode gerar um conjunto de alterações, que é o resumo das alterações propostas. Os conjuntos de alterações permitem ver como as alterações podem afetar os recursos em execução, especialmente para recursos críticos, antes de implementá-las.

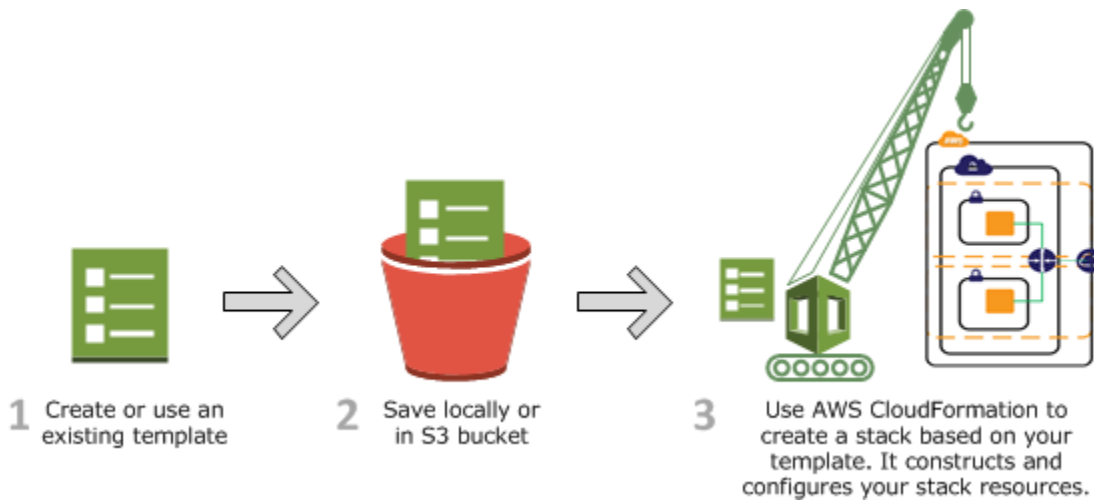
Por exemplo, se você alterar o nome de uma instância de banco de dados do Amazon RDS, o CloudFormation criará um novo banco de dados e excluirá o anterior. Você perderá os dados no banco de dados anterior, a menos que já tenha feito backup deles. Se gerar um conjunto de alterações, você verá que a alteração fará o banco de dados ser substituído, e poderá planejar de acordo antes de atualizar a pilha.

Como o CloudFormation funciona

Quando você usa o CloudFormation para criar sua pilha, o CloudFormation faz chamadas de serviço subjacentes à AWS para provisionar e configurar os recursos descritos no seu modelo. Você precisa de permissão para criar esses recursos. Por exemplo, para criar instâncias do EC2 usando o

CloudFormation, você precisa ter permissões para criar instâncias. Você gerencia essas permissões com [AWS Identity and Access Management](#) (IAM).

As chamadas que o CloudFormation faz são todas declaradas por seu modelo. Por exemplo, suponha que você tem um modelo que descreve uma instância EC2 com um tipo de instância `t2.micro`. Quando você usa esse modelo para criar uma pilha, o CloudFormation chama a API de criação de instância do Amazon EC2 e especifica o tipo de instância como `t2.micro`. O diagrama a seguir resume o fluxo de trabalho do CloudFormation para a criação de pilhas.



Para criar uma pilha

1. Use um editor de texto para criar um modelo do CloudFormation no formato YAML ou JSON. O modelo de CloudFormation descreve os recursos que você deseja e suas configurações. Use [Compositor de Infraestrutura da AWS](#) para visualizar e validar seu modelo. Isso ajuda você a garantir que seu modelo esteja adequadamente estruturado e livre de erros de sintaxe. Para obter mais informações, consulte [Trabalhar com modelos do CloudFormation](#).
2. Salve o modelo do localmente ou em um bucket do Amazon S3.
3. Crie uma pilha do CloudFormation especificando o local de seu arquivo de modelo, como um caminho em seu computador local ou um URL do Amazon S3. Se o modelo contiver parâmetros, você poderá especificar valores de entrada ao criar a pilha. Os parâmetros permitem que você passe valores para seu modelo, para que possa personalizar seus recursos toda vez que você criar uma pilha.

Note

Se você especificar um arquivo de modelo armazenado localmente, o CloudFormation o carregará em um bucket do S3 na sua conta da AWS. O CloudFormation cria um

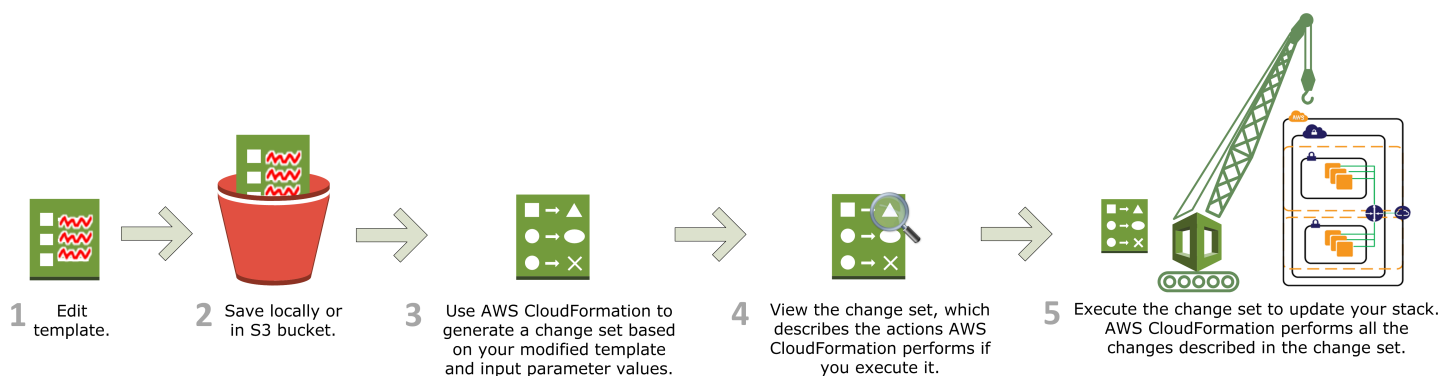
bucket para cada região na qual você carrega um arquivo de modelo. Os buckets são acessíveis a qualquer pessoa com permissões do Amazon Simple Storage Service (Amazon S3) na sua conta da AWS. Se um bucket criado pelo CloudFormation já estiver presente, o modelo será adicionado a esse bucket.

É possível utilizar seu próprio bucket e gerenciar suas permissões carregando modelos manualmente no Amazon S3. Em seguida, sempre que você criar ou atualizar uma pilha, especifique o URL do Amazon S3 de um arquivo de modelo.

Depois que todos os recursos foram criados, o CloudFormation informa que sua pilha foi criada. Em seguida, você pode começar a usar os recursos de sua pilha. Se houver falha na criação da pilha, o CloudFormation reverterá as alterações excluindo os recursos que ele criou.

Atualizar uma pilha com um conjunto de alterações

Quando precisar atualizar os recursos da pilha, você poderá modificar o modelo da pilha. Você não precisa criar uma nova pilha e excluir a antiga. Para atualizar uma pilha, crie um conjunto de alterações enviando uma versão modificada do modelo da pilha original, diferentes valores de parâmetros de entrada ou ambos. O CloudFormation compara o modelo modificado com o modelo original e gera um conjunto de alterações. O conjunto de alterações lista as alterações propostas. Depois de revisar as alterações, você pode iniciar o conjunto de alterações para atualizar sua pilha ou pode criar um novo conjunto de alterações. O diagrama a seguir resume o fluxo de trabalho da atualização de uma pilha.



Para atualizar uma pilha com um conjunto de alterações

1. Você pode modificar um modelo de pilha do CloudFormation usando o [Compositor de Infraestrutura da AWS](#) ou um editor de texto. Para obter mais informações, consulte [Atualizar seu modelo de pilha](#).

Ao atualizar seu modelo, lembre-se de que as atualizações podem causar interrupções. Dependendo do recurso e das propriedades que você está atualizando, uma atualização pode interromper ou até substituir um recurso existente. Para obter mais informações, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

2. Salve o modelo do CloudFormation localmente ou em um bucket do S3.
3. Crie um conjunto de alterações especificando a pilha que você deseja atualizar e o local do modelo modificado, como um caminho no seu computador local ou um URL do Amazon S3. Para obter mais informações sobre como criar conjuntos de alterações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

 Note

Se você especificar um modelo armazenado em seu computador local, o CloudFormation carregará esse modelo automaticamente em um bucket do S3 na sua Conta da AWS.

4. Visualize o conjunto de alterações para verificar se o CloudFormation executará as alterações que você espera. Por exemplo, verifique se o CloudFormation substituirá todos os recursos cruciais da pilha. Você pode criar quantos conjuntos de alterações precisar até ter incluído as alterações desejadas.

 Important

Os conjuntos de alterações não indicam se a atualização da pilha será bem-sucedida. Por exemplo, um conjunto de alterações não verifica se você vai ultrapassar a [cota](#) de uma conta, se estiver atualizando um recurso que não permita alterações nem se tem [permissões](#) suficientes para modificar um recurso, o que pode provocar falha em uma atualização de pilha.

5. Inicie o conjunto de alterações que você deseja aplicar à sua pilha. O CloudFormation atualiza a pilha atualizando apenas os recursos que você modificou e sinaliza que sua pilha foi atualizada com êxito. Se houver falha na atualização da pilha, o CloudFormation reverterá as alterações para restaurar a pilha para o estado de funcionamento conhecido mais recente.

Maneiras de começar a usar o CloudFormation

Para criar uma pilha Hello world do CloudFormation com o console, consulte [Criar sua primeira pilha](#).

Para um aprendizado guiado, experimente o workshop [Introdução ao CloudFormation](#), que oferece experiência prática com desenvolvimento de modelos.

Cadastrar-se em uma Conta da AWS

Ao cadastrar-se na AWS, sua Conta da AWS é automaticamente cadastrada em todos os serviços da AWS, incluindo o CloudFormation. Se você já tiver uma Conta da AWS, pule para o próximo tópico. Se você não tem uma Conta da AWS, siga o procedimento abaixo para criar uma.

Inscrever-se para uma Conta da AWS

Se você ainda não tem uma Conta da AWS, siga as etapas abaixo para criar uma.

Como cadastrar uma Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve para uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

A AWS envia um e-mail de confirmação depois que o processo de inscrição é concluído. A qualquer momento, é possível exibir as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se cadastrar em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS, habilite o AWS IAM Identity Center e crie um usuário administrativo para não usar o usuário-raiz em tarefas cotidianas.

Proteger o Usuário raiz da conta da AWS

1. Faça login no [Console de gerenciamento da AWS](#) como o proprietário da conta ao escolher a opção Usuário-raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS.

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo MFA virtual para sua Conta da AWS de usuário-raiz \(console\)](#) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center.

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para obter um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como a fonte de identidade, consulte [Configurar o acesso dos usuários com o Diretório do Centro de Identidade do IAM padrão](#) no Guia do usuário do AWS IAM Identity Center.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do Centro de Identidade do IAM, consulte [Fazer login no portal de acesso da AWS](#), no Guia do usuário do Início de Sessão da AWS.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center.

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do AWS IAM Identity Center.

Note

Para obter mais informações sobre como gerenciar quem acessa o quê, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Criar sua primeira pilha

Este tópico mostra a você o passo a passo da criação da sua primeira pilha do CloudFormation usando o Console de gerenciamento da AWS. Seguindo este tutorial, você aprenderá a provisionar recursos básicos da AWS, monitorar eventos de pilha e gerar saídas.

Neste exemplo, o modelo do CloudFormation é escrito em YAML. O YAML é um formato legível por humanos que é amplamente usado para definir infraestrutura como código. À medida que aprender mais sobre o CloudFormation, também poderá encontrar outros modelos no formato JSON, mas para este tutorial, o YAML é escolhido por sua legibilidade.

Note

O CloudFormation é gratuito, mas os recursos do Amazon EC2 e do Amazon S3 que você criar serão cobrados. Porém, se você for novo na AWS, poderá aproveitar o [nível gratuito](#) para minimizar ou eliminar os custos durante este processo de aprendizagem.

Tópicos

- [Pré-requisitos](#)
- [Criar uma pilha do CloudFormation com o console](#)
- [Monitorar a criação da pilha](#)
- [Testar o servidor Web](#)

- [Solução de problemas](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

- Você deve ter acesso a uma Conta da AWS com um usuário ou perfil do IAM que tenha permissões para usar o Amazon EC2, o Amazon S3 e o CloudFormation ou ter acesso de usuário administrativo.
- Você deve ter uma nuvem privada virtual (VPC) que tenha acesso à internet. Este modelo passo a passo requer uma VPC padrão, que vem automaticamente com as Contas da AWS mais recentes. Se você não tiver uma VPC padrão, ou se ela tiver sido excluída, consulte a seção de solução de problemas neste tópico para ver as soluções alternativas.

Criar uma pilha do CloudFormation com o console

Para criar uma pilha Hello world do CloudFormation com o console

1. Abra o [console do CloudFormation](#).
2. Selecione Create Stack (Criar pilha).
3. Na página Criar pilha, escolha Compilar do Infrastructure Composer e depois Criar no Infrastructure Composer. Isso leva você ao Infrastructure Composer no modo de console do CloudFormation, onde você pode carregar e validar o exemplo de modelo.
4. Para carregar e validar o modelo de exemplo, faça o seguinte:
 - a. Escolha Modelos. Depois, copie e cole o seguinte modelo do CloudFormation no editor de modelos:

```
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for WebServer with Security Group and EC2
  Instance

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

```
InstanceType:
  Description: WebServer EC2 instance type
  Type: String
  Default: t2.micro
  AllowedValues:
    - t3.micro
    - t2.micro
  ConstraintDescription: must be a valid EC2 instance type.

MyIP:
  Description: Your IP address in CIDR format (e.g. 203.0.113.1/32).
  Type: String
  MinLength: '9'
  MaxLength: '18'
  Default: 0.0.0.0/0
  AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\/\d{1,2}$'
  ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP access via my IP address
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: !Ref MyIP

  WebServer:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      SecurityGroupIds:
        - !Ref WebServerSecurityGroup
      UserData: !Base64 |
        #!/bin/bash
        yum update -y
        yum install -y httpd
        systemctl start httpd
        systemctl enable httpd
```

```
echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/index.html
```

Outputs:

WebsiteURL:

Value: !Join

- ''

- - http://

- !GetAtt WebServer.PublicDnsName

Description: Website URL

Antes de passar para a próxima etapa, vamos examinar o modelo e entender alguns conceitos-chave do Lambda.

- A seção **Parameters** declara os valores que podem ser passados para o modelo quando você cria a pilha. Os recursos especificados posteriormente nesse modelo fazem referência a esses valores e usam os dados. Parâmetros são uma maneira eficaz de especificar informações que você não deseja armazenar no modelo em si. Também são uma maneira de especificar informações que sejam exclusivas da aplicação ou da configuração que você estiver implantando.
- O modelo define os seguintes parâmetros:
 - **LatestAmiId**: recupera o ID do Amazon Linux 2 AMI mais recente da AWS Systems Manager Parameter Store.
 - **InstanceType**: permite a seleção do tipo de instância do EC2 (padrão: t2.micro, permitido: t3.micro, t2.micro).
 - **MyIP**: especifica o intervalo de endereços IP para acesso por HTTP (padrão: 0.0.0.0/0, permitindo acesso de qualquer IP).
- A seção **Resources** contém as definições dos recursos da AWS que você deseja criar com o modelo. As declarações de recursos são uma maneira eficiente de especificar todas essas definições da configuração de uma só vez. Quando você inclui declarações de recursos em um modelo, pode criar e configurar todos os recursos declarados usando esse modelo para criar uma pilha. Você também pode criar novas pilhas a partir do mesmo modelo para iniciar configurações de recursos idênticas.
- Este modelo cria os seguintes recursos:
 - **WebServerSecurityGroup**: um grupo de segurança do EC2 que permite tráfego HTTP de entrada na porta 80 originado no intervalo de IP especificado.
 - **WebServer**: uma instância do EC2 com a seguinte configuração:

- Usa o Amazon Linux 2 AMI mais recente
 - Aplica o tipo de instância selecionado
 - Adiciona o `WebServerSecurityGroup` à propriedade `SecurityGroupIds`
 - Inclui um script de dados de usuário para instalar o Apache HTTP Server
 - Um nome lógico é especificado no início de cada declaração de recurso e parâmetro. Por exemplo, `WebServerSecurityGroup` é o nome lógico atribuído ao recurso do grupo de segurança do EC2. A função `Ref` é usada então para referenciar recursos e parâmetros por seus nomes lógicos em outras partes do modelo. Quando um recurso referencia outro, isso cria uma dependência entre eles.
 - A seção **Outputs** define valores personalizados que são retornados após a criação da pilha. Você pode usar os valores de saída para retornar informações dos recursos na pilha, como os identificadores ou URLs dos recursos.
 - O modelo define uma única saída:
 - **WebsiteURL**: a URL do servidor Web implantado, construída usando o nome de DNS público da instância EC2. A função `Join` ajuda a combinar o `http://` fixo com o `PublicDnsName` variável em uma única string, facilitando a saída da URL completa do servidor Web.
- b. Escolha Validar para se certificar de que o código YAML seja válido antes de carregar o modelo.
 - c. Depois, escolha Criar modelo para criar o modelo e adicioná-lo a um bucket do S3.
 - d. Na caixa de diálogo que se abre, anote o nome do bucket do S3 para poder excluí-lo posteriormente. Depois, escolha Confirmar e continuar no CloudFormation. Isso leva você para o console do CloudFormation, onde o caminho do S3 para o modelo agora está especificado.
5. Na página Criar pilha, selecione Avançar.
 6. Na página Especificar detalhes da tarefa, digite um nome no campo Nome da pilha. O nome da pilha não pode conter espaços. Neste exemplo, use **MyTestStack**.
 7. Em Parâmetros, especifique os valores dos parâmetros da seguinte maneira:
 - `LatestAmild`: isso é definido por padrão como o Amazon Linux 2 AMI mais recente.
 - `InstanceType`: escolha `t2.micro` ou `t3.micro` como o tipo de instância do EC2.

 Note

Se você for novo na AWS, poderá usar o nível gratuito para iniciar e usar uma instância `t2.micro` gratuitamente por 12 meses (em regiões onde `t2.micro` não estiver disponível, será possível usar uma instância do `t3.micro` no nível gratuito).

- MyIP: especifique seu endereço IP público real com um sufixo `/32`. O sufixo `/32` é usado na notação CIDR para especificar que apenas um endereço IP é permitido. Isso significa, em essência, permitir tráfego de e para esse endereço IP específico, e nenhum outro.
8. Escolha Avançar duas vezes para ir até a página Revisar e criar. Para este tutorial, você pode deixar os padrões na página Configurar opções da pilha como estão.
 9. Revise as informações da pilha. Quando estiver satisfeito com as configurações, clique em Submit (Enviar).

Monitorar a criação da pilha

Depois que você escolhe Enviar, o CloudFormation começa a criar os recursos especificados no modelo. Sua nova pilha, **MyTestStack**, aparecerá na lista na parte superior do console do CloudFormation. Seu status deveria ser `CREATE_IN_PROGRESS`. Você pode ver o status detalhado de uma pilha visualizando seus eventos.

Para visualizar os eventos da pilha

1. No console do CloudFormation, escolha a pilha **MyTestStack** na lista.
2. No painel de detalhes da pilha, clique na guia Eventos.

O console atualiza automaticamente a lista de eventos com os eventos mais recentes a cada 60 segundos.

A guia Events (Eventos) exibe cada etapa principal da criação da pilha classificada pelo tempo de cada evento, com os eventos mais recentes na parte superior.

O primeiro evento (na parte inferior da lista de eventos) é o início do processo de criação da pilha:

```
2024-12-23 18:54 UTC-7 MyTestStack CREATE_IN_PROGRESS User initiated
```

Em seguida, estão os eventos que marcam o início e a conclusão da criação de cada recurso. Por exemplo, a criação da instância EC2 resulta nas seguintes entradas:

```
2024-12-23 18:59 UTC-7 WebServer CREATE_COMPLETE
```

```
2024-12-23 18:54 UTC-7 WebServer CREATE_IN_PROGRESS Resource creation
initiated
```

O evento `CREATE_IN_PROGRESS` é registrado quando o CloudFormation relata que começou a criar o recurso. O evento `CREATE_COMPLETE` é registrado em log quando o recurso é criado com êxito.

Quando o CloudFormation tiver criado a pilha com êxito, você verá o seguinte evento na parte superior da guia Eventos:

```
2024-12-23 19:17 UTC-7 MyTestStack CREATE_COMPLETE
```

Se não puder criar um recurso, o CloudFormation relatará um evento `CREATE_FAILED` e, por padrão, reverterá a pilha e excluirá todos os recursos que foram criados. A coluna `Status Reason` (Motivo do status) exibe o problema que provocou a falha.

Depois que a pilha é criada, você pode acessar a guia `Recursos` para visualizar a instância e o grupo de segurança do EC2 criados.

Testar o servidor Web

Depois que a pilha for criada com sucesso, navegue até a guia `Saídas` no console do CloudFormation. Procure o campo `WebsiteURL`. Isso conterá a URL pública da instância do EC2.

Abra um navegador e acesse a URL listada em `WebsiteURL`. Você deve ver uma mensagem "Hello World!" simples exibida no navegador.

Isso confirma que a instância do EC2 está executando o Apache HTTP Server e servindo uma página da Web básica.

Solução de problemas

Se você experimentar uma reversão durante a criação de uma pilha, pode ser devido à falta de uma VPC. Confira como resolver esse problema.

Nenhuma VPC padrão disponível

O modelo neste passo a passo requer uma VPC padrão. Se a criação da pilha falhar devido a erros de disponibilidade de uma VPC ou sub-rede, você talvez não tenha uma VPC padrão em sua conta. Você tem as seguintes opções:

- Criar uma nova VPC padrão: você pode criar uma nova VPC padrão no console da Amazon VPC. Para obter instruções, consulte [Criar uma VPC padrão](#) no Guia do usuário da Amazon VPC.
- Modificar o modelo para especificar uma sub-rede: caso tenha uma VPC não padrão, você poderá modificar o modelo para especificar explicitamente os IDs da VPC e da sub-rede. Adicione o seguinte parâmetro ao modelo:

```
SubnetId:
  Description: The subnet ID to launch the instance into
  Type: AWS::EC2::Subnet::Id
```

Em seguida, atualize o recurso WebServer para incluir o ID da sub-rede:

```
WebServer:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    SubnetId: !Ref SubnetId
    UserData: !Base64 |
      #!/bin/bash
      yum update -y
      yum install -y httpd
      systemctl start httpd
      systemctl enable httpd
      echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html
```

Ao criar a pilha, você precisará especificar uma sub-rede que tenha acesso à internet para que o servidor Web possa ser acessado.

Limpeza

Para ter certeza de que você não será cobrado por qualquer serviço não desejado, poderá fazer uma limpeza excluindo a pilha e os recursos dela. Você também pode excluir o bucket do S3 que armazena o modelo da pilha.

Para excluir a pilha e seus recursos

1. Abra o [console do CloudFormation](#).
2. Na página Pilhas, selecione a opção ao lado do nome da pilha que você criou (**MyTestStack**) e escolha Excluir.
3. Quando a confirmação for solicitada, escolha Excluir.
4. Monitore o andamento do processo de exclusão da pilha na guia Evento. O status de **MyTestStack** é alterado para DELETE_IN_PROGRESS. Quando o CloudFormation conclui a exclusão da pilha, ela é removida da lista.

Se tiver terminado de trabalhar com o exemplo de modelo e não precisar mais do bucket do Amazon S3, exclua-o. Para excluir um bucket, primeiro exclua seu conteúdo. Esvaziar um bucket exclui todos os objetos nele contidos.

Para esvaziar e excluir o bucket do Amazon S3

1. Abra o [console Amazon S3](#).
2. No painel de navegação, no lado esquerdo do console, selecione Buckets.
3. Na lista Buckets, selecione a opção ao lado do nome do bucket que você criou para este tutorial e depois escolha Esvaziar.
4. Na página Esvaziar bucket, confirme se deseja esvaziar o bucket digitando **permanently delete** no campo de texto, depois escolha Esvaziar.
5. Monitore o andamento do processo de esvaziamento do bucket na página Esvaziar bucket: status.
6. Para retornar à sua lista de buckets, escolha Exit (Sair).
7. Selecione a opção ao lado do nome do bucket e escolha Excluir.
8. Quando for solicitada a confirmação, digite o nome do bucket e escolha Excluir bucket.
9. Monitore o andamento do processo de exclusão do bucket na lista Buckets. Quando o Amazon S3 conclui a exclusão do bucket, remove-o da lista.

Próximas etapas

Parabéns! Você criou uma pilha, monitorou sua criação e usou sua saída com sucesso.

Para continuar aprendendo:

- Saiba mais sobre modelos para que possa criar os seus próprios. Para obter mais informações, consulte [Trabalhar com modelos do CloudFormation](#).
- Confira o workshop [Getting Started with CloudFormation](#) para praticar ainda mais a criação de modelos.
- Para obter uma versão resumida de [Getting Started with CloudFormation](#), consulte [Implantar aplicações no Amazon EC2](#). Este tópico descreve o mesmo cenário de uso de um script auxiliar do CloudFormation, `cfn-init`, para fazer o bootstrap de uma instância do Amazon EC2.

Práticas recomendadas do CloudFormation

As práticas recomendadas são sugestões que podem ajudar você a usar o CloudFormation de forma mais eficaz e adotar práticas seguras em todo o seu fluxo de trabalho. Saiba como planejar e organizar suas pilhas, criar modelos que descrevem os recursos e os aplicativos de software que são executados neles, e gerenciar as pilhas e seus respectivos recursos. As práticas recomendadas a seguir são baseadas em experiência real de clientes atuais do CloudFormation.

Planejamento e organização

- [Encurtar o ciclo de feedback para melhorar a velocidade de desenvolvimento](#)
- [Organizar pilhas por ciclo de vida e propriedade](#)
- [Use referências entre pilhas para retornar o valor de uma saída exportada por outra pilha](#)
- [Use os StackSets do CloudFormation para implantações em várias contas e em várias regiões](#)
- [Reutilizar modelos para replicar pilhas em vários ambientes](#)
- [Verificar cotas para todos os tipos de recursos](#)
- [Usar módulos para reutilizar configurações de recursos](#)
- [Adote a infraestrutura como práticas de código](#)

Criação de modelos

- [Não incorporar credenciais em seus modelos](#)
- [Usar tipos de parâmetros específicos da AWS](#)
- [Usar restrições de parâmetros](#)
- [Use pseudoparâmetros para promover a portabilidade](#)
- [Usar `AWS::CloudFormation::Init` para implantar aplicações de software em instâncias do Amazon EC2](#)
- [Usar os scripts auxiliares mais recentes](#)
- [Validar modelos antes de usá-los](#)
- [Usando YAML ou JSON para criação de modelos](#)
- [Implemente uma estratégia abrangente de marcação](#)
- [Aproveite as macros de modelo para transformações avançadas](#)

Gerenciamento de pilhas

- [Gerenciar todos os recursos de pilha via CloudFormation](#)

- [Criar conjuntos de alteração antes de atualizar as pilhas](#)
- [Use as políticas de pilha para proteger os recursos](#)
- [Use o AWS CloudTrail para registrar chamadas do CloudFormation](#)
- [Usar revisões de código e controles de revisão para gerenciar seus modelos](#)
- [Atualizar suas instâncias do Amazon EC2 regularmente](#)
- [Use a detecção de desvios regularmente](#)
- [Configurar gatilhos de reversão para recuperação automática](#)
- [Implemente estratégias eficazes de refatoração de pilhas](#)
- [Use os hooks do CloudFormation para gerenciamento do ciclo de vida](#)

Ferramentas de criação

- [Use o Gerador de IaC para criar modelos com recursos existentes](#)
- [Use AWS Infrastructure Composer para design de modelo visual](#)
- [Considere o uso do AWS Cloud Development Kit \(AWS CDK\) para uma infraestrutura complexa](#)

Segurança e conformidade

- [Usar o IAM para controlar o acesso](#)
- [Aplicação do princípio de privilégio mínimo](#)
- [Parâmetros confidenciais seguros](#)
- [Implemente a política como código com o AWS CloudFormation Guard](#)

Encurtar o ciclo de feedback para melhorar a velocidade de desenvolvimento

Adote práticas e ferramentas que ajudam a reduzir o ciclo de feedback da infraestrutura que você descreve com os modelos do CloudFormation. Isso inclui fazer mais cedo o linting e os testes de modelos em sua estação de trabalho. Com isso, você descobre possíveis problemas de sintaxe e configuração antes mesmo de enviar suas contribuições para um repositório de código-fonte. A descoberta precoce desses problemas ajuda a evitar que eles atinjam os ambientes formais do ciclo de vida, como desenvolvimento, garantia de qualidade e produção. Essa estratégia de testar cedo e antecipar-se à falha tem a vantagem de reduzir o tempo de espera de retrabalho, diminuir as possíveis áreas de impacto e aumentar seu nível de confiança no sucesso das operações de provisionamento.

As opções de ferramentas que ajudam você a adotar a prática de antecipar-se à falha incluem as ferramentas de linha de comando [CloudFormation Linter](#) (`cfn-lint`) e [TaskCat](#). A ferramenta `cfn-lint` permite validar os modelos do CloudFormation em relação à [Especificação de recursos do CloudFormation](#). Isso inclui verificar se foram usados valores válidos nas propriedades dos recursos, bem como as práticas recomendadas. Os plug-ins para o `cfn-lint` estão [disponíveis para vários editores de código](#), o que permite que você visualize os problemas no editor e obtenha feedback direto da ferramenta de linter. Você também pode optar por integrar o `cfn-lint` à configuração do repositório de código-fonte, para poder validar os modelos quando confirmar suas contribuições. Para obter mais informações, consulte [Validação pré-confirmação do Git de modelos do CloudFormation com o cfn-lint](#). Depois de realizar o linting inicial e corrigir quaisquer problemas que o `cfn-lint` possa ter apontado, você poderá usar o TaskCat para testar seus modelos criando programaticamente pilhas nas Regiões da AWS de sua escolha. O TaskCat também gera um relatório com os resultados de aprovação/reprovação para cada região escolhida.

Para obter uma demonstração prática passo a passo de como usar as duas ferramentas para encurtar o ciclo de feedback, siga o [Laboratório de linting e testes](#) do [Workshop do CloudFormation](#).

Organizar pilhas por ciclo de vida e propriedade

Use o ciclo de vida e a propriedade dos recursos da AWS para ajudar você a decidir quais deles devem estar contidos em cada pilha. Inicialmente, é possível colocar todos os recursos em uma pilha, porém, à medida que ela cresce em escala e aumenta em escopo, o gerenciamento de uma única pilha pode ser trabalhoso e demorado. Ao agrupar recursos com ciclos de vida e propriedade comuns, os proprietários podem fazer alterações no conjunto de recursos deles usando um processo e uma programação próprios sem afetar outros recursos.

Por exemplo, imagine uma equipe de desenvolvedores e engenheiros que possuem um site hospedado em instâncias do Amazon EC2 Auto Scaling atrás de um balanceador de carga. Como o site tem seu próprio ciclo de vida e é mantido pela equipe do site, é possível criar uma pilha para o site e seus respectivos recursos. Agora, imagine que o site também usa bancos de dados de backend, em que os bancos de dados estão em uma pilha separada que cujas propriedade e manutenção são dos administradores de bancos de dados. Sempre que a equipe do site ou a equipe do banco de dados precisa atualizar seus recursos, eles podem fazê-lo sem afetar a pilha uns dos outros. Se todos os recursos estavam em uma única pilha, a coordenação e a comunicação de atualizações pode ser difícil.

Para obter mais orientações sobre como organizar suas pilhas, você pode usar duas estruturas comuns: uma arquitetura em várias camadas e a arquitetura orientada a serviços (SOA).

A arquitetura em camadas organiza pilhas em várias camadas horizontais que se sobrepõem, em que cada camada tem uma dependência na camada diretamente abaixo dela. Você pode ter uma ou mais pilhas em cada camada, mas dentro de cada camada, suas pilhas devem ter recursos da AWS com ciclos de vida e propriedade semelhantes.

Com uma arquitetura orientada a serviços, é possível organizar grandes problemas de negócios em partes gerenciáveis. Cada uma dessas partes é um serviço que tem uma finalidade claramente definida e representa uma unidade independente de funcionalidade. Você pode mapear esses serviços para uma pilha, em que cada pilha tenha seu próprio ciclo de vida e proprietários. Esses serviços (pilhas) podem ser conectados para que possam interagir uns com os outros.

Use referências entre pilhas para retornar o valor de uma saída exportada por outra pilha

Ao organizar os recursos da AWS com base no ciclo de vida e na propriedade, talvez você queira criar uma pilha que use recursos existentes em outra pilha. Você pode codificar valores ou usar parâmetros de entrada para transmitir os nomes e os IDs dos recursos. No entanto, esses métodos podem dificultar a reutilização dos modelos ou aumentar a sobrecarga para colocar uma pilha em execução. Em vez disso, use as referências entre pilhas para retornar o valor de uma saída exportada por outra pilha para que outras pilhas possam usá-la. As pilhas podem usar os recursos exportados chamando-os com a função `Fn::ImportValue`.

Por exemplo, você pode ter uma pilha de rede que inclui uma VPC, um security group e uma sub-rede. Você deseja que todas as aplicações web públicas usem esses recursos. Ao exportar os recursos, você permite que todas as pilhas com aplicações web públicas usem tais recursos. Para obter mais informações, consulte [Obter resultados exportados de uma pilha do CloudFormation implantada](#).

Use os StackSets do CloudFormation para implantações em várias contas e em várias regiões

Os StackSets do CloudFormation amplia a capacidade das pilhas, permitindo que você crie, atualize ou exclua pilhas em várias contas e regiões com uma única operação. Use StackSets para implantar componentes comuns de infraestrutura, controles de conformidade ou serviços compartilhados em toda a sua organização.

Ao usar StackSets, implemente permissões gerenciadas por serviços com AWS Organizations para simplificar o gerenciamento de permissões. Essa abordagem permite que você implante os

StackSets em contas da sua organização sem a necessidade de configurar manualmente os perfis do IAM em cada conta.

Para obter mais informações sobre os StackSets, consulte [Conceitos de StackSets](#).

Verificar cotas para todos os tipos de recursos

Antes de executar uma pilha, certifique-se de que pode criar todos os recursos que desejados sem atingir os limites de sua conta da AWS. Se você atingir um limite, o CloudFormation não criará a pilha com êxito até que você aumente sua cota ou exclua recursos adicionais. Cada serviço pode ter vários limites que você deve levar em conta antes de executar uma pilha. Por exemplo, por padrão, você pode executar somente 2000 pilhas do CloudFormation por região em sua Conta da AWS. Para mais informações sobre limites e como aumentar os limites padrão, consulte [Cotas de serviços da AWS](#), no Referência geral da AWS.

Reutilizar modelos para replicar pilhas em vários ambientes

Depois de configurar suas pilhas e recursos, você pode reutilizar os modelos para replicar sua infraestrutura em vários ambientes. Por exemplo, você pode criar ambientes para desenvolvimento, teste e produção para testar as alterações antes de implementá-las na produção. Para tornar os modelos reutilizáveis, use as seções de parâmetros, mapeamentos e condições para personalizar suas pilhas ao criá-las. Por exemplo, para ambientes de desenvolvimento, é possível especificar um tipo de instância de custo mais baixo em comparação com o ambiente de produção, mas todas as outras configurações e definições são as mesmas. Para obter mais informações sobre parâmetros, mapeamentos e condições, consulte [Seções de modelos do CloudFormation](#).

Usar módulos para reutilizar configurações de recursos

À medida que sua infraestrutura cresce, podem surgir padrões comuns em que é possível declarar os mesmos componentes em cada um dos modelos. Os módulos são uma maneira de empacotar configurações de recursos para inclusão entre modelos de pilha, de modo transparente, gerenciável e repetível. Os módulos podem encapsular configurações comuns de serviço e práticas recomendadas como blocos de construção modulares e personalizáveis para você incluir em seus modelos de pilha.

Esses blocos de construção podem ser para um único recurso, como práticas recomendadas para definir uma instância do Amazon Elastic Compute Cloud (Amazon EC2), ou podem ser para vários recursos, para definir padrões comuns de arquitetura de aplicações. Esses blocos fundamentais

podem ser aninhados a outros módulos, para que você possa empilhar suas práticas recomendadas em blocos fundamentais de nível superior. Módulos do CloudFormation estão disponíveis no [Registro do CloudFormation](#), para que você possa utilizá-los como um recurso nativo. Quando você usa um módulo do CloudFormation, o modelo de módulo é expandido para o modelo de consumo, o que possibilita acessar os recursos dentro do módulo usando uma função [Ref](#) ou [Fn::GetAtt](#). Para obter mais informações, consulte [Criar configurações de recursos reutilizáveis que podem ser incluídas em modelos com os módulos do CloudFormation](#).

Adote a infraestrutura como práticas de código

Trate seus modelos do CloudFormation como código implementando práticas de infraestrutura como código (IaC). Armazene seus modelos em sistemas de controle de versão, implemente análises de código e use testes automatizados para validar as alterações. Essa abordagem garante consistência, melhora a colaboração e fornece uma trilha de auditoria para alterações na infraestrutura.

Considere implementar pipelines de CI/CD para seu código de infraestrutura para automatizar o teste e a implantação de seus modelos do CloudFormation. Ferramentas como AWS CodePipeline, AWS CodeBuild e AWS CodeDeploy podem ser usadas para criar fluxos de trabalho automatizados para suas implantações de infraestrutura.

Para obter mais informações sobre a implementação das práticas recomendadas de IaC, consulte [Usando o AWS CloudFormation como uma ferramenta de IaC](#).

Para obter mais informações sobre o uso da entrega contínua com o CloudFormation, consulte [Entrega contínua com o CodePipeline](#).

Não incorporar credenciais em seus modelos

Em vez de incorporar informações confidenciais nos modelos do CloudFormation, é recomendável usar referências dinâmicas no modelo de pilha.

As referências dinâmicas fornecem uma maneira compacta e avançada para você fazer referência a valores externos que são armazenados e gerenciados em outros serviços, como o Repositório de parâmetros do AWS Systems Manager ou o AWS Secrets Manager. Quando você usa uma referência dinâmica, o CloudFormation recupera o valor da referência especificada, se necessário, durante operações de pilha e conjunto de alterações e passa o valor para o recurso apropriado. No entanto, o CloudFormation nunca armazena o valor real do parâmetro. Para obter mais informações, consulte [Usar referências dinâmicas para especificar valores de modelo](#).

O [AWS Secrets Manager](#) ajuda você a criptografar, armazenar e recuperar credenciais com segurança para bancos de dados e outros serviços. O [Repositório de parâmetros do AWS Systems Manager](#) fornece armazenamento hierárquico seguro para o gerenciamento de dados de configuração.

Para obter mais informações sobre a definição de parâmetros do modelo, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Usar tipos de parâmetros específicos da AWS

Se o modelo exige entradas de valores existentes específicos da AWS, como IDs existentes do Amazon Virtual Private Cloud ou um nome de par de chaves do Amazon EC2, use os tipos de parâmetros específicos da AWS. Por exemplo, é possível especificar um parâmetro como tipo `AWS::EC2::KeyPair::KeyName`, que usa um nome de par de chaves existente que está em sua Conta da AWS e na região em que você está criando a pilha. O CloudFormation pode validar rapidamente valores de tipos de parâmetros específicos da AWS antes de criar a pilha. Além disso, se você usar o console do CloudFormation, o CloudFormation mostrará uma lista suspensa de valores válidos, de modo que você não precisará examinar ou memorizar os IDs da VPC corretos ou os nomes de pares de chaves. Para obter mais informações, consulte [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#).

Usar restrições de parâmetros

Com restrições, é possível descrever valores de entrada permitidos para que o CloudFormation detecte todos os valores inválidos antes de criar uma pilha. É possível definir restrições como um tamanho mínimo ou máximo e padrões permitidos. Por exemplo, você pode definir as restrições em um valor de nome de usuário do banco de dados para que ele tenha no mínimo oito caracteres e contenha apenas caracteres alfanuméricos. Para obter mais informações, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Use pseudoparâmetros para promover a portabilidade

Você pode usar [pseudoparâmetros](#) em seus modelos como argumentos para [funções intrínsecas](#), como `Ref` e `Fn::Sub`. Os pseudoparâmetros são parâmetros que são predefinidos pelo CloudFormation. Você não os declarará em seu modelo. O uso de pseudoparâmetros em funções intrínsecas aumenta a portabilidade de seus modelos de pilha entre regiões e contas.

Por exemplo, imagine que você deseja criar um modelo em que, para uma determinada propriedade de recurso, seja necessário especificar o [nome do recurso da Amazon](#) (ARN) de outro recurso existente. Nesse caso, o recurso existente é um recurso do [AWS Systems Manager Parameter Store](#) com o seguinte ARN: `arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter`. Você precisará adaptar o [formato do ARN](#) à partição, região e ID da conta da AWS de destino. Em vez de realizar a codificação rígida desses valores, você pode usar os pseudoparâmetros `AWS::Partition`, `AWS::Region` e `AWS::AccountId` para tornar seu modelo mais portátil. Nesse caso, o exemplo a seguir mostra como encadear elementos em um ARN com o CloudFormation: `!Sub 'arn:${AWS::Partition}:ssm:${AWS::Region}:${AWS::AccountId}:parameter/MySampleParameter'`.

Como outro exemplo, suponha que você deseje compartilhar recursos ou configurações em várias pilhas. Neste exemplo, suponha que você tenha criado uma [sub-rede](#) para sua VPC e, em seguida, exportado seu ID para uso com outras pilhas na mesma região e Conta da AWS. Em outra pilha, você faz referência ao valor exportado do ID da sub-rede ao descrever uma instância do Amazon EC2. Para obter um exemplo detalhado sobre o uso do campo de saída `Export` e da função intrínseca `Fn::ImportValue`, consulte [Consultar saídas de recurso em outra pilha do CloudFormation](#).

As exportações de pilha devem ser exclusivas por conta e região. Portanto, nesse caso, você pode usar o pseudoparâmetro `AWS::StackName` para criar um prefixo para sua exportação. Como os nomes de pilha também devem ser exclusivos por conta e região, o uso desse pseudoparâmetro como prefixo aumenta a possibilidade de haver um nome de exportação exclusivo e, ao mesmo tempo, promove uma abordagem reutilizável entre as pilhas das quais você exporta valores. Alternativamente, você pode usar um prefixo de sua escolha.

Usar **AWS::CloudFormation::Init** para implantar aplicações de software em instâncias do Amazon EC2

Ao executar pilhas, você pode instalar e configurar aplicativos de software em instâncias do Amazon EC2 usando o script auxiliar `cfn-init` e o recurso `AWS::CloudFormation::Init`. Ao usar `AWS::CloudFormation::Init`, você pode descrever as configurações que deseja em vez de criar o script de etapas procedimentais. Também é possível atualizar as configurações sem recriar instâncias. E se algo errado der errado com a configuração, o CloudFormation gerará logs que podem ser usados para investigar problemas.

Em seu modelo, especifique estados de instalação e configuração no recurso

`AWS::CloudFormation::Init`. Para obter uma explicação detalhada que mostra como usar `cfn-init` e `AWS::CloudFormation::Init`, consulte [Implantar aplicações no Amazon EC2](#).

Usar os scripts auxiliares mais recentes

Os scripts auxiliares do CloudFormation são atualizados periodicamente. Certifique-se de incluir o seguinte comando na propriedade `UserData` do seu modelo antes de chamar os scripts auxiliares para garantir que suas instâncias executadas obtenham os scripts auxiliares mais recentes:

```
yum install -y aws-cfn-bootstrap
```

Para obter mais informações sobre como receber os scripts auxiliares mais recentes, consulte a [Referência de scripts auxiliares do CloudFormation](#) no Guia de referência de modelos do CloudFormation.

Validar modelos antes de usá-los

Antes de usar um modelo para criar ou atualizar uma pilha, você pode usar o CloudFormation para validá-lo. A validação de um modelo pode ajudá-lo a perceber alguns erros de sintaxe e semântica, como dependências circulares, antes que o CloudFormation crie algum recurso. Se você usar o console do CloudFormation, ele validará automaticamente o modelo depois que você especificar os parâmetros de entrada. Para a AWS CLI ou a API do CloudFormation, use o comando [validate-template](#) da CLI ou a operação [ValidateTemplate](#) da API.

Durante a validação, o CloudFormation primeiro verifica se o modelo é JSON válido. Caso não seja, o CloudFormation verifica se o modelo é YAML válido. Se as duas verificações falharem, o CloudFormation retornará um erro de validação de modelo.

Validar modelos para a conformidade com a política da organização

Você também pode validar a conformidade do seu modelo com as diretrizes da política da organização. O AWS CloudFormation Guard (`cfn-guard`) é uma ferramenta de interface de linha de comando (CLI) de código aberto que fornece uma linguagem de política como código para definir regras que podem verificar as configurações de recursos obrigatórios e proibidos. Ele permite que você valide seus modelos em relação a essas regras. Por exemplo, os administradores podem criar regras para garantir que os usuários sempre criem buckets do Amazon S3 criptografados.

É possível usar o `cfn-guard` localmente, durante a edição de modelos, ou automaticamente, como parte de um pipeline de CI/CD para interromper a implantação de recursos não compatíveis.

Além disso, o `cfn-guard` inclui um recurso, `rulegen`, que permite extrair regras de modelos existentes compatíveis do CloudFormation.

Para obter mais informações, consulte o repositório [cfn-guard](#) no GitHub.

Usando YAML ou JSON para criação de modelos

O CloudFormation é compatível com os formatos YAML e JSON para modelos. Cada um tem suas vantagens e a escolha depende de suas necessidades específicas:

Use YAML quando

- Você prioriza a legibilidade e a capacidade de manutenção humana
- Você deseja incluir comentários para documentar seu modelo
- Você está trabalhando em modelos complexos com estruturas aninhadas
- Você deseja usar atributos específicos do YAML, como âncoras e aliases, para reduzir a repetição

Use JSON quando:

- Você precisa se integrar com ferramentas ou sistemas que preferem JSON
- Você está trabalhando com geração ou manipulação programática de modelos
- Você precisa de uma validação rigorosa de dados

O YAML geralmente é recomendado para criação manual de modelos devido à sua legibilidade e suporte a comentários. É particularmente útil para modelos complexos em que a estrutura baseada em recuo ajuda a visualizar hierarquias de recursos. O JSON pode ser vantajoso em fluxos de trabalho automatizados ou ao trabalhar com APIs que esperam entrada JSON. Também é benéfico quando você precisa garantir a adesão estrita a uma estrutura específica. Independentemente do formato escolhido, concentre-se na criação de modelos bem estruturados, documentados e de fácil manutenção. Se estiver usando o YAML, aproveite seus atributos, como âncoras e aliases, para reduzir a repetição e melhorar a capacidade de manutenção.

Implemente uma estratégia abrangente de marcação

Implemente uma estratégia de marcação consistente para todos os recursos criados pelos seus modelos do CloudFormation. As tags ajudam na organização de recursos, alocação de custos, controle de acesso e automação. Considere incluir tags para ambiente, proprietário, centro de custos, aplicação e finalidade.

Use a propriedade de `AWS::CloudFormation::Stack` do recurso do Tags para aplicar tags a todos os recursos compatíveis em uma pilha. Você também pode usar a propriedade `TagSpecifications` disponível em vários tipos de recursos para aplicar tags durante a criação do recurso.

Para obter mais informações sobre marcação, consulte [Recursos de tag](#).

Aproveite as macros de modelo para transformações avançadas

As macros do CloudFormation permitem que você realize processamento personalizado em modelos, desde ações simples, como operações de localizar e substituir, até transformações complexas que geram recursos adicionais. Use macros para ampliar os recursos dos modelos do CloudFormation e implementar padrões reutilizáveis em toda a sua organização.

O AWS Serverless Application Model é um exemplo de macro que simplifica o desenvolvimento de aplicações com tecnologia sem servidor. Considere criar macros personalizadas para padrões e requisitos específicos da organização.

Para obter mais informações sobre o uso de macros em seus modelos, consulte [Visão geral das macros do CloudFormation](#).

Gerenciar todos os recursos de pilha via CloudFormation

Depois de iniciar uma pilha, use o [console](#) do CloudFormation, a [API](#) ou a [AWS CLI](#) para atualizar recursos na sua pilha. Não faça alterações nos recursos da pilha fora do CloudFormation. Isso pode criar uma incompatibilidade entre seu modelo de pilha e o estado atual dos recursos da pilha, o que pode causar erros se você atualizar ou excluir a pilha. Isso é conhecido como deriva. Se uma alteração for feita em um recurso fora do modelo do CloudFormation e você atualizar a pilha, as alterações feitas diretamente no recurso serão descartadas, e a configuração do recurso será revertida para a configuração no modelo.

Para obter mais informações sobre o desvio, consulte [O que é desvio?](#).

Para obter mais informações sobre como atualizar pilhas, consulte [Atualizar uma pilha do CloudFormation](#).

Criar conjuntos de alteração antes de atualizar as pilhas

Conjuntos de alterações permitem que você veja como as alterações propostas para uma pilha podem afetar os recursos em execução antes de implementá-las. O CloudFormation não fará alterações na pilha até que você execute o conjunto de alterações, permitindo que você decida se deseja continuar com as alterações propostas ou criar outro conjunto de alterações.

Use conjuntos de alterações para verificar como as alterações podem afetar os recursos em execução, especialmente em relação a recursos críticos. Por exemplo, se você alterar o nome de uma instância de banco de dados do Amazon RDS, o CloudFormation criará um novo banco de dados e excluirá o antigo. Você perderá os dados do banco de dados antigo, a menos que já tenha feito o backup dele. Se você gerar um conjunto de alterações, verá que sua alteração substituirá seu banco de dados. Isso pode ajudar no planejamento antes de atualizar a pilha. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

Use as políticas de pilha para proteger os recursos

As políticas de pilha ajudam a proteger recursos de pilha críticos contra atualizações não intencionais capazes de fazer com que os recursos sejam interrompidos ou até mesmo substituídos. Uma política de pilha é um documento JSON que descreve quais ações de atualização podem ser executadas nos recursos designados. Especifique uma política de pilha sempre que criar uma pilha com recursos críticos.

Durante uma atualização de pilha, é necessário especificar explicitamente os recursos protegidos que deseja atualizar, caso contrário, nenhuma alteração será feita nos recursos protegidos. Para obter mais informações, consulte [Impedir atualizações nos recursos de pilha](#).

Use o AWS CloudTrail para registrar chamadas do CloudFormation

O AWS CloudTrail rastreia qualquer pessoa que faça chamadas de API do CloudFormation na Conta da AWS. As chamadas de API são registradas em log sempre que alguém usa a API do CloudFormation, o console do CloudFormation, um console de back-end ou comandos da AWS CLI do CloudFormation. Ative o registro em log e especifique um bucket do Amazon S3 para armazenar os logs. Dessa forma, se você precisar, poderá auditar quem fez qual chamada do CloudFormation em sua conta.

Para obter mais informações, consulte [Registrar em log chamadas de API do CloudFormation com o AWS CloudTrail](#).

Usar revisões de código e controles de revisão para gerenciar seus modelos

Os modelos de pilha descrevem a configuração de seus recursos da AWS, como seus respectivos valores de propriedades. Para analisar as alterações e manter um histórico exato de seus recursos, use revisões de código e controles de revisão. Esses métodos podem ajudá-lo a rastrear as alterações entre versões diferentes de seus modelos, o que pode ajudar você a rastrear alterações nos recursos de sua pilha. Além disso, com a manutenção de um histórico, sempre é possível reverter a pilha para uma determinada versão do modelo.

Atualizar suas instâncias do Amazon EC2 regularmente

Em todas as instâncias Windows do Amazon EC2 e instâncias Linux do Amazon EC2 criadas com o CloudFormation, execute regularmente o comando `yum update` para atualizar o pacote RPM. Isso garante que você obtenha as mais recentes correções e atualizações de segurança.

Use a detecção de desvios regularmente

Use regularmente o atributo de detecção de desvios do CloudFormation para identificar recursos que foram modificados fora do gerenciamento do CloudFormation. Detectar e resolver desvios ajuda a manter a integridade de sua abordagem de infraestrutura como código e garante que seus modelos reflitam com precisão o estado dos recursos implantados.

Considere implementar a detecção automatizada de desvios como parte de seus procedimentos operacionais. Você pode usar as funções do AWS Lambda acionadas pelas regras do Amazon EventBridge para verificar periodicamente se há desvios e notificar sua equipe quando forem detectadas discrepâncias.

Para obter mais informações sobre desvios, consulte [Detectar alterações de configuração não gerenciadas em pilhas e recursos com a detecção de desvios](#).

Configurar gatilhos de reversão para recuperação automática

Use os gatilhos de reversão para especificar os alarmes do Amazon CloudWatch que o CloudFormation deve monitorar durante as operações de criação e atualização da pilha. Se algum

dos alarmes especificados entrar no estado de ALARM, o CloudFormation reverte automaticamente toda a operação da pilha, ajudando a garantir que sua infraestrutura permaneça estável.

Configure acionadores de reversão para métricas críticas, como taxas de erro da aplicação, utilização de recursos do sistema ou métricas comerciais personalizadas que indicam a integridade de sua aplicação e da infraestrutura.

Para obter mais informações sobre gatilhos de reversão, consulte [Reverter sua pilha em caso de violação de alarme](#).

Implemente estratégias eficazes de refatoração de pilhas

À medida que sua infraestrutura evolui, talvez seja necessário refatorar suas pilhas do CloudFormation para melhorar a capacidade de manutenção, reduzir a complexidade ou se adaptar às mudanças nos requisitos. A refatoração de pilhas exige a reestruturação de seus modelos e recursos, preservando o comportamento e a funcionalidade externos. É vantajoso usar a refatoração de pilhas com o CloudFormation das seguintes formas:

- Divisão de pilhas monolíticas: divida pilhas grandes e complexas em pilhas menores e mais gerenciáveis, organizadas por ciclo de vida ou propriedade
- Consolidação de recursos relacionados: combine recursos relacionados de várias pilhas em uma única pilha coesa para simplificar o gerenciamento
- Extração de componentes reutilizáveis: mova padrões comuns para módulos ou pilhas aninhadas para promover a reutilização e a consistência
- Aperfeiçoamento da organização dos recursos: reestruture os recursos dentro de uma pilha para refletir melhor seus relacionamentos e dependências

Para obter mais informações sobre como refatorar suas pilhas do CloudFormation, consulte [Refatoração de pilhas](#).

Use os hooks do CloudFormation para gerenciamento do ciclo de vida

Os hooks do CloudFormation fornecem código que inspeciona proativamente a configuração de seus recursos da AWS antes do provisionamento e executa verificações de validação complexas. Eles verificam se seus recursos, pilhas e conjuntos de alterações estão em conformidade com as necessidades de segurança, operacionais e de otimização de custos de sua organização. Eles

fornece avisos antes do provisionamento de um recurso ou falham na operação e a interrompem completamente, dependendo de como ela foi configurada. Violações e avisos são registrados no Amazon CloudWatch para fornecer visibilidade sobre implantações não compatíveis.

Para obter mais informações sobre essas práticas recomendadas para Hooks, consulte [Conceitos de Hooks do AWS CloudFormation](#).

Para obter mais informações sobre o que os Hooks podem fazer pelos seus recursos do CloudFormation, consulte [O que são Hooks do AWS CloudFormation?](#)

Use o Gerador de IaC para criar modelos com recursos existentes

O gerador de IaC do CloudFormation (infraestrutura como código) ajuda você a criar modelos do CloudFormation a partir de seus recursos da AWS existentes. Essa capacidade é particularmente útil quando você precisa replicar a infraestrutura existente, documentar recursos criados manualmente ou colocar recursos anteriormente não gerenciados sob o gerenciamento do CloudFormation. O gerador de IaC é útil para criar seus modelos do CloudFormation das seguintes maneiras:

- Criação acelerada de modelos: gere modelos com recursos existentes em vez de escrevê-los do zero
- Infraestrutura consistente: garanta que os novos ambientes correspondam aos existentes usando modelos gerados como ponto de partida
- Migração para a infraestrutura como código: coloque gradualmente os recursos criados manualmente sob o gerenciamento do CloudFormation
- Documentação: crie um registro de sua infraestrutura existente em formato de modelo

Para obter mais informações sobre o Gerador de IaC, consulte [Gerar modelos de recursos existentes com o gerador de IaC](#).

Use AWS Infrastructure Composer para design de modelo visual

O AWS Infrastructure Composer é uma ferramenta de design visual que ajuda a criar, visualizar e modificar modelos do CloudFormation usando uma interface de arrastar e soltar. Isso pode ser particularmente benéfico ao usar o CloudFormation das seguintes maneiras:

- Planejamento de arquitetura: projete e valide arquiteturas de infraestrutura antes da implementação

- Modernização do modelo: visualize os modelos existentes para entender sua estrutura e identificar oportunidades de melhoria
- Treinamento e integração: ajude os novos membros da equipe a entender os conceitos do CloudFormation e as relações de serviço por meio do aprendizado visual da AWS
- Comunicação com as partes interessadas: apresente projetos de infraestrutura para partes interessadas não técnicas usando representações visuais claras
- Análises de conformidade: use diagramas visuais para facilitar as análises de segurança e conformidade de seus projetos de infraestrutura
- Análises de conformidade: use diagramas visuais para facilitar as análises de segurança e conformidade de seus projetos de infraestrutura

Para obter mais informações sobre o Compositor de Infraestrutura, consulte [O que é o AWS Infrastructure Composer?](#).

Considere o uso do AWS Cloud Development Kit (AWS CDK) para uma infraestrutura complexa

Para requisitos de infraestrutura complexas, considere usar o CDK para definir seus recursos de nuvem usando linguagens de programação familiares, como TypeScript, Python, Java e .NET. O AWS CDK gera modelos do CloudFormation a partir do seu código, permitindo que você aproveite todos os recursos do CloudFormation enquanto usa as abstrações e as construções de programação de sua linguagem preferida.

O AWS CDK fornece construções de alto nível que encapsulam as práticas recomendadas e simplificam a definição de padrões comuns de infraestrutura. Isso pode reduzir significativamente a quantidade de código necessário para definir sua infraestrutura, garantindo ao mesmo tempo a adesão às práticas recomendadas.

Para obter mais informações sobre CDK, consulte o [AWS Cloud Development Kit \(AWS CDK\)](#)..

Usar o IAM para controlar o acesso

O IAM é um serviço da AWS que você pode usar para gerenciar usuários e suas respectivas permissões na AWS. É possível usar o IAM com o CloudFormation para especificar quais ações do CloudFormation os usuários podem executar, como visualizar os modelos de pilha, criar pilhas ou excluir pilhas. Além disso, qualquer pessoa que gerencie as pilhas do CloudFormation precisará de

permissões para os recursos dentro dessas pilhas. Por exemplo, se os usuários quiserem usar o CloudFormation para executar, atualizar ou encerrar instâncias do Amazon EC2, eles precisarão ter permissão para chamar as ações relevantes do Amazon EC2.

Na maioria dos casos, os usuários precisam de acesso total para gerenciar todos os recursos em um modelo. O CloudFormation faz chamadas para criar, modificar e excluir esses recursos em nome dele. Para permissões separadas entre um usuário e o serviço do CloudFormation, use um perfil de serviço. O CloudFormation usa a política do perfil de serviço para fazer chamadas em vez da política do usuário. Para obter mais informações, consulte [Função de serviço do CloudFormation](#).

Aplicação do princípio de privilégio mínimo

Ao configurar perfis do IAM para os perfis serviço do CloudFormation ou para recursos criados por seus modelos, sempre aplique o princípio do privilégio mínimo. Conceda somente as permissões necessárias para a funcionalidade pretendida e evite usar permissões curinga sempre que possível.

Use o analisador de acesso do IAM para analisar as permissões concedidas aos seus perfis de serviço do CloudFormation e identificar permissões não utilizadas que podem ser removidas. Revise e atualize regularmente as políticas do IAM para garantir que permaneçam alinhadas com seus requisitos de segurança.

Parâmetros confidenciais seguros

Para informações confidenciais, como senhas, chaves de API e outros segredos, use o Parameter Store do AWS Systems Manager ou o AWS Secrets Manager em vez de incorporá-las diretamente em seus modelos. Use referências dinâmicas em seus modelos para recuperar com segurança esses valores durante as operações de pilha.

Ao usar parâmetros em seus modelos, defina a propriedade NoEcho como `true` para parâmetros confidenciais para evitar que seus valores sejam exibidos no console, nas respostas da API ou na saída da CLI. Lembre-se de que o NoEcho não impede que o valor seja registrado se for passado para outros serviços ou recursos que possam registrar o valor.

Para obter mais informações sobre como usar o Parameter Store do AWS Systems Manager com o CloudFormation, consulte [Obter um valor em texto simples do Parameter Store do AWS Systems Manager](#).

Para obter mais informações sobre o uso da propriedade NoEcho, consulte a [Sintaxe de parâmetros do modelo do CloudFormation](#).

Para obter mais informações sobre como usar o AWS Secrets Manager com o CloudFormation, consulte [Criar segredos do AWS Secrets Manager no AWS CloudFormation](#).

Implemente a política como código com o AWS CloudFormation Guard

O AWS CloudFormation Guard(cfn-guard) é uma ferramenta de código aberto de política como código que permite definir e aplicar regras para seus modelos do CloudFormation. Use o cfn-guard para garantir que seus modelos estejam em conformidade com as políticas organizacionais, as práticas recomendadas de segurança e os requisitos de governança.

Integre o cfn-guard aos seus pipelines de CI/CD para validar automaticamente os modelos de acordo com suas regras de política antes da implantação. Isso ajuda a evitar que recursos não compatíveis sejam implantados em seu ambiente e fornece feedback antecipado aos desenvolvedores sobre violações de políticas.

Para obter mais informações sobre o Guard, consulte [O que é o AWS CloudFormation Guard?](#)

Trabalhar com modelos do CloudFormation

Um modelo do AWS CloudFormation define os recursos da AWS que você deseja criar, atualizar ou excluir como parte de uma pilha. Ele consiste em várias seções, mas a única seção necessária é a seção [Resources](#), que deve declarar pelo menos um recurso.

É possível criar modelos usando os seguintes métodos:

- AWS Infrastructure Composer: uma interface visual para criar modelos.
- Editor de texto: escreva modelos diretamente na sintaxe JSON ou YAML.
- Gerador de IaC: gere modelos com base em recursos provisionados em sua conta que não são gerenciados pelo CloudFormation no momento. O gerador de IaC funciona com uma grande variedade de tipos de recursos compatíveis com a API do Cloud Control na sua região.

Esta seção fornece um guia abrangente sobre como usar as diferentes seções de um modelo do CloudFormation e como começar a criar modelos de pilha. Ela abrange os seguintes tópicos:

Tópicos

- [Onde os modelos são armazenados](#)
- [Validar modelos](#)
- [Conceitos básicos de modelos](#)
- [Exemplos de modelo](#)
- [Formato de modelo do CloudFormation](#)
- [Seções de modelos do CloudFormation](#)
- [Crie modelos visualmente com o Infrastructure Composer](#)
- [Servidor de linguagens do AWS CloudFormation](#)
- [Gerar modelos a partir de recursos existentes com o gerador de IaC](#)
- [Obter valores armazenados em outros serviços usando referências dinâmicas](#)
- [Obter valores da AWS usando pseudoparâmetros](#)
- [Obter resultados exportados de uma pilha do CloudFormation implantada](#)
- [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#)

- [Orientações passo a passo do CloudFormation](#)
- [Trechos de modelo do CloudFormation](#)
- [Implantar pilhas baseadas em Windows usando o CloudFormation](#)
- [Amplie os recursos do seu modelo com os tipos de recursos fornecidos pelo CloudFormation](#)
- [Criar configurações de recursos reutilizáveis que podem ser incluídas em modelos com os módulos do CloudFormation](#)

Onde os modelos são armazenados

Bucket do Amazon S3

Você pode armazenar modelos do CloudFormation em um bucket do Amazon S3. Ao criar ou atualizar uma pilha, é possível especificar o URL do modelo no S3 em vez de carregá-lo diretamente.

Se você fizer upload de modelos diretamente por meio do Console de gerenciamento da AWS ou AWS CLI, um bucket do S3 será criado automaticamente para você. Para obter mais informações, consulte [Criar uma pilha no console do CloudFormation](#).

Repositório Git

Com o [Git sync](#), é possível armazenar modelos em um repositório Git. Ao criar ou atualizar uma pilha, é possível especificar o local do repositório Git e a ramificação que contém o modelo, em vez de carregá-lo diretamente ou fazer referência a um URL do S3. O CloudFormation monitora automaticamente o repositório e a ramificação especificados em busca de alterações no modelo. Para obter mais informações, consulte [Criar uma pilha com base no código-fonte do repositório com a sincronização do Git](#).

Validar modelos

Validação de sintaxe

É possível verificar a sintaxe JSON ou YAML do seu modelo usando o comando [validate-template](#) da CLI ou especificando o modelo no console. O console realiza a validação automaticamente. Para obter mais informações, consulte [Criar uma pilha no console do CloudFormation](#).

No entanto, esses métodos apenas verificam a sintaxe do seu modelo e não validam os valores de propriedade especificados para um recurso.

Ferramentas adicionais de validação

Para validações mais complexas e verificações de melhores práticas, é possível usar ferramentas adicionais, como:

- [CloudFormation Linter \(cfn-lint\)](#): valide modelos em relação aos [esquemas do provedor de recursos do CloudFormation](#). Isso inclui verificar se foram usados valores válidos para propriedades dos recursos e práticas recomendadas.
- [CloudFormation Rain \(rain fmt\)](#): formate seus modelos do CloudFormation em um padrão consistente ou reformate um modelo de JSON para YAML (ou YAML para JSON). Ele preserva os comentários ao usar o YAML e muda o uso de funções intrínsecas para a sintaxe curta sempre que possível.

Conceitos básicos de modelos

Para começar a criar um modelo do CloudFormation, siga estas etapas:

1. Escolha recursos: identifique os recursos da AWS que deseja incluir em sua pilha, como instâncias do EC2, VPCs, grupos de segurança e muito mais.
2. Escreva o modelo: escreva o modelo no formato JSON ou YAML, definindo os recursos e suas propriedades.
3. Salvar o modelo: salve o modelo localmente com uma extensão de arquivo como: `.json`, `.yaml` ou `.txt`.
4. Valide o modelo: valide o modelo usando os métodos descritos na seção [Validar modelos](#).
5. Crie uma pilha: crie uma pilha usando o modelo validado.

Planejar o uso da referência do modelo do CloudFormation

Ao criar seus modelos, você poderá encontrar documentação sobre a sintaxe detalhada de diferentes tipos de recursos na [Referência de tipos de propriedades e recursos da AWS](#).

Geralmente, seus modelos de pilha exigirão funções intrínsecas para atribuir valores de propriedades que não estão disponíveis até o momento da execução, bem como atributos especiais para controlar o comportamento dos recursos. Ao escrever seu modelo, consulte os seguintes recursos para obter orientação:

- [Referência de função intrínseca](#): algumas funções intrínsecas comumente usadas incluem:

- **Ref**: recupera o valor de um parâmetro ou o ID físico de um recurso.
- **Sub**: substitui espaços reservados em strings por valores reais.
- **GetAtt**: retorna o valor de um atributo de um recurso no modelo.
- **Join**: une um conjunto de valores em uma única string.
- **Referência de atributo de recurso**: alguns atributos especiais comumente usados incluem:
 - **DependsOn**: use esse atributo para especificar que um recurso deve ser criado depois de outro.
 - **DeletionPolicy**: use esse atributo para especificar como o CloudFormation deve lidar com a exclusão de um recurso.

Exemplos de modelo

O CloudFormation fornece modelos de pilha de código aberto que você pode usar para começar. Para obter mais informações, consulte [CloudFormation Sample Templates](#) no site do GitHub.

Tenha em mente que esses modelos não pretendem estar prontos para produção. Você deve dedicar algum tempo para aprender como eles funcionam, adaptá-los às suas necessidades e garantir que atendam aos padrões de conformidade da sua empresa.

Cada modelo nesse repositório passa pelas verificações do [Linter do CloudFormation](#) (cfn-lint) e também por um conjunto básico de regras do AWS CloudFormation Guard baseadas nos Center for Internet Security (CIS) Top 20, com exceção de algumas regras em que era importante manter a amostra concentrada em um único caso de uso.

Formato de modelo do CloudFormation

Você pode criar modelos do CloudFormation nos formatos JSON ou YAML. Ambos os formatos atendem ao mesmo propósito, mas oferecem vantagens distintas em termos de legibilidade e complexidade.

- **JSON**: o JSON é um formato leve de intercâmbio de dados que é fácil de ser analisado e gerado por computadores. No entanto, ele pode ser complicado para humanos lerem e escreverem, especialmente em configurações complexas. Em JSON, o modelo é estruturado usando chaves {} e colchetes [] aninhados para definir recursos, parâmetros e outros componentes. Sua sintaxe requer uma declaração explícita de cada elemento, o que pode tornar o modelo extremamente detalhado, mas garante a adesão estrita a um formato estruturado.

- **YAML:** o YAML foi projetado para ser mais legível por humanos e menos detalhado do que o JSON. Ele usa recuo em vez de chaves e colchetes para denotar o aninhamento, o que pode facilitar a visualização da hierarquia de recursos e parâmetros. O YAML geralmente é preferido por sua clareza e facilidade de uso, especialmente ao lidar com modelos mais complexos. No entanto, a dependência do YAML do uso de recuos pode levar a erros se o espaçamento não for consistente, o que requer atenção cuidadosa para manter a precisão.

Estrutura do modelo

Os modelos do CloudFormation são divididos em seções diferentes, e cada seção se destina a conter um tipo específico de informação. Algumas seções devem ser declaradas em uma ordem específica e, para outras, a ordem não importa. No entanto, à medida que você cria o modelo, pode ser útil usar a ordem lógica dos exemplos a seguir, porque valores em uma seção podem fazer referência a valores de uma seção anterior.

Ao criar modelos, não use seções importantes duplicadas, por exemplo, a seção `Resources`. Embora o CloudFormation possa aceitar o modelo, ele terá um comportamento indefinido ao processá-lo e poderá provisionar recursos incorretamente ou retornar erros inexplicáveis.

JSON

O exemplo a seguir mostra a estrutura de um modelo no formato JSON com todas as seções disponíveis.

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Rules" : {
    set of rules
  },
```

```
"Mappings" : {  
    set of mappings  
},  
  
"Conditions" : {  
    set of conditions  
},  
  
"Transform" : {  
    set of transforms  
},  
  
"Resources" : {  
    set of resources  
},  
  
"Outputs" : {  
    set of outputs  
}  
}
```

YAML

O exemplo a seguir mostra a estrutura de um modelo no formato YAML com todas as seções disponíveis.

```
---  
AWSTemplateFormatVersion: version date  
  
Description:  
    String  
  
Metadata:  
    template metadata  
  
Parameters:  
    set of parameters  
  
Rules:  
    set of rules  
  
Mappings:
```

set of mappings

Conditions:

set of conditions

Transform:

set of transforms

Resources:

set of resources

Outputs:

set of outputs

Comentários

Em modelos formatados em JSON, não há suporte a comentários. O JSON, por design, não inclui uma sintaxe para comentários, o que significa que não é possível adicionar comentários diretamente na estrutura JSON. No entanto, se você precisar incluir notas explicativas ou documentação, considere adicionar metadados. Para obter mais informações, consulte o atributo [Metadada](#).

Nos modelos formatados em YAML, é possível incluir comentários em linha usando o símbolo #.

O exemplo a seguir mostra um modelo YAML com comentários em linha.

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample CloudFormation template with YAML comments.
# Resources section
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      # Linux AMI
      ImageId: ami-1234567890abcdef0
      InstanceType: t2.micro
      KeyName: MyKey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
      Ebs:
        VolumeType: io1
        Iops: 200
        DeleteOnTermination: false
```

VolumeSize: 20

Especificações

O CloudFormation é compatível com as seguintes especificações de JSON e YAML:

JSON

O CloudFormation segue o padrão JSON ECMA-404. Para obter mais informações sobre o formato JSON, consulte <http://www.json.org>.

YAML

O CloudFormation é compatível com a especificação do YAML versão 1.1 com algumas exceções. O CloudFormation não é compatível com os seguintes recursos:

- As tags `binary`, `omap`, `pairs`, `set` e `timestamp`
- Aliases
- Mesclagens de hash

Para obter mais informações sobre o YAML, consulte <https://yaml.org/>.

Saiba mais

Para cada recurso especificado no modelo, você define as propriedades e os valores usando as regras de sintaxe específicas do JSON ou do YAML. Para obter mais informações sobre a sintaxe de modelo de cada formato, consulte [Seções de modelos do CloudFormation](#).

Usar expressões regulares em modelos do CloudFormation

É possível usar expressões regulares (conhecidas como regexes) em vários lugares dentro de um modelo do CloudFormation, como para a propriedade `AllowedPattern` ao criar o [parâmetro](#) de um modelo.

Todas as expressões regulares no CloudFormation estão em conformidade com a sintaxe regex de Java. Consulte uma descrição abrangente da sintaxe regex do Java e de seus constructos em [java.util.regex.Pattern](#).

Se escrever o modelo do CloudFormation na sintaxe JSON, você deverá escapar qualquer caractere de barra invertida (`\`) na expressão regular adicionando uma barra invertida adicional. Isso porque

o JSON interpreta barras invertidas como caracteres de escape, e você precisa escapá-las para garantir que elas sejam tratadas como barras invertidas literais na expressão regular.

Por exemplo, se você incluir `\d` em sua expressão regular para corresponder a um caractere de dígito, você precisará escrevê-lo como `\\d` em seu modelo JSON.

No exemplo a seguir, a propriedade `AllowedPattern` especifica uma expressão regular que corresponde a quatro caracteres de dígitos consecutivos (`\d{4}`). No entanto, como a expressão regular é definida em um modelo JSON, o caractere de barra invertida precisa ser escapado com uma barra invertida adicional (`\\d`).

```
{
  "Parameters": {
    "MyParameter": {
      "Type": "String",
      "AllowedPattern": "\\d{4}"
    }
  }
}
```

Se escrever o modelo do CloudFormation na sintaxe YAML, você deverá colocar a expressão regular entre aspas simples (`'`). Nenhum escape adicional é necessário.

```
Parameters:
  MyParameter:
    Type: String
    AllowedPattern: '\d{4}'
```

Note

Expressões regulares no CloudFormation só são aceitas para fins de validação em contextos específicos, como `AllowedPattern`. Elas não são aceitas como operações de correspondência de padrões nas funções intrínsecas do CloudFormation, como `Fn::Equals`, que realizam somente a comparação exata de strings, e não a correspondência de padrões.

Seções de modelos do CloudFormation

Cada modelo do CloudFormation consiste em uma ou mais seções, cada uma servindo a um propósito específico.

A seção **Recursos** é obrigatória em todos os modelos do CloudFormation e forma o núcleo do modelo. Ela especifica os recursos da pilha e suas propriedades, como uma instância do Amazon EC2 ou um bucket do Amazon S3. Cada recurso é definido com um ID lógico exclusivo, tipo e detalhes de configuração específicos.

A seção **Parâmetros**, embora opcional, desempenha um papel importante para tornar os modelos mais flexíveis. Ele permite que os usuários passem valores em tempo de execução ao criarem ou atualizarem uma pilha. Esses parâmetros podem ser referenciados nas seções **Resources** e **Outputs**, permitindo a personalização sem alterar o modelo em si. Por exemplo, você pode usar parâmetros para especificar tipos de instância ou configurações de ambiente que variam entre as implantações.

A seção **Saídas**, também opcional, define os valores que são retornados quando as propriedades de uma pilha são visualizadas. As saídas fornecem informações úteis, como identificadores de recursos ou URLs, as quais podem ser aproveitadas para fins operacionais ou para integração com outras pilhas. Esta seção ajuda os usuários a recuperar e usar detalhes importantes sobre os recursos criados pelo modelo.

Outras seções opcionais incluem **Mapeamentos**, que funcionam como tabelas de pesquisa para gerenciar valores condicionais. Com os mapeamentos, você define pares de chave valor e os usa com a função intrínseca `Fn::FindInMap` nas seções **Resources** e **Outputs**. Isso é útil para cenários em que você precisa ajustar as configurações com base em condições como Região da AWS ou ambiente.

As seções **Metadados** e **Regras**, embora menos usadas, fornecem funcionalidades adicionais. **Metadata** pode incluir informações adicionais sobre o modelo, ao mesmo tempo que **Rules** valida um parâmetro ou uma combinação de parâmetros durante a criação ou as atualizações da pilha, garantindo que atendam a critérios específicos. A seção **Condições** aumenta ainda mais a flexibilidade ao controlar se determinados recursos são criados ou propriedades são atribuídas a um valor com base em condições como o tipo de ambiente.

Por fim, a seção **Transformar** é usada para aplicar macros durante o processamento do modelo. Para aplicações sem servidor (também chamadas de aplicações do Lambda), ele especifica a versão

do [AWS Serverless Application Model \(AWS SAM\)](#) que deve ser usada. Quando você especifica uma transformação, você pode usar a sintaxe AWS SAM para declarar recursos em seu modelo. O modelo define a sintaxe que você pode usar e como ela é processada. Você também pode usar a transformação `AWS::Include` para incluir trechos de modelo que são armazenados separadamente do modelo principal do CloudFormation.

Os tópicos a seguir fornecem informações adicionais e exemplos para usar cada seção.

Tópicos

- [Sintaxe de Resources de modelo do CloudFormation](#)
- [Sintaxe de Parameters de modelo do CloudFormation](#)
- [Sintaxe de Outputs de modelo do CloudFormation](#)
- [Sintaxe de Mappings de modelo do CloudFormation](#)
- [Sintaxe de Metadata de modelo do CloudFormation](#)
- [Sintaxe de Rules de modelo do CloudFormation](#)
- [Sintaxe de Conditions de modelo do CloudFormation](#)
- [Seção Transform do modelo do CloudFormation](#)
- [Sintaxe de versão de formato de modelo do CloudFormation](#)
- [Sintaxe de Description de modelo do CloudFormation](#)

Sintaxe de Resources de modelo do CloudFormation

A seção `Resources` é uma seção obrigatória de nível superior em um modelo do CloudFormation. Ela declara os recursos da AWS que você deseja que o CloudFormation provisione e configure como parte da sua pilha.

Sintaxe

A seção `Resources` usa a seguinte sintaxe:

JSON

```
"Resources" : {  
  "LogicalResourceName1" : {  
    "Type" : "AWS::ServiceName::ResourceType",
```



```

    "Properties" : {
        "PropertyName1" : "PropertyValue1",
        ...
    }
},

"LogicalResourceName2" : {
    "Type" : "AWS::ServiceName::ResourceType",
    "Properties" : {
        "PropertyName1" : "PropertyValue1",
        ...
    }
}
}

```

YAML

```

Resources:
  LogicalResourceName1:
    Type: AWS::ServiceName::ResourceType
    Properties:
      PropertyName1: PropertyValue1
      ...

  LogicalResourceName2:
    Type: AWS::ServiceName::ResourceType
    Properties:
      PropertyName1: PropertyValue1
      ...

```

ID lógico (também chamado de nome lógico)

Em um modelo do CloudFormation, os recursos são identificados por seus nomes de recursos lógicos. Esses nomes devem ser alfanuméricos (A-Z a-z 0-9) e exclusivos no modelo. Os nomes lógicos são usados para referenciar recursos de outras seções do modelo.

Tipo de recurso

Um recurso deve ter um atributo `Type`, que define que tipo de recurso da AWS ele é. O atributo `Type` tem o formato `AWS::ServiceName::ResourceType`. Por exemplo, o atributo `Type` para um bucket do Amazon S3 é `AWS::S3::Bucket`.

Para obter uma lista completa dos tipos de recursos compatíveis, consulte a [Referência de tipos de propriedades e recursos da AWS](#).

Propriedades de recursos

Propriedades de recursos são opções adicionais que você pode especificar para definir detalhes de configuração para o tipo de recurso específico. Algumas propriedades são obrigatórias, enquanto outras são opcionais. Algumas propriedades têm valores padrão, portanto, especificá-las é opcional.

Para obter detalhes sobre as propriedades com suporte para cada tipo de recurso, consulte os tópicos na [Referência de tipos de propriedades e recursos da AWS](#).

Os valores das propriedades podem ser strings literais, listas de strings, booleanos, referências dinâmicas, referências de parâmetros, pseudoreferências ou o valor retornado por uma função. Os exemplos a seguir mostram como declarar diferentes tipos de valor de propriedade:

JSON

```
"Properties" : {  
  "String" : "A string value",  
  "Number" : 123,  
  "LiteralList" : [ "first-value", "second-value" ],  
  "Boolean" : true  
}
```

YAML

```
Properties:  
  String: A string value  
  Number: 123  
  LiteralList:  
    - first-value  
    - second-value  
  Boolean: true
```

ID físico

Além do ID lógico, determinados recursos também têm um ID físico, que é o nome real atribuído a esse recurso, como o ID de uma instância EC2 ou o nome de um bucket do S3. Use os IDs físicos para identificar recursos fora dos modelos do CloudFormation, mas apenas depois que os recursos forem criados. Por exemplo, suponha que você forneça a um recurso de instância do EC2

um ID lógico de MyEC2Instance. Quando o CloudFormation cria a instância, ele gera e atribui automaticamente um ID físico (como i-1234567890abcdef0) à instância. Você pode usar esse ID físico para identificar a instância e visualizar suas propriedades (como o nome DNS) usando o console do Amazon EC2.

Para buckets do Amazon S3 e muitos outros recursos, o CloudFormation gera automaticamente um nome físico exclusivo para o recurso, caso você não especifique um de maneira explícita. Esse nome físico é baseado em uma combinação do nome da pilha do CloudFormation, do nome lógico do recurso especificado no modelo do CloudFormation e de um ID exclusivo. Por exemplo, se você tiver um bucket do Amazon S3 com o nome lógico MyBucket em uma pilha chamada MyStack, o CloudFormation poderá nomear o bucket com o nome físico MyStack-MyBucket-abcdefghijkl1.

Para recursos que oferecem suporte a nomes personalizados, é possível atribuir seus próprios nomes físicos para ajudar a identificar os recursos rapidamente. Por exemplo, você pode nomear um bucket do S3 que armazena logs como MyPerformanceLogs. Para obter mais informações, consulte [Tipo de nome](#).

Fazer referência a recursos

Frequentemente, é necessário definir as propriedades em um recurso com base no nome ou na propriedade de outro recurso. Por exemplo, é possível criar uma instância do EC2 que usa grupos de segurança do EC2 ou uma distribuição do CloudFront baseada em um bucket do S3. Todos esses recursos podem ser criados no mesmo modelo do CloudFormation.

O CloudFormation oferece funções intrínsecas que você pode usar para fazer referência a outros recursos e suas propriedades. Essas funções permitem criar dependências entre recursos e passar valores de um recurso para outro.

A função **Ref**

A função Ref é comumente usada para recuperar uma propriedade de identificação de recursos definidos no mesmo modelo do CloudFormation. O que ela retorna depende do tipo de recurso. Para a maioria de recursos, ela retorna o nome físico do recurso. No entanto, para alguns tipos de recurso, ela pode retornar um valor diferente, como um endereço IP para um recurso AWS::EC2::EIP ou um nome do recurso da amazon (ARN) para um tópico do Amazon SNS.

Os exemplos a seguir demonstram como usar a função Ref em propriedades. Em cada um desses exemplos, a função Ref retornará o nome real do recurso LogicalResourceName declarado em outro lugar no modelo. O exemplo de sintaxe de !Ref no exemplo do YAML é apenas uma forma mais curta de escrever a função Ref.

JSON

```
"Properties" : {  
  "PropertyName" : { "Ref" : "LogicalResourceName" }  
}
```

YAML

```
Properties:  
  PropertyName1:  
    Ref: LogicalResourceName  
  PropertyName2: !Ref LogicalResourceName
```

Para obter informações mais detalhadas sobre como usar a função Ref, consulte [Ref](#).

A função **Fn::GetAtt**

A função Ref será útil se o parâmetro ou o valor retornado para um recurso for exatamente o que você deseja. No entanto, outras propriedades de um recurso podem ser necessárias. Por exemplo, se você deseja criar uma distribuição do CloudFront com uma origem do S3, especifique o local do bucket usando um endereço no estilo de DNS. Diversos recursos têm atributos adicionais cujos valores você pode usar em seu modelo. Para obter esses atributos, é necessário usar a função **Fn::GetAtt**.

Os exemplos a seguir demonstram como usar a função GetAtt em propriedades. A função **Fn::GetAtt** utiliza dois parâmetros, o nome lógico do recurso e o nome do atributo para ser recuperado. O exemplo de sintaxe de **!GetAtt** no exemplo do YAML é apenas uma forma mais curta de escrever a função GetAtt.

JSON

```
"Properties" : {  
  "PropertyName" : {  
    "Fn::GetAtt" : [ "LogicalResourceName", "AttributeName" ]  
  }  
}
```

YAML

```
Properties:
```

```
PropertyName1:
  Fn::GetAtt:
    - LogicalResourceName
    - AttributeName
PropertyName2: !GetAtt LogicalResourceName.AttributeName
```

Para obter informações mais detalhadas sobre como usar a função `GetAtt`, consulte [Fn::GetAtt](#).

Exemplos

Os exemplos a seguir ilustram como declarar recursos e como os modelos do CloudFormation podem referenciar outros recursos definidos no mesmo modelo e nos recursos da AWS existentes.

Tópicos

- [Declarar um único recurso com um nome personalizado](#)
- [Fazer referência a outros recursos com a função `Ref`](#)
- [Fazer referência a atributos de recursos usando a função `Fn::GetAtt`](#)

Declarar um único recurso com um nome personalizado

O exemplo a seguir declara um único recurso do tipo `AWS::S3::Bucket` com o nome lógico `MyBucket`. A propriedade `BucketName` está definida como `amzn-s3-demo-bucket`, que deve ser substituída pelo nome desejado para seu bucket do S3.

Se você usar essa declaração de recurso para criar uma pilha, o CloudFormation criará um bucket do Amazon S3 com configurações padrão. Para outros recursos, como uma instância do Amazon EC2 ou grupo do Auto Scaling, o CloudFormation requer mais informações.

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "BucketName": "amzn-s3-demo-bucket"
      }
    }
  }
}
```

YAML

```
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
```

Fazer referência a outros recursos com a função **Ref**

Os exemplos a seguir mostram uma declaração de recurso que define uma instância do EC2 e um grupo de segurança. O recurso `Ec2Instance` faz referência ao recurso `InstanceSecurityGroup` como parte de sua propriedade `SecurityGroupIds` usando a função `Ref`. Ele também inclui um grupo de segurança (`sg-12a4c434`) existente que não está declarado no modelo. Você pode usar strings literais para fazer referência a recursos da AWS existentes.

JSON

```
{
  "Resources": {
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "SecurityGroupIds": [
          {
            "Ref": "InstanceSecurityGroup"
          },
          "sg-12a4c434"
        ],
        "KeyName": "MyKey",
        "ImageId": "ami-1234567890abcdef0"
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
```

```

        "CidrIp": "0.0.0.0/0"
      }
    ]
  }
}

```

YAML

```

Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
        - sg-12a4c434
      KeyName: MyKey
      ImageId: ami-1234567890abcdef0
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0

```

Fazer referência a atributos de recursos usando a função **Fn::GetAtt**

Os exemplos a seguir mostram uma declaração de recursos que define um recurso de distribuição do CloudFront e um bucket do S3. O recurso `MyDistribution` especifica o nome DNS do recurso `MyBucket` usando a função `Fn::GetAtt` para obter o atributo `DomainName` do bucket. Você verá que a função `Fn::GetAtt` lista os dois parâmetros em uma matriz. Para as funções que usam vários parâmetros, você deve usar uma matriz para especificá-los.

JSON

```

{
  "Resources": {
    "MyBucket": {

```

```

    "Type": "AWS::S3::Bucket"
  },
  "MyDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Origins": [
          {
            "DomainName": {
              "Fn::GetAtt": [
                "MyBucket",
                "DomainName"
              ]
            },
            "Id": "MyS3Origin",
            "S3OriginConfig": {}
          }
        ],
        "Enabled": "true",
        "DefaultCacheBehavior": {
          "TargetOriginId": "MyS3Origin",
          "ForwardedValues": {
            "QueryString": "false"
          },
          "ViewerProtocolPolicy": "allow-all"
        }
      }
    }
  }
}

```

YAML

```

Resources:
  MyBucket:
    Type: AWS::S3::Bucket
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: !GetAtt

```



```
- MyBucket
- DomainName
Id: MyS3Origin
S3OriginConfig: {}
Enabled: 'true'
DefaultCacheBehavior:
  TargetOriginId: MyS3Origin
  ForwardedValues:
    QueryString: 'false'
ViewerProtocolPolicy: allow-all
```

Sintaxe de Parameters de modelo do CloudFormation

Use seção opcional `Parameters` para personalizar os modelos. Com os parâmetros, é possível inserir valores personalizados no modelo toda vez que criar ou atualizar uma pilha. Ao usar parâmetros nos modelos, é possível criar modelos reutilizáveis e flexíveis que podem ser personalizados para cenários específicos.

Ao definir parâmetros do tipo apropriado, você pode escolher entre uma lista de identificadores de recursos existentes ao usar o console para criar sua pilha. Para obter mais informações, consulte [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#).

Os parâmetros são uma forma popular de especificar valores de propriedades de recursos de pilha. No entanto, pode haver configurações dependentes de região ou um pouco complexas para os usuários entenderem devido a outras condições ou dependências. Nesses casos, talvez você queira colocar alguma lógica no modelo em si de modo que os usuários possam especificar valores mais simples (ou não) para obter os resultados desejados, por exemplo, usando um mapeamento. Para obter mais informações, consulte [Sintaxe de Mappings de modelo do CloudFormation](#).

Sintaxe

Você declara parâmetros em uma seção `Parameters` do modelo, que usa a seguinte sintaxe geral:

JSON

```
"Parameters" : {
  "ParameterLogicalID" : {
    "Description": "Information about the parameter",
    "Type" : "DataType",
    "Default" : "value",
```

```
"AllowedValues" : ["value1", "value2"]
}
```

YAML

```
Parameters:
  ParameterLogicalID:
    Description: Information about the parameter
    Type: DataType
    Default: value
    AllowedValues:
      - value1
      - value2
```

Um parâmetro contém uma lista de atributos que definir o valor e as restrições em relação ao seu valor. O único atributo obrigatório é `Type`, que pode ser `String`, `Number` ou um tipo de parâmetro fornecido pelo CloudFormation. Também é possível adicionar um atributo `Description` que descreve que tipo de valor deve ser especificado. O nome e a descrição do parâmetro são exibidos na página Especificar parâmetros quando um usuário usa o modelo no assistente Criar pilha.

Note

Por padrão, o console do CloudFormation lista os parâmetros de entrada em ordem alfabética por ID lógico. Para substituir essa ordem padrão e agrupar os parâmetros relacionados, use a chave de metadados `AWS::CloudFormation::Interface` no modelo. Para obter mais informações, consulte [Organizar parâmetros do CloudFormation com metadados AWS::CloudFormation::Interface](#).

Para parâmetros com valores padrão, o CloudFormation usa os valores padrão, a menos que os usuários especifiquem outro valor. Se você omitir o atributo padrão, os usuários deverão especificar um valor para esse parâmetro. No entanto, exigir que o usuário insira um valor não garante que o valor seja válido. Para validar o valor de um parâmetro, você pode declarar restrições ou especificar um tipo de parâmetro específico da AWS.

Para parâmetros sem valores padrão, os usuários devem especificar o valor do nome de uma chave na criação da pilha. Se isso não for feito, o CloudFormation falhará ao criar a pilha e lançará uma exceção:

Parameters: [KeyName] must have values

Propriedades

AllowedPattern

Uma expressão regular que representa os padrões a serem permitidos para tipos `String` ou `CommaDelimitedList`. Quando aplicado a um parâmetro do tipo `String`, o padrão deve corresponder ao valor de parâmetro inteiro fornecido. Quando aplicado a um parâmetro do tipo `CommaDelimitedList`, o padrão deve corresponder a cada valor da lista.

Obrigatório: não

AllowedValues

Uma matriz que contém a lista de valores permitidos para o parâmetro. Quando aplicado a um parâmetro do tipo `String`, o valor do parâmetro deve ser um dos valores permitidos. Quando aplicado a um parâmetro do tipo `CommaDelimitedList`, cada valor da lista deve ser um dos valores permitidos especificados.

Obrigatório: não

Note

Se você estiver usando YAML e quiser usar as strings `Yes` e `No` para `AllowedValues`, use aspas simples para evitar que o analisador YAML considere esses valores como booleanos.

ConstraintDescription

Uma sequência que explica uma restrição quando a restrição é violada. Por exemplo, sem uma descrição da restrição, um parâmetro que tem um padrão permitido de `[A-Za-z0-9]+` exibe a seguinte mensagem de erro quando o usuário especifica um valor inválido:

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

Ao adicionar uma descrição de restrição, como deve conter apenas letras (maiúsculas e minúsculas) e números, você pode exibir a seguinte mensagem de erro personalizada:

Malformed input-Parameter MyParameter must only contain uppercase and lowercase letters and numbers

Obrigatório: não

Default

Um valor do tipo apropriado para o modelo a ser usado se nenhum valor for especificado quando uma pilha é criada. Se definir restrições para o parâmetro, você deverá especificar um valor que esteja de acordo com essas restrições.

Obrigatório: não

Description

Uma sequência de até 4000 caracteres que descreve o parâmetro.

Obrigatório: não

MaxLength

Um valor inteiro que determina o maior número de caracteres que você deseja permitir para tipos `String`.

Obrigatório: não

MaxValue

Um valor numérico que determina o maior valor numérico que você deseja permitir para tipos `Number`.

Obrigatório: não

MinLength

Um valor inteiro que determina o menor número de caracteres que você deseja permitir para tipos `String`.

Obrigatório: não

MinValue

Um valor numérico que determina o menor valor numérico que você deseja permitir para tipos `Number`.

Obrigatório: não

NoEcho

Se o valor do parâmetro deve ser mascarado para impedir que ele seja exibido no console, nas ferramentas de linha de comando ou na API. Se você definir o atributo NoEcho como `true`, o CloudFormation retornará o valor do parâmetro mascarado como asteriscos (*****) para qualquer chamada que descreva a pilha ou os eventos de pilha, exceto informações armazenadas nos locais especificados abaixo.

Obrigatório: não

Important

O uso do atributo NoEcho não mascara informações armazenadas no seguinte:

- A seção de modelo de Metadata. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção Metadata. Para obter mais informações, consulte [Metadados](#).
- A seção de modelo de Outputs. Para obter mais informações, consulte [Saídas](#).
- O atributo Metadata de uma definição de recurso. Para obter mais informações, consulte o atributo [Metadata](#).

É altamente recomendável não usar esses mecanismos para incluir informações confidenciais, como senhas ou segredos.

Important

Em vez de incorporar informações confidenciais diretamente em modelos do CloudFormation, recomendamos usar os parâmetros dinâmicos no modelo da pilha para fazer referência a informações confidenciais que são armazenadas e gerenciadas de forma externa ao CloudFormation, como no AWS Systems Manager Parameter Store ou no AWS Secrets Manager.

Para obter mais informações, consulte a prática recomendada [Não incorporar credenciais nos seus modelos](#).

Important

É altamente recomendável evitar parâmetros NoEcho, ou dados confidenciais, nas propriedades do recurso que fazem parte do identificador primário de um recurso.

Quando um parâmetro NoEcho está incluído em uma propriedade que forma um identificador de recurso primário, o CloudFormation pode utilizar o valor de texto sem formatação real no identificador do recurso primário. Esse ID de recurso pode aparecer em quaisquer saídas ou destinos derivados.

Para determinar quais propriedades de recursos compreendem o identificador primário de um tipo de recurso, consulte a documentação de referência de recursos desse recurso na [Referência de tipos de propriedades e recursos da AWS](#). Na seção Return values (Valores de retorno), o valor de retorno da função Ref representa as propriedades do recurso que compõem o identificador primário do tipo de recurso.

Type

O tipo de dados para o parâmetro (DataType).

Obrigatório: Sim

O CloudFormation oferece suporte aos seguintes tipos de parâmetros:

String

Uma sequência literal. É possível usar os seguintes atributos para declarar restrições: MinLength, MaxLength, Default, AllowedValues e AllowedPattern.

Por exemplo, os usuários podem especificar "MyUserName".

Number

Um inteiro ou flutuante. O CloudFormation valida o valor do parâmetro como um número. No entanto, quando você usa o parâmetro em outro lugar no modelo (por exemplo, usando a função intrínseca Ref), o valor do parâmetro se torna uma string.

É possível usar os seguintes atributos para declarar restrições: MinValue, MaxValue, Default e AllowedValues.

Por exemplo, os usuários podem especificar "8888".

List<Number>

Uma matriz de inteiros ou floats que são separados por vírgulas. O CloudFormation valida o valor do parâmetro como números. No entanto, quando você usa o parâmetro em outro lugar no modelo (por exemplo, usando a função intrínseca Ref), o valor do parâmetro se torna uma lista de strings.

Por exemplo, os usuários poderiam especificar "80, 20", e uma Ref resultaria em ["80", "20"].

CommaDelimitedList

Uma matriz de sequências literais que são separadas por vírgulas. O total de sequências deve ser um número a mais que o número total de vírgulas. Além disso, cada sequência membro é truncado por espaço.

Por exemplo, os usuários poderiam especificar "test, dev, prod", e uma Ref resultaria em ["test", "dev", "prod"].

AWS Tipos de parâmetros específicos da

Os valores da AWS, como nomes de pares de chaves do Amazon EC2 e IDs de VPC. Para obter mais informações, consulte [Especificar recursos existentes no runtime](#).

Tipos de parâmetros do Systems Manager

Parâmetros que correspondem a parâmetros existentes no Armazenamento de parâmetros do Systems Manager. Você especifica uma chave de parâmetro do Systems Manager como o valor do tipo de parâmetro do Systems Manager e o CloudFormation recupera o valor mais recente no Parameter Store para usá-lo na pilha. Para obter mais informações, consulte [Especificar recursos existentes no runtime](#).

Requisitos gerais de parâmetros

Os seguintes requisitos se aplicam ao usar parâmetros:

- É possível ter no máximo 200 parâmetros em um modelo do CloudFormation.
- Cada parâmetro deve receber um nome lógico (também chamado de ID lógico), que deve ser alfanumérico e exclusivo entre todos os nomes lógicos no modelo.
- A cada parâmetro deve ser atribuído um tipo de parâmetro compatível com o CloudFormation. Para obter mais informações, consulte [Tipo](#).
- A cada parâmetro deve ser atribuído um valor em runtime para que o CloudFormation provisione a pilha com êxito. Se preferir, especifique um valor padrão para o CloudFormation usar, a menos que outro valor seja fornecido.
- Os parâmetros devem ser declarados e referenciados no mesmo modelo. Você pode fazer referência a parâmetros nas seções Resources e Outputs do modelo.

Exemplos

Tópicos

- [Parâmetro String simples](#)
- [Parâmetro de senha](#)
- [Referenciar parâmetros](#)
- [Parâmetro de lista delimitada por vírgulas](#)
- [Retornar um valor de um parâmetro da lista delimitada por vírgula](#)

Parâmetro String simples

O exemplo a seguir declara um parâmetro chamado `InstanceTypeParameter` do tipo `String`. Esse parâmetro permite que você especifique o tipo de instância do Amazon EC2 para a pilha. Se nenhum valor for fornecido durante a criação ou atualização da pilha, o CloudFormation usará o valor padrão de `t2.micro`.

JSON

```
"Parameters" : {
  "InstanceTypeParameter" : {
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro.",
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"]
  }
}
```

YAML

```
Parameters:
  InstanceTypeParameter:
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
```


Parâmetro de senha

O exemplo a seguir declara um parâmetro chamado DBPwd do tipo `String` sem valor padrão. A propriedade `NoEcho` é definida como `true` para evitar que o valor do parâmetro seja exibido nas descrições da pilha. O comprimento mínimo que pode ser especificado é 1, e o valor máximo é 41. O padrão permite caracteres alfabéticos maiúsculos e minúsculos e numerais. Esse exemplo também ilustra o uso de uma expressão regular para a propriedade `AllowedPattern`.

JSON

```
"Parameters" : {
  "DBPwd" : {
    "NoEcho" : "true",
    "Description" : "The database admin account password",
    "Type" : "String",
    "MinLength" : "1",
    "MaxLength" : "41",
    "AllowedPattern" : "^[a-zA-Z0-9]*$"
  }
}
```

YAML

```
Parameters:
  DBPwd:
    NoEcho: true
    Description: The database admin account password
    Type: String
    MinLength: 1
    MaxLength: 41
    AllowedPattern: ^[a-zA-Z0-9]*$
```

Referenciar parâmetros

Você usa a função intrínseca `Ref` para fazer referência a um parâmetro, e o CloudFormation usa o valor do parâmetro para provisionar a pilha. Você pode fazer referência a parâmetros nas seções `Resources` e `Outputs` do mesmo modelo.

No exemplo a seguir, a propriedade `InstanceType` do recurso da instância do EC2 faz referência ao valor do parâmetro `InstanceTypeParameter`:

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "InstanceType" : { "Ref" : "InstanceTypeParameter" },
    "ImageId" : "ami-0ff8a91507f77f867"
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType:
      Ref: InstanceTypeParameter
    ImageId: ami-0ff8a91507f77f867
```

Parâmetro de lista delimitada por vírgulas

O tipo de parâmetro `CommaDelimitedList` pode ser útil quando você precisa fornecer vários valores para uma única propriedade. O exemplo a seguir declara um parâmetro chamado `DbSubnetIpBlocks` com um valor padrão de três blocos CIDR separados por vírgulas.

JSON

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three CIDR blocks",
    "Type": "CommaDelimitedList",
    "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

YAML

```
Parameters:
  DbSubnetIpBlocks:
    Description: "Comma-delimited list of three CIDR blocks"
    Type: CommaDelimitedList
```

```
Default: "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
```

Retornar um valor de um parâmetro da lista delimitada por vírgula

Para fazer referência a um valor específico em uma lista delimitada por vírgulas de um parâmetro, use a função intrínseca `Fn::Select` na seção `Resources` do modelo. Passe o valor do índice do objeto que você deseja e uma lista de objetos, conforme mostrado no exemplo a seguir.

JSON

```
{
  "Parameters": {
    "VPC": {
      "Type": "String",
      "Default": "vpc-123456"
    },
    "VpcAzs": {
      "Type": "CommaDelimitedList",
      "Default": "us-west-2a, us-west-2b, us-west-2c"
    },
    "DbSubnetIpBlocks": {
      "Type": "CommaDelimitedList",
      "Default": "172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26"
    }
  },
  "Resources": {
    "DbSubnet1": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "AvailabilityZone": {
          "Fn::Select": [
            0,
            {
              "Ref": "VpcAzs"
            }
          ]
        },
        "VpcId": {
          "Ref": "VPC"
        },
        "CidrBlock": {
          "Fn::Select": [
            0,
```

```

        { "Ref": "DbSubnetIpBlocks" }
      ]
    }
  },
  "DbSubnet2": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {
        "Fn::Sub": [
          "${AWS::Region}${AZ}",
          {
            "AZ": {
              "Fn::Select": [
                1,
                { "Ref": "VpcAzs" }
              ]
            }
          }
        ]
      },
      "VpcId": {
        "Ref": "VPC"
      },
      "CidrBlock": {
        "Fn::Select": [
          1,
          { "Ref": "DbSubnetIpBlocks" }
        ]
      }
    }
  },
  "DbSubnet3": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {
        "Fn::Sub": [
          "${AWS::Region}${AZ}",
          {
            "AZ": {
              "Fn::Select": [
                2,
                { "Ref": "VpcAzs" }
              ]
            }
          }
        ]
      }
    }
  }
}

```

```
    }  
  }  
]  
,  
"VpcId": {  
  "Ref": "VPC"  
},  
"CidrBlock": {  
  "Fn::Select": [  
    2,  
    { "Ref": "DbSubnetIpBlocks" }  
  ]  
}  
}  
}  
}
```

YAML

```
Parameters:
  VPC:
    Type: String
    Default: vpc-123456
  VpcAzs:
    Type: CommaDelimitedList
    Default: us-west-2a, us-west-2b, us-west-2c
  DbSubnetIpBlocks:
    Type: CommaDelimitedList
    Default: 172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26

Resources:
  DbSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Select
        - 0
        - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 0
        - !Ref DbSubnetIpBlocks
  DbSubnet2:
    Type: AWS::EC2::Subnet
```

```
Properties:
  AvailabilityZone: !Sub
    - ${AWS::Region}${AZ}
    - AZ: !Select
      - 1
      - !Ref VpcAzs
  VpcId: !Ref VPC
  CidrBlock: !Select
    - 1
    - !Ref DbSubnetIpBlocks
DbSubnet3:
  Type: AWS::EC2::Subnet
  Properties:
    AvailabilityZone: !Sub
      - ${AWS::Region}${AZ}
      - AZ: !Select
        - 2
        - !Ref VpcAzs
    VpcId: !Ref VPC
    CidrBlock: !Select
      - 2
      - !Ref DbSubnetIpBlocks
```

Recursos relacionados

O CloudFormation também permite o uso de referências dinâmicas para especificar valores de propriedades dinamicamente. Por exemplo, talvez você precise fazer referência a strings seguras armazenadas no Systems Manager Parameter Store. Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

Você também pode usar pseudoparâmetros em uma Ref ou em uma função Sub para preencher valores dinamicamente. Para obter mais informações, consulte [Obter valores da AWS usando pseudoparâmetros](#).

Sintaxe de Outputs de modelo do CloudFormation

A seção Outputs opcional declara os valores de saída para a pilha. Esses valores de saída podem ser usados de várias maneiras:

- Capture detalhes importantes sobre os recursos: uma saída é uma forma conveniente de coletar informações importantes sobre os recursos. Por exemplo, você pode produzir o nome do bucket

do S3 para uma pilha para tornar o bucket mais fácil de encontrar. Os valores de saída podem ser visualizados na guia Saídas do console do CloudFormation ou via comando [describe-stacks](#) da CLI.

- Referências entre pilhas: é possível importar valores de saída para outras pilhas para [criar referências entre pilhas](#). Isso é útil quando você precisa compartilhar recursos ou configurações em várias pilhas.

Important

O CloudFormation não edita nem ofusca nenhuma informação incluída na seção Outputs. É altamente recomendável não usar essa seção para gerar informações confidenciais, como senhas ou segredos.

Os valores de saída estão disponíveis após a conclusão da operação da pilha. Os valores de saída da pilha não estão disponíveis quando um status de pilha está em qualquer um dos [status](#) IN_PROGRESS. Não recomendamos estabelecer dependências entre um runtime de serviço e o valor de saída da pilha porque os valores de saída podem não estar disponíveis em todos os momentos.

Sintaxe

A seção Outputs consiste no nome da chave Outputs. Você pode declarar no máximo 200 saídas em um modelo.

O exemplo a seguir demonstra a estrutura da seção Outputs.

JSON

Use chaves para circunscrever todas as declarações de saída. Delimite várias saídas com vírgulas.

```
"Outputs" : {  
  "OutputLogicalID" : {  
    "Description" : "Information about the value",  
    "Value" : "Value to return",  
    "Export" : {  
      "Name" : "Name of resource to export"  
    }  
  }  
}
```

```
}
```

YAML

Outputs:

OutputLogicalID:

Description: *Information about the value*

Value: *Value to return*

Export:

Name: *Name of resource to export*

Campos de saída

A seção `Outputs` pode incluir os seguintes campos.

ID lógico (também chamado de nome lógico)

Um identificador para a saída atual. O ID lógico deve ser alfanumérico (a-z, A-Z, 0-9) e exclusivo no modelo.

Description (opcional)

Um tipo `String` que descreve o valor da saída. O valor da declaração da descrição deve ser uma sequência literal com 0 a 1024 bytes de comprimento. Você não pode usar um parâmetro ou uma função para especificar a descrição.

Value (obrigatório)

O valor da propriedade retornado pelo comando [describe-stacks](#). O valor de uma saída pode incluir literais, referências a parâmetros, pseudoparâmetros, um valor de mapeamento ou funções intrínsecas.

Export (opcional)

O nome da saída do recurso saída a ser exportado para uma referência de pilha cruzada.

Você pode usar funções intrínsecas para personalizar o valor de `Name` de uma exportação.

Para obter mais informações, consulte [Obter resultados exportados de uma pilha do CloudFormation implantada](#).

Para associar uma condição a uma saída, defina a condição na seção [Conditions](#) do modelo.

Exemplos

Os exemplos a seguir ilustram como funciona a saída da pilha.

Tópicos

- [Saída da pilha](#)
- [Personalizar o nome da exportação usando Fn::Sub](#)
- [Personalizar o nome da exportação usando Fn::Join](#)
- [Retornar um URL construído usando Fn::Join](#)

Saída da pilha

No exemplo a seguir, a saída chamada BackupLoadBalancerDNSName retorna o nome DNS do recurso com o ID lógico BackupLoadBalancer apenas quando a condição CreateProdResources é verdadeira. A saída chamada InstanceID retorna o ID da instância do EC2 com o ID lógico EC2Instance.

JSON

```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ] },
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

YAML

```
Outputs:
  BackupLoadBalancerDNSName:
    Description: The DNSName of the backup load balancer
    Value: !GetAtt BackupLoadBalancer.DNSName
    Condition: CreateProdResources
  InstanceID:
    Description: The Instance ID
```

```
Value: !Ref EC2Instance
```

Personalizar o nome da exportação usando **Fn::Sub**

Nos exemplos a seguir, a saída chamada `StackVPC` retorna o ID de uma VPC e, em seguida, exporta o valor para a referência de pilha cruzada com o nome `VPCID` anexado ao nome da pilha.

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : {"Fn::Sub": "${AWS::StackName}-VPCID" }
    }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Sub "${AWS::StackName}-VPCID"
```

Para obter mais informações sobre a função `Fn::Sub`, consulte [Fn::Sub](#).

Personalizar o nome da exportação usando **Fn::Join**

Use a função `Fn::Join` para construir valores com base em parâmetros, atributos de recursos e outras strings.

Os exemplos apresentados a seguir usam a função `Fn::Join`, em vez da função `Fn::Sub`, para personalizar o nome da exportação. A função `Fn::Join` de exemplo concatena o nome da pilha com o nome `VPCID` usando dois pontos como um separador.

JSON

```
"Outputs" : {
```

```
"StackVPC" : {
  "Description" : "The ID of the VPC",
  "Value" : { "Ref" : "MyVPC" },
  "Export" : {
    "Name" : { "Fn::Join" : [ ":", [ { "Ref" : "AWS::StackName" }, "VPCID" ] ] }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Join [ ":", [ !Ref "AWS::StackName", VPCID ] ]
```

Para obter mais informações sobre a função `Fn::Join`, consulte [Fn::Join](#).

Retornar um URL construído usando **Fn::Join**

No exemplo a seguir de um modelo que cria um site do WordPress, `InstallURL` é a string retornada por uma chamada da função `Fn::Join` que concatena `http://`, o nome DNS do recurso `ElasticLoadBalancer` e `/wp-admin/install.php`. O valor de saída seria semelhante ao seguinte:

```
http://mywptests-elasticl-1gb51l6sl8y5v-206169572.aws-region.elb.amazonaws.com/wp-admin/install.php
```

JSON

```
{
  "Outputs": {
    "InstallURL": {
      "Value": {
        "Fn::Join": [
          "",
          [
            "http://",
            {
```

```

        "Fn::GetAtt": [
            "ElasticLoadBalancer",
            "DNSName"
        ]
    },
    "/wp-admin/install.php"
]
},
"Description": "Installation URL of the WordPress website"
}
}
}

```

YAML

```

Outputs:
  InstallURL:
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt
          - ElasticLoadBalancer
          - DNSName
        - /wp-admin/install.php
    Description: Installation URL of the WordPress website

```

Para obter mais informações sobre a função `Fn::Join`, consulte [Fn::Join](#).

Sintaxe de Mappings de modelo do CloudFormation

A seção opcional `Mappings` ajuda a criar pares de chave-valor que podem ser usados para especificar valores com base em determinadas condições ou dependências.

Um caso de uso comum para a seção `Mappings` é definir valores com base na Região da AWS em que a pilha é implantada. Isso pode ser feito por meio do pseudoparámetro `AWS::Region`. O pseudoparámetro `AWS::Region` é um valor que o CloudFormation resolve para a região em que a pilha é criada. Os pseudoparámetros são resolvidos pelo CloudFormation quando você cria a pilha.

Para recuperar valores em um mapa, você pode usar a função intrínseca `Fn::FindInMap` na seção `Resources` do seu modelo.

Sintaxe

A seção Mappings usa a seguinte sintaxe:

JSON

```
"Mappings" : {  
  "MappingLogicalName" : {  
    "Key1" : {  
      "Name" : "Value1"  
    },  
    "Key2" : {  
      "Name" : "Value2"  
    },  
    "Key3" : {  
      "Name" : "Value3"  
    }  
  }  
}
```

YAML

```
Mappings:  
  MappingLogicalName:  
    Key1:  
      Name: Value1  
    Key2:  
      Name: Value2  
    Key3:  
      Name: Value3
```

- MappingLogicalName é o nome lógico do mapeamento.
- Dentro de um mapeamento, cada mapa é uma chave seguida por outro mapeamento.
- A chave deve ser um mapa de pares nome/valor e exclusiva dentro do mapeamento.
- O par de nome/valor é um rótulo e o valor para mapear. Nomeando os valores, você pode mapear mais de um conjunto de valores para uma chave.
- As chaves em mapeamentos devem ser sequências literais.
- Os valores podem ser do tipo String ou List.

Note

Você não pode incluir parâmetros, pseudoparâmetros ou funções intrínsecas na seção **Mappings**.

Se os valores em um mapeamento não estiverem sendo usados atualmente pela pilha, você não poderá atualizar apenas o mapeamento. Você precisa incluir alterações que adicionam, modificam ou excluem recursos.

Exemplos

Tópicos

- [Mapeamento básico](#)
- [Mapeamento com vários valores](#)
- [Retornar um valor de um mapeamento](#)
- [Parâmetro de entrada e Fn::FindInMap](#)

Mapeamento básico

O exemplo a seguir mostra uma seção **Mappings** com um mapa **RegionToInstanceType** que contém cinco chaves que são mapeadas para pares de nome e valor que contêm valores de sequência únicos. As chaves são nomes de regiões. Cada par de nome-valor é um tipo de instância da família T que está disponível na região representada pela chave. Os pares de nome-valor têm um nome (**InstanceType** no exemplo) e um valor.

JSON

```
"Mappings" : {
  "RegionToInstanceType" : {
    "us-east-1"      : { "InstanceType" : "t2.micro" },
    "us-west-1"      : { "InstanceType" : "t2.micro" },
    "eu-west-1"      : { "InstanceType" : "t2.micro" },
    "eu-north-1"     : { "InstanceType" : "t3.micro" },
    "me-south-1"     : { "InstanceType" : "t3.micro" }
  }
}
```

YAML

```
Mappings:
  RegionToInstanceType:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
```

Mapeamento com vários valores

O exemplo a seguir tem chaves de regiões que são mapeadas para dois conjuntos de valores: uma chamada MyAMI1 e a outra MyAMI2.

Note

Os IDs de AMI mostrados nesses exemplos são espaços reservados para fins de demonstração. Sempre que possível, considere usar referências dinâmicas aos parâmetros AWS Systems Manager como alternativa à seção Mappings. Para evitar atualizar todos os modelos com um novo ID sempre que a AMI que você deseja usar for alterada, use um parâmetro AWS Systems Manager para recuperar o ID da AMI mais recente quando a pilha for criada ou atualizada. As versões mais recentes de AMIs comumente usadas também estão disponíveis como parâmetros públicos no Systems Manager. Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

JSON

```
"Mappings" : {
  "RegionToAMI" : {
    "us-east-1"      : { "MyAMI1" : "ami-12345678901234567", "MyAMI2" :
"ami-23456789012345678" },
    "us-west-1"     : { "MyAMI1" : "ami-34567890123456789", "MyAMI2" :
"ami-45678901234567890" },
```

```

    "eu-west-1"      : { "MyAMI1" : "ami-56789012345678901", "MyAMI2" :
"ami-67890123456789012" },
    "ap-southeast-1" : { "MyAMI1" : "ami-78901234567890123", "MyAMI2" :
"ami-89012345678901234" },
    "ap-northeast-1" : { "MyAMI1" : "ami-90123456789012345", "MyAMI2" :
"ami-01234567890123456" }
  }
}

```

YAML

```

Mappings:
  RegionToAMI:
    us-east-1:
      MyAMI1: ami-12345678901234567
      MyAMI2: ami-23456789012345678
    us-west-1:
      MyAMI1: ami-34567890123456789
      MyAMI2: ami-45678901234567890
    eu-west-1:
      MyAMI1: ami-56789012345678901
      MyAMI2: ami-67890123456789012
    ap-southeast-1:
      MyAMI1: ami-78901234567890123
      MyAMI2: ami-89012345678901234
    ap-northeast-1:
      MyAMI1: ami-90123456789012345
      MyAMI2: ami-01234567890123456

```

Retornar um valor de um mapeamento

Você pode usar a função `Fn::FindInMap` para retornar um valor nomeado com base em uma chave especificada. O modelo de exemplo a seguir contém um recurso do Amazon EC2 cuja propriedade `InstanceType` é atribuída pela função `FindInMap`. A função `FindInMap` especifica a chave como Região da AWS onde a pilha é criada (usando o pseudoparámetro `AWS::Region`) e `InstanceType` como o nome do valor para o qual mapear. O `ImageId` usa um parâmetro do Systems Manager para recuperar dinamicamente a AMI mais recente do Amazon Linux 2. Para obter mais informações sobre pseudoparámetros, consulte [Obter valores da AWS usando pseudoparámetros](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionToInstanceType" : {
      "us-east-1"      : { "InstanceType" : "t2.micro" },
      "us-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-north-1"     : { "InstanceType" : "t3.micro" },
      "me-south-1"     : { "InstanceType" : "t3.micro" }
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap" : [ "RegionToInstanceType", { "Ref" :
"AWS::Region" }, "InstanceType" ] }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Mappings:
  RegionToInstanceType:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
Resources:
  myEC2Instance:
```

```
Type: AWS::EC2::Instance
Properties:
  ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
  InstanceType: !FindInMap [RegionToInstanceType, !Ref 'AWS::Region', InstanceType]
```

Parâmetro de entrada e **Fn::FindInMap**

O seguinte exemplo mostra como criar uma instância do EC2 usando vários mapeamentos. O modelo usa mapeamentos aninhados para selecionar automaticamente o tipo de instância e o grupo de segurança apropriados com base no destino Região da AWS e no tipo de ambiente (Dev ou Prod). Ele também usa um parâmetro do Systems Manager para recuperar dinamicamente a AMI mais recente do Amazon Linux 2.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [ "Dev", "Prod" ]
    }
  },
  "Mappings" : {
    "RegionAndEnvironmentToInstanceType" : {
      "us-east-1"      : { "Dev" : "t3.micro", "Prod" : "c5.large" },
      "us-west-1"      : { "Dev" : "t2.micro", "Prod" : "m5.large" }
    },
    "RegionAndEnvironmentToSecurityGroup" : {
      "us-east-1"      : { "Dev" : "sg-12345678", "Prod" : "sg-abcdef01" },
      "us-west-1"      : { "Dev" : "sg-ghijkl23", "Prod" : "sg-45678abc" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
```

```

    "InstanceType" : { "Fn::FindInMap": [ "RegionAndEnvironmentToInstanceType",
{ "Ref": "AWS::Region" }, { "Ref": "EnvironmentType" } ]},
    "SecurityGroupIds" : [{ "Fn::FindInMap" :
[ "RegionAndEnvironmentToSecurityGroup", { "Ref" : "AWS::Region" }, { "Ref" :
"EnvironmentType" } ]}]
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  RegionAndEnvironmentToInstanceType:
    us-east-1:
      Dev: t3.micro
      Prod: c5.large
    us-west-1:
      Dev: t2.micro
      Prod: m5.large
  RegionAndEnvironmentToSecurityGroup:
    us-east-1:
      Dev: sg-12345678
      Prod: sg-abcdef01
    us-west-1:
      Dev: sg-ghijkl23
      Prod: sg-45678abc
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'

```

```
InstanceType: !FindInMap [RegionAndEnvironmentToInstanceType, !Ref
'AWS::Region', !Ref EnvironmentType]
SecurityGroupIds:
  - !FindInMap [RegionAndEnvironmentToSecurityGroup, !Ref 'AWS::Region', !Ref
EnvironmentType]
```

Recursos relacionados

Esses tópicos relacionados podem ser úteis à medida que você desenvolve modelos que usam a função `Fn::FindInMap`.

- [Fn::FindInMap](#)
- [Fn::FindInMap Melhorias do](#)
- [Fn::Sub](#)

Sintaxe de Metadata de modelo do CloudFormation

Metadata armazena informações adicionais usando objetos JSON ou YAML. Os tipos de metadados em nível de modelo que você pode usar em seu modelo incluem:

Metadados personalizados

Armazena pares de chave-valor definidos pelo usuário. Por exemplo, você pode fornecer informações adicionais que não afetam a criação de recursos, mas oferecem contexto adicional sobre as especificidades da infraestrutura, da equipe ou da implantação.

`AWS::CloudFormation::Interface`

Define o agrupamento e a classificação de parâmetros de entrada quando eles são exibidos no console do CloudFormation. Por padrão, o console do CloudFormation classifica alfabeticamente os parâmetros por ID lógico.

`AWS::CloudFormation::Designer`

O CloudFormation Designer (Designer) atingiu o fim da vida útil em 5 de fevereiro de 2025.

⚠ Important

Durante a atualização de uma pilha, você não pode atualizar a seção Metadata por si só. Você pode atualizá-la apenas ao incluir alterações que adicionam, modificam ou excluem recursos.

O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção Metadata. Por isso, é altamente recomendável não usar essa seção para armazenar informações confidenciais, como senhas ou segredos.

Sintaxe

Para declarar metadados personalizados no modelo do CloudFormation, use a seguinte sintaxe:

JSON

```
"Metadata" : {  
  "Instances" : {"Description" : "Information about the instances"},  
  "Databases" : {"Description" : "Information about the databases"}  
}
```

YAML

```
Metadata:  
  Instances:  
    Description: "Information about the instances"  
  Databases:  
    Description: "Information about the databases"
```

Para obter a sintaxe do `AWS::CloudFormation::Interface`, consulte [Organizar parâmetros do CloudFormation com metadados AWS::CloudFormation::Interface](#).

Organizar parâmetros do CloudFormation com metadados

AWS::CloudFormation::Interface

`AWS::CloudFormation::Interface` é uma chave de metadados que define como os parâmetros são agrupados e classificados no console do CloudFormation. Por padrão, o console lista os parâmetros de entrada em ordem alfabética pelos IDs lógicos quando você cria ou atualiza pilhas no console. Usando essa chave, você pode definir seu próprio agrupamento e classificação de

parâmetros de forma que os usuários possam especificar valores de parâmetros de forma eficiente. Por exemplo, você pode agrupar todos os parâmetros relacionados ao EC2 em um grupo e todos os parâmetros relacionados à VPC em outro grupo.

Na chave de metadados, você pode especificar os grupos a serem criados, os parâmetros a serem incluídos em cada grupo e a ordem na qual o console mostra cada parâmetro em seu grupo.

Você também pode definir rótulos para parâmetros. Um rótulo é um nome amigável ou uma descrição que o console exibe em vez de um ID lógico do parâmetro. Os rótulos são úteis para ajudar os usuários a compreenderem os valores a serem especificados para cada parâmetro. Por exemplo, você pode rotular um parâmetro `KeyPair Select` an `EC2 key pair`.

Todos os parâmetros aos quais você faz referência na chave de metadados devem ser declarados na seção `Parameters` do modelo.

Note

Somente o console do CloudFormation usa a chave de metadados `AWS::CloudFormation::Interface`. A AWS CLI e as chamadas da API não usam essa chave.

Sintaxe

Para declarar essa entidade no modelo do CloudFormation, use a seguinte sintaxe:

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups": [
      {
        "Label": {
          "default": "Group Label"
        },
        "Parameters": [
          "Parameter1",
          "Parameter2"
        ]
      }
    ],
    "ParameterLabels": {
```

```
"Parameter1": {  
  "default": "Friendly Name for Parameter1"  
}  
}  
}
```

YAML

```
Metadata:  
  AWS::CloudFormation::Interface:  
    ParameterGroups:  
      - Label:  
          default: Group Label  
        Parameters:  
          - Parameter1  
          - Parameter2  
    ParameterLabels:  
      Parameter1:  
        default: Friendly Name for Parameter1
```

Propriedades

ParameterGroups

Uma lista de tipos de parameter group, em que você especifica nomes de grupo, os parâmetros em cada grupo e a ordem na qual os parâmetros são mostrados.

Obrigatório: não

Label

Um nome do grupo de parâmetros.

Obrigatório: não

default

O rótulo padrão usado pelo console do CloudFormation para nomear um grupo de parâmetros.

Obrigatório: não

Tipo: string

Parameters

Uma lista de IDs lógicas de parâmetro que diferenciam maiúsculas de minúsculas a serem incluídas no grupo. Os parâmetros já devem ser definidos na seção `Parameters` do modelo. Um parâmetro pode ser incluído apenas em um grupo de parâmetros.

O console lista os parâmetros que você não associa a um grupo de parâmetros em ordem alfabética no grupo `Other parameters`.

Obrigatório: não

Tipo: lista de valores de strings

ParameterLabels

Um mapeamento de parâmetros e os nomes amigáveis deles que o console do CloudFormation mostra quando uma pilha é criada ou atualizada.

Obrigatório: não

Rótulo do parâmetro

Um rótulo de um parâmetro. O rótulo define um nome ou uma descrição amigável que o console do CloudFormation mostra na página `Especificar parâmetros` quando uma pilha é criada ou atualizada. O rótulo do parâmetro deve ser o ID lógico que diferencia maiúsculas de minúsculas de um parâmetro válido que foi declarado na seção `Parameters` do modelo.

Obrigatório: não

`default`

O rótulo padrão usado pelo console do CloudFormation para nomear um parâmetro.

Obrigatório: não

Tipo: string

Exemplo

O exemplo a seguir define dois `parameter groups`: `Network Configuration` e `Amazon EC2 Configuration`. O grupo `Network Configuration` inclui os parâmetros `VPCID`, `SubnetId` e `SecurityGroupId`, que são definidos na seção `Parameters` do modelo (não mostrado). A ordem na qual o console mostra esses parâmetros é definida pela ordem na qual os parâmetros são

listados, começando com o parâmetro VPCID. O exemplo agrupa e classifica os parâmetros Amazon EC2 Configuration de forma semelhante.

O exemplo também define um rótulo para o parâmetro VPCID. O console mostrará a mensagem Which VPC should this be deployed to? (Em qual VPC isso deve ser implantado?) em vez de mostrar o ID lógico do parâmetro (VPCID).

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups" : [
      {
        "Label" : { "default" : "Network Configuration" },
        "Parameters" : [ "VPCID", "SubnetId", "SecurityGroupID" ]
      },
      {
        "Label" : { "default": "Amazon EC2 Configuration" },
        "Parameters" : [ "InstanceType", "KeyName" ]
      }
    ],
    "ParameterLabels" : {
      "VPCID" : { "default" : "Which VPC should this be deployed to?" }
    }
  }
}
```

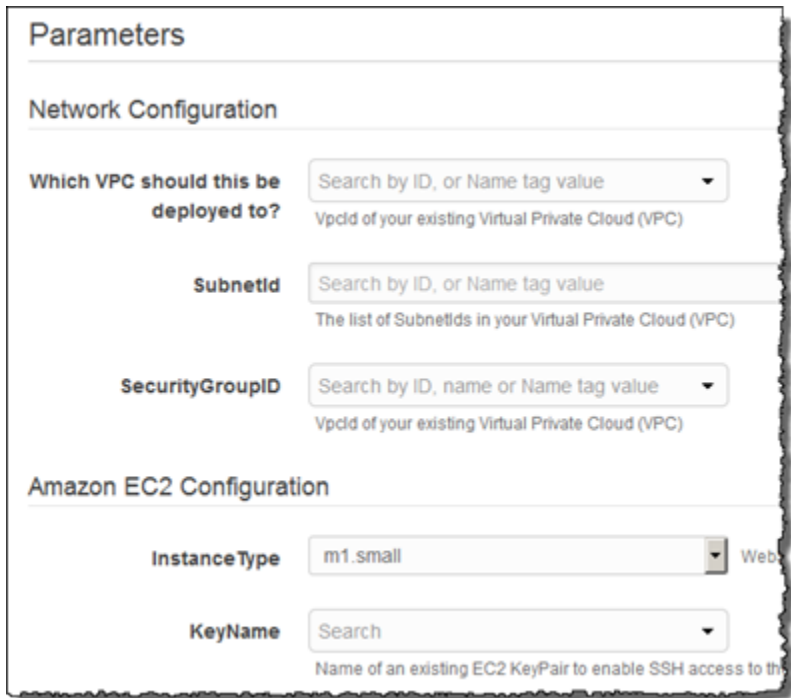
YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Network Configuration"
        Parameters:
          - VPCID
          - SubnetId
          - SecurityGroupID
      - Label:
          default: "Amazon EC2 Configuration"
        Parameters:
          - InstanceType
          - KeyName
```

```
ParameterLabels:
  VPCID:
    default: "Which VPC should this be deployed to?"
```

Grupos de parâmetros no console

Usando a chave de metadados deste exemplo, a figura a seguir mostra como o console exibe os grupos de parâmetros quando uma pilha é criada ou atualizada: Grupos de parâmetros no console



The screenshot shows the AWS CloudFormation console interface. At the top, the 'Parameters' section is expanded, displaying two parameter groups: 'Network Configuration' and 'Amazon EC2 Configuration'. The 'Network Configuration' group contains three parameters: 'Which VPC should this be deployed to?' (with a search dropdown), 'SubnetId' (with a search dropdown), and 'SecurityGroupId' (with a search dropdown). The 'Amazon EC2 Configuration' group contains two parameters: 'InstanceType' (with a dropdown menu showing 'm1.small') and 'KeyName' (with a search dropdown). The interface is clean and professional, with a light gray background and clear labels for each parameter.

Sintaxe de Rules de modelo do CloudFormation

A seção `Rules` é uma parte opcional de um modelo do CloudFormation que permite uma lógica de validação personalizada. Quando incluída, esta seção contém funções de regras que validam os valores dos parâmetros antes que o CloudFormation crie ou atualize quaisquer recursos.

As regras são úteis quando as restrições de parâmetros padrão são insuficientes. Por exemplo, quando o SSL estiver habilitado, um certificado e um nome de domínio deverão ser fornecidos. Uma regra pode garantir que essas dependências sejam atendidas.

Sintaxe

A seção `Rules` usa a seguinte sintaxe:

JSON

A seção Rules de um modelo consiste no nome da chave Rules, seguido por dois-pontos. Você deve usar chaves para delimitar todas as declarações de regras. Se você declarar várias regras, elas serão separadas por vírgulas. Para cada regra, você declara um nome lógico entre aspas seguido por uma vírgula e chaves que incluem a condição da regra e as declarações.

```
{
  "Rules": {
    "LogicalRuleName1": {
      "RuleCondition": {
        "rule-specific intrinsic function": "Value"
      },
      "Assertions": [
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        },
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        }
      ]
    },
    "LogicalRuleName2": {
      "Assertions": [
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        }
      ]
    }
  }
}
```

YAML

```
Rules:
  LogicalRuleName1:
    RuleCondition:
      rule-specific intrinsic function: Value
    Assertions:
      - Assert:
          rule-specific intrinsic function: Value
        AssertDescription: Information about this assert
      - Assert:
          rule-specific intrinsic function: Value
        AssertDescription: Information about this assert
  LogicalRuleName2:
    Assertions:
      - Assert:
          rule-specific intrinsic function: Value
        AssertDescription: Information about this assert
```

Campos de regras

A seção **Rules** pode incluir os seguintes campos.

ID lógico (também chamado de nome lógico)

Um identificador exclusivo para cada regra.

RuleCondition (opcional)

Uma propriedade que determina quando uma regra entra em vigor. Se você não definir uma condição de regra, as declarações da regra sempre entram em vigor. Para cada regra, você pode definir apenas uma condição de regra.

Assertions (obrigatório)

Uma ou mais declarações que especificam os valores aceitáveis para um determinado parâmetro.

Assert

Uma condição que deve ser avaliada como `true`.

AssertDescription

Uma mensagem exibida quando uma asserção falha.

Funções intrínsecas específicas de regras

Para definir suas regras, é necessário usar funções específicas à regra, que são funções que podem ser usadas apenas na seção `Rules` de um modelo. Embora essas funções possam ser aninhadas, o resultado final de uma condição de regra ou asserção deve ser `true` ou `false`.

As seguintes funções de regras estão disponíveis:

- [Fn::And](#)
- [Fn::Contains](#)
- [Fn::EachMemberEquals](#)
- [Fn::EachMemberIn](#)
- [Fn::Equals](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)
- [Fn::RefAll](#)
- [Fn::ValueOf](#)
- [Fn::ValueOfAll](#)

Essas funções são usadas na condição ou nas asserções de uma regra. A propriedade da condição determina se o CloudFormation se aplica a asserções. Se a condição for avaliada como `true`, o CloudFormation avaliará as asserções para verificar se um valor de parâmetro é válido quando um produto provisionado for criado ou atualizado. Se o valor de um parâmetro for inválido, o CloudFormation não criará nem atualizará a pilha. Se a condição for avaliada como `false`, o CloudFormation não verificará o valor do parâmetro e prosseguirá com a operação de pilha.

Exemplos

Tópicos

- [Verificar condicionalmente um valor de parâmetro](#)
- [Validação de parâmetros cruzados](#)

Verificar condicionalmente um valor de parâmetro

No exemplo a seguir, as duas regras verificam o valor do parâmetro `InstanceType`. Dependendo do valor do parâmetro de ambiente (`test` ou `prod`), o usuário deve especificar `t3.medium` ou `t3.large` para o parâmetro `InstanceType`. Os parâmetros `InstanceType` e `Environment` já devem estar declarados na seção `Parameters` do mesmo modelo.

JSON

```
{
  "Rules": {
    "testInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "test"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.medium"],
              {"Ref": "InstanceType"}
            ]
          },
          "AssertDescription": "For a test environment, the instance type must be t3.medium"
        }
      ]
    },
    "prodInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "prod"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.large"],
```

```

        {"Ref": "InstanceType"}
      ]
    },
    "AssertDescription": "For a production environment, the instance type must be
t3.large"
  }
]
}
}
}
}

```

YAML

```

Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':
            - - t3.medium
              - !Ref InstanceType
          AssertDescription: 'For a test environment, the instance type must be
t3.medium'
      - prodInstanceType:
          RuleCondition: !Equals
            - !Ref Environment
            - prod
          Assertions:
            - Assert:
                'Fn::Contains':
                  - - t3.large
                    - !Ref InstanceType
                AssertDescription: 'For a production environment, the instance type must be
t3.large'

```

Validação de parâmetros cruzados

Os modelos de exemplo a seguir demonstram o uso de regras para validações entre parâmetros. Eles criam um site de exemplo executado em um grupo do Auto Scaling atrás de um balanceador de carga. O site está disponível na porta 80 ou 443, dependendo dos parâmetros de entrada. As

instâncias no grupo do Auto Scaling podem ser configuradas para escutar em qualquer porta (com 8888 como padrão).

As regras neste modelo validam os parâmetros de entrada antes da criação da pilha. Elas verificam se todas as sub-redes pertencem à VPC especificada e garantem que, quando o parâmetro UseSSL for definido como Yes, um ARN de certificado SSL e um nome de zona hospedada sejam fornecidos.

Note

Você será cobrado pelos recursos da AWS usados se criar uma pilha com base nesse modelo.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "VpcId": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be the VPC Id of an existing Virtual Private Cloud."
    },
    "Subnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be a list of at least two existing subnets associated with at least two different availability zones."
    },
    "InstanceType": {
      "Description": "WebServer EC2 instance type",
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": ["t2.micro", "t3.micro"],
      "ConstraintDescription": "must be a valid EC2 instance type."
    },
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    }
  }
}
```



```

    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "UseSSL": {
      "AllowedValues": ["Yes", "No"],
      "Default": "No",
      "Description": "Select \"Yes\" to implement SSL, \"No\" to skip (default).",
      "Type": "String"
    },
    "ALBSSLCertificateARN": {
      "Default": "",
      "Description": "[Optional] The ARN of the SSL certificate to be used for the Application Load Balancer",
      "Type": "String"
    },
    "HostedZoneName": {
      "AllowedPattern": "^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-\\-]*[A-Za-z0-9])$",
      "Default": "",
      "Description": "[Optional] The domain name of a valid Hosted Zone on AWS.",
      "Type": "String"
    }
  },
  "Conditions": {
    "UseALBSSL": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]}
  },
  "Rules": {
    "SubnetsInVPC": {
      "Assertions": [
        {
          "Assert": {"Fn::EachMemberEquals": [{"Fn::ValueOf": ["Subnets", "VpcId"]}, {"Ref": "VpcId"}]},
          "AssertDescription": "All subnets must be in the VPC"
        }
      ]
    }
  }
}

```

```

    "ValidateHostedZone": {
      "RuleCondition": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]},
      "Assertions": [
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "ALBSSLCertificateARN"},
""]]}],
          "AssertDescription": "ACM Certificate value cannot be empty if SSL is
required"
        },
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "HostedZoneName"}, ""]]}],
          "AssertDescription": "Route53 Hosted Zone Name is mandatory when SSL is
required"
        }
      ]
    }
  },
  "Resources": {
    "WebServerGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
        "VPCZoneIdentifier": {"Ref": "Subnets"},
        "LaunchTemplate": {
          "LaunchTemplateId": {"Ref": "LaunchTemplate"},
          "Version": {"Fn::GetAtt": ["LaunchTemplate", "LatestVersionNumber"]}
        },
        "MinSize": "2",
        "MaxSize": "2",
        "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
      },
      "CreationPolicy": {
        "ResourceSignal": {"Timeout": "PT15M"}
      },
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "1",
          "MaxBatchSize": "1",
          "PauseTime": "PT15M",
          "WaitOnResourceSignals": true
        }
      }
    },
    "LaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",

```

```

"Metadata": {
  "Comment": "Install a simple application",
  "AWS::CloudFormation::Init": {
    "config": {
      "packages": {"yum": {"httpd": []}},
      "files": {
        "/var/www/html/index.html": {
          "content": {"Fn::Join": ["\n", ["<h1>Congratulations, you have
successfully launched the AWS CloudFormation sample.</h1>"]]},
          "mode": "000644",
          "owner": "root",
          "group": "root"
        },
        "/etc/cfn/cfn-hup.conf": {
          "content": {"Fn::Join": ["", [
            "[main]\n",
            "stack=", {"Ref": "AWS::StackId"}, "\n",
            "region=", {"Ref": "AWS::Region"}, "\n"
          ]]},
          "mode": "000400",
          "owner": "root",
          "group": "root"
        },
        "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
          "content": {"Fn::Join": ["", [
            "[cfn-auto-reloader-hook]\n",
            "triggers=post.update\n",
            "path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init\n",
            "action=/opt/aws/bin/cfn-init -v ",
            "          --stack ", {"Ref": "AWS::StackName"},
            "          --resource LaunchTemplate ",
            "          --region ", {"Ref": "AWS::Region"}, "\n",
            "runas=root\n"
          ]]},
          "mode": "000400",
          "owner": "root",
          "group": "root"
        }
      },
    },
    "services": {
      "sysvinit": {
        "httpd": {
          "enabled": "true",
          "ensureRunning": "true"
        }
      }
    }
  }
}

```

```

    },
    "cfn-hup": {
      "enabled": "true",
      "ensureRunning": "true",
      "files": [
        "/etc/cfn/cfn-hup.conf",
        "/etc/cfn/hooks.d/cfn-auto-reloader.conf"
      ]
    }
  }
},
{
  "Properties": {
    "LaunchTemplateName": {"Fn::Sub": "${AWS::StackName}-launch-template"},
    "LaunchTemplateData": {
      "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
      "SecurityGroupIds": [{"Ref": "InstanceSecurityGroup"}],
      "InstanceType": {"Ref": "InstanceType"},
      "KeyName": {"Ref": "KeyName"},
      "UserData": {
        "Fn::Base64": {"Fn::Join": ["", [
          "#!/bin/bash\n",
          "yum install -y aws-cfn-bootstrap\n",
          "/opt/aws/bin/cfn-init -v ",
          "    --stack ", {"Ref": "AWS::StackName"},
          "    --resource LaunchTemplate ",
          "    --region ", {"Ref": "AWS::Region"}, "\n",
          "/opt/aws/bin/cfn-signal -e $? ",
          "    --stack ", {"Ref": "AWS::StackName"},
          "    --resource WebServerGroup ",
          "    --region ", {"Ref": "AWS::Region"}, "\n"
        ]]}
      ]
    }
  }
},
{
  "ELBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Allow access to the ELB",
      "VpcId": {"Ref": "VpcId"},

```

```

    "SecurityGroupIngress": [{
      "Fn::If": [
        "UseALBSSL",
        {
          "IpProtocol": "tcp",
          "FromPort": 443,
          "ToPort": 443,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        }
      ]
    }]
  },
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Subnets": {"Ref": "Subnets"},
      "SecurityGroups": [{"Ref": "ELBSecurityGroup"}]
    }
  },
  "ALBListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "DefaultActions": [{
        "Type": "forward",
        "TargetGroupArn": {"Ref": "ALBTargetGroup"}
      }],
      "LoadBalancerArn": {"Ref": "ApplicationLoadBalancer"},
      "Port": {"Fn::If": ["UseALBSSL", 443, 80]},
      "Protocol": {"Fn::If": ["UseALBSSL", "HTTPS", "HTTP"]},
      "Certificates": [{
        "Fn::If": [
          "UseALBSSL",
          {"CertificateArn": {"Ref": "ALBSSLCertificateARN"}},
          {"Ref": "AWS::NoValue"}
        ]
      }]
    }
  }
}

```

```

    },
    "ALBTargetGroup": {
      "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties": {
        "HealthCheckIntervalSeconds": 30,
        "HealthCheckTimeoutSeconds": 5,
        "HealthyThresholdCount": 3,
        "Port": 80,
        "Protocol": "HTTP",
        "UnhealthyThresholdCount": 5,
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "SourceSecurityGroupId": {"Fn::Select": [0, {"Fn::GetAtt":
["ApplicationLoadBalancer", "SecurityGroups"]}]}
          },
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": {"Ref": "SSHLocation"}
          }
        ],
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "RecordSet": {
      "Type": "AWS::Route53::RecordSetGroup",
      "Condition": "UseALBSSL",
      "Properties": {
        "HostedZoneName": {"Fn::Join": [ "", [{"Ref": "HostedZoneName"}, "." ] ] },
        "RecordSets": [ {
          "Name": {"Fn::Join": [ "", [
            {"Fn::Select": [ "0", {"Fn::Split": [ ".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]} ] ] ] ] } ] ] },

```

```

        ". ",
        {"Ref": "HostedZoneName"},
        ". "
    ]],
    "Type": "A",
    "AliasTarget": {
        "DNSName": {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]},
        "EvaluateTargetHealth": true,
        "HostedZoneId": {"Fn::GetAtt": ["ApplicationLoadBalancer",
"CanonicalHostedZoneID"]}
    }
    ]}
    }
},
"Outputs": {
    "URL": {
        "Description": "URL of the website",
        "Value": {"Fn::Join": ["", [
            {"Fn::If": [
                "UseALBSSL",
                {"Fn::Join": ["", [
                    "https://",
                    {"Fn::Join": ["", [
                        {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]},
                        ". ",
                        {"Ref": "HostedZoneName"},
                        ". "
                    ]}]
                ]}],
                {"Fn::Join": ["", [
                    "http://",
                    {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
                ]}]
            ]}
        ]}]
    }
}
}
}
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: must be the VPC Id of an existing Virtual Private Cloud.
  Subnets:
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
    ConstraintDescription: >-
      must be a list of at least two existing subnets associated with at least
      two different availability zones. They should be residing in the selected
      Virtual Private Cloud.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  UseSSL:
    AllowedValues:
      - 'Yes'
      - 'No'
    ConstraintDescription: Select Yes to create a HTTPS Listener
    Default: 'No'
    Description: 'Select "Yes" to implement SSL, "No" to skip (default).'
    Type: String
  ALBSSLCertificateARN:
```



```

    Default: ''
    Description: >-
        [Optional] The ARN of the SSL certificate to be used for the Application
        Load Balancer
    Type: String
    HostedZoneName:
        AllowedPattern: >-
            ^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9]|[A-Za-
z0-9][A-Za-z0-9\-\-]*[A-Za-z0-9])$
        Default: ''
        Description: '[Optional] The domain name of a valid Hosted Zone on AWS.'
        Type: String
    Conditions:
        UseALBSSL: !Equals
            - !Ref UseSSL
            - 'Yes'
    Rules:
        SubnetsInVPC:
            Assertions:
                - Assert:
                    'Fn::EachMemberEquals':
                        - 'Fn::ValueOf':
                            - Subnets
                            - VpcId
                        - Ref: VpcId
                    AssertDescription: All subnets must be in the VPC
        ValidateHostedZone:
            RuleCondition: !Equals
                - !Ref UseSSL
                - 'Yes'
            Assertions:
                - Assert: !Not
                    - !Equals
                        - !Ref ALBSSLCertificateARN
                        - ''
                    AssertDescription: ACM Certificate value cannot be empty if SSL is required
                - Assert: !Not
                    - !Equals
                        - !Ref HostedZoneName
                        - ''
                    AssertDescription: Route53 Hosted Zone Name is mandatory when SSL is required
    Resources:
        WebServerGroup:
            Type: AWS::AutoScaling::AutoScalingGroup

```

```

Properties:
  VPCZoneIdentifier: !Ref Subnets
  LaunchTemplate:
    LaunchTemplateId: !Ref LaunchTemplate
    Version: !GetAtt LaunchTemplate.LatestVersionNumber
  MinSize: '2'
  MaxSize: '2'
  TargetGroupARNs:
    - !Ref ALBTargetGroup
CreationPolicy:
  ResourceSignal:
    Timeout: PT15M
UpdatePolicy:
  AutoScalingRollingUpdate:
    MinInstancesInService: '1'
    MaxBatchSize: '1'
    PauseTime: PT15M
    WaitOnResourceSignals: 'true'
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
Metadata:
  Comment: Install a simple application
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
      files:
        /var/www/html/index.html:
          content: !Join
            - |+
            - - >-
              <h1>Congratulations, you have successfully launched the AWS
                CloudFormation sample.</h1>
          mode: '000644'
          owner: root
          group: root
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
          mode: '000400'
          owner: root

```

```

    group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
      content: !Sub |-
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
        runas=root
        mode: '000400'
        owner: root
        group: root
    services:
      sysvinit:
        httpd:
          enabled: 'true'
          ensureRunning: 'true'
        cfn-hup:
          enabled: 'true'
          ensureRunning: 'true'
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf

  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
          /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerGroup --region ${AWS::Region}
      ELBSecurityGroup:
        Type: AWS::EC2::SecurityGroup
        Properties:
          GroupDescription: Allow access to the ELB

```

```
VpcId: !Ref VpcId
SecurityGroupIngress:
  - !If
    - UseALBSSL
    - IpProtocol: tcp
      FromPort: 443
      ToPort: 443
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets: !Ref Subnets
    SecurityGroups:
      - !Ref ELBSecurityGroup
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: !If
      - UseALBSSL
      - 443
      - 80
    Protocol: !If
      - UseALBSSL
      - HTTPS
      - HTTP
    Certificates:
      - !If
        - UseALBSSL
        - CertificateArn: !Ref ALBSSLCertificateARN
        - !Ref 'AWS::NoValue'
ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 30
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 3
```

```

    Port: 80
    Protocol: HTTP
    UnhealthyThresholdCount: 5
    VpcId: !Ref VpcId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access and HTTP access on the inbound port
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId: !Select
          - 0
          - !GetAtt
            - ApplicationLoadBalancer
            - SecurityGroups
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
    VpcId: !Ref VpcId
RecordSet:
  Type: AWS::Route53::RecordSetGroup
  Condition: UseALBSSL
  Properties:
    HostedZoneName: !Join
      - ''
      - - !Ref HostedZoneName
      - .
  RecordSets:
    - Name: !Join
      - ''
      - - !Select
          - '0'
          - !Split
            - .
            - !GetAtt
              - ApplicationLoadBalancer
              - DNSName
        - .
        - !Ref HostedZoneName
      - .
    Type: A

```

```

AliasTarget:
  DNSName: !GetAtt
    - ApplicationLoadBalancer
    - DNSName
  EvaluateTargetHealth: true
  HostedZoneId: !GetAtt
    - ApplicationLoadBalancer
    - CanonicalHostedZoneID

Outputs:
  URL:
    Description: URL of the website
    Value: !Join
      - ''
      - - !If
          - UseALBSSL
          - !Join
            - ''
            - - 'https://'
              - !Join
                - ''
                - - !Select
                    - '0'
                    - !Split
                      - .
                      - !GetAtt
                        - ApplicationLoadBalancer
                        - DNSName
                  - .
                  - !Ref HostedZoneName
                - .
            - !Join
              - ''
              - - 'http://'
                - !GetAtt
                  - ApplicationLoadBalancer
                  - DNSName

```

Sintaxe de Conditions de modelo do CloudFormation

A seção `Conditions` opcional contém instruções que definem as circunstâncias nas quais entidades são criadas ou configuradas. Por exemplo, é possível criar uma condição e associá-la a um recurso ou uma saída para que o CloudFormation crie o recurso ou a saída se a condição for `true`. Da mesma forma, é possível associar uma condição a uma propriedade para que o

CloudFormation define a propriedade como um valor específico se a condição for true. Se a condição for false, o CloudFormation definirá a propriedade para um valor alternativo que você especificar.

Você poderá usar as condições quando quiser reutilizar um modelo para criar recursos em contextos diferentes, como ambientes de teste versus de produção. Por exemplo, no modelo, você pode adicionar um parâmetro de entrada `EnvironmentType` que aceita `prod` ou `test` como entradas. Para o ambiente de `prod`, você pode incluir instâncias do EC2 com determinados recursos. No entanto, para o ambiente de `test`, você deve usar recursos reduzidos para economizar. Essa definição de condição permite que você defina quais recursos são criados e como eles são configurados para cada tipo de ambiente.

Sintaxe

A seção `Conditions` consiste no nome da chave `Conditions`. Cada declaração de condição inclui um ID lógico e uma ou mais funções intrínsecas.

JSON

```
"Conditions": {  
  "LogicalConditionName1": {  
    "Intrinsic function": ...  
  },  
  
  "LogicalConditionName2": {  
    "Intrinsic function": ...  
  }  
}
```

YAML

```
Conditions:  
  LogicalConditionName1:  
    Intrinsic function:  
      ...  
  
  LogicalConditionName2:  
    Intrinsic function:  
      ...
```

Como as condições funcionam

Para usar as condições, siga estas etapas:

1. Adicionar uma definição de parâmetro: defina as entradas que as condições avaliarão na seção `Parameters` do seu modelo. As condições são avaliadas como `true` ou `false` com base nos valores dos parâmetros de entrada. Observe que os pseudoparâmetros estão disponíveis automaticamente e não exigem definição explícita na seção `Parameters`. Para obter mais informações sobre pseudoparâmetros, consulte [Obter valores da AWS usando pseudoparâmetros](#).
2. Adicionar uma definição de condição: defina condições na seção `Conditions` usando funções intrínsecas como `Fn::If` ou `Fn::Equals`. Essas condições determinam quando o CloudFormation cria os recursos associados. As condições podem ser baseadas em:
 - Valores de parâmetros de entrada ou pseudoparâmetros
 - Outras condições
 - Valores de mapeamentoNo entanto, você não pode fazer referência a IDs lógicos de recursos ou seus atributos em condições.
3. Associar condições a recursos ou saídas: referencie condições em recursos ou saídas usando a chave `Condition` e o ID lógico de uma condição. Opcionalmente, use `Fn::If` em outras partes do modelo (como valores de propriedades) para definir valores com base em uma condição. Para obter mais informações, consulte [Usar a chave Condition](#).

O CloudFormation avalia as condições ao criar ou atualizar uma pilha. O CloudFormation cria entidades que são associadas a uma condição verdadeira e ignora entidades que são associadas a uma condição falsa. O CloudFormation também reavalia essas condições em cada atualização de pilha antes de modificar qualquer recurso. As entidades que permanecem associadas a uma condição `true` são atualizadas, enquanto as que se tornam associadas a uma condição `false` são excluídas.

Important

Durante uma atualização de pilha, você não pode atualizar as condições por si só. Você pode atualizar condições apenas ao incluir alterações que adicionam, modificam ou excluem recursos.

Funções intrínsecas da condição

Você pode usar as seguintes funções intrínsecas para definir condições:

- [Fn::And](#)
- [Fn::Equals](#)
- [Fn::ForEach](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)

Note

`Fn::If` só tem suporte no atributo de metadados, no atributo de atualização de política e em valores de propriedade nas seções `Resources` e `Outputs` de um modelo.

Usar a chave **Condition**

Depois que uma condição é definida, é possível aplicá-la em vários lugares no modelo, como `Resources` e `Outputs`, usando a chave `Condition`. A chave `Condition` faz referência ao nome lógico de uma condição e retorna o resultado avaliado da condição especificada.

Tópicos

- [Associar condições a recursos](#)
- [Associar condições a saídas](#)
- [Fazer referência a condições em outras condições](#)
- [Retorne condicionalmente valores de propriedade usando `Fn::If`](#)

Associar condições a recursos

Para criar recursos condicionalmente, adicione a chave `Condition` e o ID lógico da condição como um atributo ao recurso. O CloudFormation cria o recurso somente quando a condição é avaliada como verdadeira.

JSON

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Condition" : "IsProduction",
  "Properties" : {
    "Size" : "100",
    "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ] }
  }
}
```

YAML

```
NewVolume:
  Type: AWS::EC2::Volume
  Condition: IsProduction
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone
```

Associar condições a saídas

Também é possível associar condições a saídas. O CloudFormation cria a saída somente quando a condição associada é avaliada como verdadeira.

JSON

```
"Outputs" : {
  "VolumeId" : {
    "Condition" : "IsProduction",
    "Value" : { "Ref" : "NewVolume" }
  }
}
```

YAML

```
Outputs:
  VolumeId:
    Condition: IsProduction
    Value: !Ref NewVolume
```

Fazer referência a condições em outras condições

Ao definir condições na seção `Conditions`, você pode fazer referência a outras condições usando a chave `Condition`. Isso permite que você crie uma lógica condicional mais complexa combinando várias condições.

No exemplo a seguir, a condição `IsProdAndFeatureEnabled` será avaliada como verdadeira somente se as condições `IsProduction` e `IsFeatureEnabled` forem avaliadas como verdadeiras.

JSON

```
"Conditions": {
  "IsProduction" : {"Fn::Equals" : [{"Ref" : "Environment"}, "prod"]},
  "IsFeatureEnabled" : { "Fn::Equals" : [{"Ref" : "FeatureFlag"}, "enabled"]},
  "IsProdAndFeatureEnabled" : {
    "Fn::And" : [
      {"Condition" : "IsProduction"},
      {"Condition" : "IsFeatureEnabled"}
    ]
  }
}
```

YAML

```
Conditions:
  IsProduction:
    !Equals [!Ref Environment, "prod"]
  IsFeatureEnabled:
    !Equals [!Ref FeatureFlag, "enabled"]
  IsProdAndFeatureEnabled: !And
    - !Condition IsProduction
    - !Condition IsFeatureEnabled
```

Retorne condicionalmente valores de propriedade usando **Fn::If**

Para um controle mais granular, é possível usar a função intrínseca `Fn::If` para retornar condicionalmente um dos dois valores de propriedade em recursos ou saídas. Essa função avalia uma condição e retorna um valor se a condição é verdadeira e outro valor se a condição é falsa.

Valores condicionais da propriedade

O exemplo a seguir demonstra a definição de um tipo de instância do EC2 com base em uma condição do ambiente. Se a condição `IsProduction` for verdadeira, o tipo de instância será definido como `c5.xlarge`. Caso contrário, ele será definido como `t3.small`.

JSON

```
"Properties" : {  
  "InstanceType" : {  
    "Fn::If" : [  
      "IsProduction",  
      "c5.xlarge",  
      "t3.small"  
    ]  
  }  
}
```

YAML

```
Properties:  
  InstanceType: !If  
    - IsProduction  
    - c5.xlarge  
    - t3.small
```

Remoção condicional de propriedades

Também é possível usar o pseudoparametro `AWS::NoValue` como um valor de retorno para remover a propriedade correspondente quando uma condição é falsa.

JSON

```
"DBSnapshotIdentifier" : {  
  "Fn::If" : [  
    "UseDBSnapshot",  
    {"Ref" : "DBSnapshotName"},  
    {"Ref" : "AWS::NoValue"}  
  ]  
}
```

YAML

```
DBSnapshotIdentifier: !If
- UseDBSnapshot
- !Ref DBSnapshotName
- !Ref "AWS::NoValue"
```

Exemplos

Tópicos

- [Criação de recursos com base no ambiente](#)
- [Provisionamento de recursos com várias condições](#)

Criação de recursos com base no ambiente

Os exemplos a seguir provisionam uma instância do EC2 e criam e anexam condicionalmente um novo volume do EBS somente se o tipo de ambiente for prod. Se o ambiente for test, eles simplesmente criam a instância do EC2 sem o volume adicional.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Description": "Environment type",
      "Default": "test",
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ],
      "ConstraintDescription": "must specify prod or test"
    }
  },
  "Conditions": {
    "IsProduction": {
      "Fn::Equals": [
        {
          "Ref": "EnvType"
        }
      ]
    }
  }
}
```

```

        "prod"
    ]
}
},
"Resources": {
    "EC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId": "ami-1234567890abcdef0",
            "InstanceType": "c5.xlarge"
        }
    },
    "MountPoint": {
        "Type": "AWS::EC2::VolumeAttachment",
        "Condition": "IsProduction",
        "Properties": {
            "InstanceId": {
                "Ref": "EC2Instance"
            },
            "VolumeId": {
                "Ref": "NewVolume"
            },
            "Device": "/dev/sdh"
        }
    },
    "NewVolume": {
        "Type": "AWS::EC2::Volume",
        "Condition": "IsProduction",
        "Properties": {
            "Size": 100,
            "AvailabilityZone": {
                "Fn::GetAtt": [
                    "EC2Instance",
                    "AvailabilityZone"
                ]
            }
        }
    }
}
}
}
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Description: Environment type
    Default: test
    Type: String
    AllowedValues:
      - prod
      - test
    ConstraintDescription: must specify prod or test
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-1234567890abcdef0
      InstanceType: c5.xlarge
  MountPoint:
    Type: AWS::EC2::VolumeAttachment
    Condition: IsProduction
    Properties:
      InstanceId: !Ref EC2Instance
      VolumeId: !Ref NewVolume
      Device: /dev/sdh
  NewVolume:
    Type: AWS::EC2::Volume
    Condition: IsProduction
    Properties:
      Size: 100
      AvailabilityZone: !GetAtt
        - EC2Instance
        - AvailabilityZone
```

Provisionamento de recursos com várias condições

Os exemplos a seguir criarão condicionalmente um bucket do S3 se um nome de bucket for fornecido, e anexarão uma política de bucket somente quando o ambiente estiver definido como

prod. Se nenhum nome de bucket for fornecido ou se o ambiente for de test, nenhum recurso será criado.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ]
    },
    "BucketName": {
      "Default": "",
      "Type": "String"
    }
  },
  "Conditions": {
    "IsProduction": {
      "Fn::Equals": [
        {
          "Ref": "EnvType"
        },
        "prod"
      ]
    },
    "CreateBucket": {
      "Fn::Not": [
        {
          "Fn::Equals": [
            {
              "Ref": "BucketName"
            },
            ""
          ]
        }
      ]
    },
    "CreateBucketPolicy": {
      "Fn::And": [
        {
```



```

        "Condition": "IsProduction"
      },
      {
        "Condition": "CreateBucket"
      }
    ]
  },
  "Resources": {
    "Bucket": {
      "Type": "AWS::S3::Bucket",
      "Condition": "CreateBucket",
      "Properties": {
        "BucketName": {
          "Ref": "BucketName"
        }
      }
    },
    "Policy": {
      "Type": "AWS::S3::BucketPolicy",
      "Condition": "CreateBucketPolicy",
      "Properties": {
        "Bucket": {
          "Ref": "Bucket"
        },
        "PolicyDocument": { ... }
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Type: String
    AllowedValues:
      - prod
      - test
  BucketName:
    Default: ''
    Type: String

```

```
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
  CreateBucket: !Not
    - !Equals
      - !Ref BucketName
      - ''
  CreateBucketPolicy: !And
    - !Condition IsProduction
    - !Condition CreateBucket
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Condition: CreateBucket
    Properties:
      BucketName: !Ref BucketName
  Policy:
    Type: AWS::S3::BucketPolicy
    Condition: CreateBucketPolicy
    Properties:
      Bucket: !Ref Bucket
      PolicyDocument: ...
```

Neste exemplo, a condição `CreateBucketPolicy` demonstra como fazer referência a outras condições usando a chave `Condition`. A política é criada somente quando as condições `IsProduction` e `CreateBucket` são avaliadas como verdadeiras.

Note

Para exemplos mais complexos do uso de condições, consulte o tópico [atributo Condition](#) no Guia de referência de modelos do CloudFormation.

Seção Transform do modelo do CloudFormation

A seção `Transform` opcional especifica uma ou mais macros que o CloudFormation usa para processar o modelo de alguma forma.

As macros podem realizar tarefas simples, como localizar e substituir texto, ou podem fazer transformações mais extensas em todo o modelo. O CloudFormation executa as macros na ordem

em que elas são especificadas. Quando você cria um conjunto de alterações, o CloudFormation gera um conjunto de alterações que inclui o conteúdo do modelo processado. Em seguida, você pode revisar as alterações e executar o conjunto de alterações. Consulte mais informações sobre como as macros funcionam em [Executar processamento personalizado em modelos do CloudFormation com macros de modelo](#).

O CloudFormation também é compatível com transformações, que são macros hospedadas pelo CloudFormation. O CloudFormation trata essas transformações da mesma forma que qualquer macro que você cria em termos de ordem de execução e escopo. Para obter mais informações, consulte a [Referência de transformações](#).

Para declarar várias macros, use um formato de lista e especifique uma ou mais macros.

Por exemplo, no modelo de exemplo abaixo, o CloudFormation avalia MyMacro e, depois, AWS::Serverless, e ambas podem processar o conteúdo do modelo inteiro devido à sua inclusão na seção Transform.

```
# Start of processable content for MyMacro and AWS::Serverless
Transform:
  - MyMacro
  - 'AWS::Serverless'
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key": "value"}]
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-1234567890abcdef0
# End of processable content for MyMacro and AWS::Serverless
```

Sintaxe de versão de formato de modelo do CloudFormation

A seção `AWSTemplateFormatVersion` (opcional) identifica a versão do formato do modelo com a qual o modelo é compatível. A `template format version` mais recente é `2010-09-09` e é o único valor válido no momento.

A versão do formato de modelo não é a mesma versão da API. A versão do formato de modelo pode ser alterada independentemente das versões da API.

O valor da declaração da `template format version` deve ser uma sequência literal. Você não pode usar um parâmetro ou uma função para especificar a versão do formato do modelo. Se você não especificar um valor, o CloudFormation assumirá a versão de formato de modelo mais recente. O trecho a seguir é um exemplo de uma declaração de `template format version`:

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
```

Sintaxe de Description de modelo do CloudFormation

A seção `Description` (opcional) permite que você inclua uma string de texto que descreva o modelo. Esta seção deve sempre seguir a seção `template format version`.

O valor da declaração da descrição deve ser uma sequência literal com 0 a 1024 bytes de comprimento. Você não pode usar um parâmetro ou uma função para especificar a descrição. O trecho a seguir é um exemplo de uma declaração da descrição:

Important

Durante a atualização de uma pilha, você não pode atualizar a seção `Description` por si só. Você pode atualizá-la apenas ao incluir alterações que adicionam, modificam ou excluem recursos.

JSON

```
"Description" : "Here are some details about the template."
```

YAML

```
Description: > Here are some details about the template.
```

Crie modelos visualmente com o Infrastructure Composer

O AWS Infrastructure Composer (anteriormente conhecido como Application Composer) ajuda você a compor e configurar visualmente aplicações modernas na AWS. Em vez de escrever código, você pode arrastar e soltar diferentes recursos para criar a aplicação visualmente.

O Infrastructure Composer no modo de console do CloudFormation é a ferramenta recomendada para trabalhar visualmente com modelos do CloudFormation. Essa versão do Infrastructure Composer que você pode acessar no console do CloudFormation é um aprimoramento de uma ferramenta antiga denominada CloudFormation Designer.

Com o Infrastructure Composer no modo de console do CloudFormation, é possível arrastar, soltar, configurar e conectar vários recursos, denominados cartões, em uma tela visual. Essa abordagem visual facilita o design e a edição da arquitetura das aplicações sem necessidade de trabalhar diretamente com os modelos. Para acessar esse modo no console do [CloudFormation](#), selecione Infrastructure Composer no menu de navegação do lado esquerdo.

Para obter mais informações, consulte [Como compor no AWS Infrastructure Composer](#), no Guia do desenvolvedor do AWS Infrastructure Composer.

Por que usar o Infrastructure Composer no modo de console do CloudFormation?

Visualizar os modelos no Infrastructure Composer ajuda a identificar lacunas e áreas a serem aprimoradas nos modelos e na arquitetura das aplicações do CloudFormation. O Infrastructure Composer melhora a experiência de desenvolvimento com a facilidade e a eficiência de criar e modificar pilhas do CloudFormation visualmente. Você pode começar com um rascunho inicial, criar código implantável e incorporar fluxos de trabalho de desenvolvimento ao designer visual no Infrastructure Composer.

Em que esse modo difere do console padrão do Infrastructure Composer?

Embora a versão de console do CloudFormation do Infrastructure Composer tenha atributos semelhantes aos do console padrão do Infrastructure Composer, existem algumas diferenças. Os cartões relacionados ao Lambda (função do Lambda e camada do Lambda) exigem compilações de código e soluções de empacotamento que não estão disponíveis no Infrastructure Composer no modo de console do CloudFormation. A sincronização local também não está disponível nesse modo.

Porém, você pode usar esses cartões relacionados ao Lambda e o atributo de sincronização local no [console do Infrastructure Composer](#) ou no AWS Toolkit for Visual Studio Code. Para obter mais informações, consulte o [Guia do desenvolvedor do AWS Infrastructure Composer](#) e [Infrastructure Composer](#) no Guia do usuário do AWS Toolkit for Visual Studio Code.

Servidor de linguagens do AWS CloudFormation

O servidor de linguagens do AWS CloudFormation fornece recursos para acelerar a criação de infraestrutura como código (IaC) e a implantação de recursos da AWS com segurança e confiança. Ele segue o [protocolo de servidor de linguagens](#) (LSP) para fornecer documentação contextual e sobre preenchimento automático, diagnóstico por meio de validação estática, acesso à definição e ações de código. Além desses recursos tradicionais de servidor de idiomas, o servidor adiciona recursos on-line para explorar e implantar recursos da AWS por meio do CloudFormation. Isso inclui a capacidade de validar e implantar modelos usando conjuntos de alterações, de visualizar diferenças de pilhas, eventos, recursos e saídas, de listar pilhas e procurar recursos por tipo e de inserir o estado ativo dos recursos diretamente nos modelos do CloudFormation. As ferramentas para desenvolvedores e os ambientes de desenvolvimento integrados (IDE) podem aproveitar os recursos do servidor de linguagens para aprimorar a experiência dos desenvolvedores de IaC no CloudFormation.

Tópicos

- [IDEs integrados ao servidor de linguagens do AWS CloudFormation](#)
- [Introdução](#)
- [Inicialização de um projeto do CloudFormation no IDE](#)
- [Código aberto](#)
- [Precisa de ajuda?](#)

IDEs integrados ao servidor de linguagens do AWS CloudFormation

A extensão do [kit de ferramentas da AWS](#) integra o servidor de linguagens do CloudFormation para aprimorar a experiência de criação. Atualmente, essa integração é compatível com os IDEs baseados em VS Code, incluindo:

- [Visual Studio Code](#)
- [Kiro](#)

- [Cursor](#)

Introdução

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: instalar ou atualizar o kit de ferramentas da AWS](#)
- [Etapa 2: acessar o CloudFormation no painel do kit de ferramentas da AWS](#)
- [Etapa 3: validar, testar e refinar seu modelo](#)
- [Etapa 4: navegar pelo modelo](#)
- [Etapa 5: validar e implantar](#)

Pré-requisitos

Antes de começar, verifique se:

- Você está usando um IDE baseado em VS Code em um sistema operacional compatível (macOS, Windows ou Linux).
- Você instalou ou atualizou para a versão mais recente do [kit de ferramentas da AWS](#) para seu IDE.

Alguns recursos no servidor de linguagens do AWS CloudFormation exigem as credenciais configuradas e ativas da Conta da AWS. Você deve fazer login em sua Conta da AWS por meio do kit de ferramentas da AWS usando credenciais válidas.

Etapa 1: instalar ou atualizar o kit de ferramentas da AWS

- Abra o gerenciador de extensões ou plug-ins do seu IDE.
- Pesquise por kit de ferramentas da AWS.
- Instale ou atualize para a versão 3.85.0 ou posterior do kit de ferramentas da AWS para seu IDE.
- Reinicie seu IDE após a instalação.

Observação: após a instalação, o kit de ferramentas da AWS habilita automaticamente o suporte ao IDE do CloudFormation. Quando você instala ou atualiza o kit de ferramentas da AWS pela

primeira vez com a extensão do servidor de linguagens do AWS CloudFormation, você recebe um prompt para conceder permissão à AWS para coletar dados de uso anônimos. Esses dados ajudam a AWS a melhorar o servidor de linguagens do CloudFormation e aprimoram a experiência de criação. Nenhuma informação sensível é coletada, e AWS não registra nem armazena conteúdo de modelos, configurações de recursos ou quaisquer dados identificáveis do cliente. Você pode alterar suas preferências de telemetria a qualquer momento nas configurações do IDE. Os dados de uso coletados focam apenas as interações de recursos e as métricas de performance. Esses insights ajudam a identificar e priorizar melhorias da AWS, como validação mais rápida, preenchimento automático aprimorado e melhores diagnósticos de erros.

Etapa 2: acessar o CloudFormation no painel do kit de ferramentas da AWS

No seu IDE, abra o painel do kit de ferramentas da AWS na barra de atividades. Em Kit de ferramentas da AWS, escolha CLOUDFORMATION.

Seções do painel:

- **Região:** exibe a Região da AWS atual. Você pode alterá-la selecionando o nome da região ou usando o comando AWS CloudFormation: Select Region na paleta de comandos.
- **Ambiente:** indica o ambiente que você selecionou durante o CFN init (por exemplo, desenvolvimento, teste ou produção). Esse valor aparece somente após a conclusão da etapa de seleção do ambiente.
- **Pilhas:** exibe uma lista paginada das pilhas do CloudFormation em sua conta.
 - Clique no ícone de atualização para atualizar a lista de pilhas.
 - Use o ícone + para implantar um novo modelo como uma pilha.
 - Cada entrada da pilha inclui:
 - Visão geral: exibe o resumo e o status da pilha
 - ID da pilha do
 - Descrição
 - Hora de criação e Hora de atualização
 - Status e Motivo do status
 - Eventos, Saídas e Recursos
 - Conjunto de alterações
- **Recursos:** depois de selecionar um tipo de recurso, o painel exibe os recursos da AWS desse tipo em sua conta. Você pode visualizá-los, atualizá-los, copiá-los ou importá-los para o seu modelo.

Etapa 3: validar, testar e refinar seu modelo

À medida que você escreve seu modelo do CloudFormation, o IDE fornece assistência de criação inteligente para ajudar a criar uma infraestrutura precisa e compatível com mais rapidez. O servidor de linguagens do CloudFormation é executado em segundo plano e fornece os seguintes recursos de criação:

- Preenchimento de código: sugere tipos de recursos, parâmetros e propriedades com base nos esquemas do CloudFormation.
- Adicionar recursos existentes da AWS: permite que você importe recursos existentes da sua Conta da AWS para o seu modelo. O IDE usa o [AWS API Cloud Control \(CCAPI\)](#) para recuperar a configuração ativa e as propriedades do recurso, ajudando você a clonar ou a reutilizar a infraestrutura existente em seu modelo.

Para adicionar recursos a seu modelo

- Expandir o painel Recursos do CloudFormation: no painel lateral do kit de ferramentas da AWS, expanda CloudFormation e, em Recursos, clique no ícone Adicionar +.
- Pesquisar um tipo de recurso: para encontrar um tipo de recurso específico da AWS, na barra de pesquisa, digite o tipo de recurso específico da AWS que você deseja adicionar. Exemplo:
 - `AWS::S3::Bucket`
 - `AWS::Lambda::Function`

Você pode selecionar o tipo de recurso usando AWS CloudFormation: Add Resource Types na paleta de comandos.

- Adicionar recursos ao seu modelo: no painel Recursos, uma lista paginada dos recursos da AWS detectados em sua conta é exibida. Se você tiver muitos recursos, somente a primeira página será exibida. Use os controles de navegação na parte inferior do painel para percorrer páginas adicionais e visualizar todos os recursos disponíveis.
- Escolha o recurso que você deseja incluir em seu modelo.
- Você pode inserir um recurso em seu modelo de duas formas, dependendo do seu objetivo:
 - Clonar um recurso existente: crie um novo recurso em seu modelo usando a configuração ativa e as propriedades de um recurso existente da AWS.
 - Importar um recurso existente: insira o recurso real em sua pilha adicionando-o ao seu modelo usando seu estado ativo.

Dicas

- Você pode atualizar o painel Recursos a qualquer momento para ver a lista mais recente de recursos disponíveis em sua conta ou região.
- Se você estiver importando recursos, não adicione um recurso que já pertença a uma pilha existente do CloudFormation na mesma conta.
- Para confirmar se um recurso já é gerenciado pelo CloudFormation, clique no ícone *i* (informações) ao lado do nome do recurso.
- Como alternativa, você pode usar o comando AWS CloudFormation: Get Stack Management Info para verificar se um recurso pertence a uma pilha.

Adicionar recursos relacionados

Você pode adicionar recursos relacionados ao recurso selecionado usando o comando AWS CloudFormation: Add Related Resources by Type. Depois de selecionar um tipo de recurso dos já definidos em seu modelo, o IDE exibirá uma lista de recursos que normalmente estão associados ou dependem desse tipo. Por exemplo, se você selecionar o `AWS::EC2::Instance`, o IDE poderá sugerir a adição de recursos relacionados, como `AWS::EC2::SecurityGroup` ou `AWS::EC2::Subnet`. Esse recurso ajuda você a criar rapidamente componentes de infraestrutura conectados sem procurar manualmente por tipos de recursos compatíveis.

Validação estática

O IDE do CloudFormation fornece validação estática integrada baseada no [AWS CloudFormation Linter \(cfn-lint\)](#) e [AWS CloudFormation Guard](#). Essas validações são executadas em segundo plano à medida que você cria modelos, ajudando a identificar erros de sintaxe, lacunas de conformidade e problemas de práticas recomendadas antes da implantação.

Visão geral da validação estática

Você verá dois tipos de validações estáticas em tempo real no IDE:

- CloudFormation Linter (`cfn-lint`): valida seu modelo em relação às especificações de recursos e regras de esquema do CloudFormation.
- Guard (`cfn-guard`): valida seu modelo em relação às regras de conformidade e aos pacotes de políticas organizacionais.

CloudFormation Linter (cfn-lint)

O CloudFormation Linter é integrado ao IDE para verificar automaticamente a sintaxe e a estrutura do modelo à medida que você digita.

- Validação do esquema: detecta erros de sintaxe e esquema para garantir que seus modelos estejam em conformidade com o esquema de recursos do CloudFormation.
- Destaque de erro: exibe linhas onduladas vermelhas ou amarelas abaixo dos problemas, representando bloqueadores ou avisos de implantação.
- Ajuda ao passar o mouse: quando você passa o mouse sobre um erro, o IDE mostra a mensagem de diagnóstico associada a esse problema. Se uma solução rápida estiver disponível, ela também será oferecida no painel suspenso.

Integração do Guard

O Guard valida seus modelos em relação aos conjuntos de regras que definem as políticas de conformidade e segurança. O IDE executa validações do Guard em tempo real por meio do servidor de linguagens do CloudFormation, fornecendo feedback imediato enquanto você cria modelos.

- Pacotes de regras padrão: o IDE inclui um conjunto pré-registrado de regras do Guard focado nas práticas recomendadas básicas de segurança de recursos e limpeza da configuração. Para saber mais, acesse [o registro de regras de proteção](#).
- Adição de pacotes de regras: para adicionar ou modificar conjuntos de regras, abra Configurações, Guard e selecione ou faça upload de pacotes de regras adicionais do Guard.

Dicas: compreensão dos indicadores de linhas onduladas

- Ondulados azuis: dicas de práticas recomendadas ou recomendações de otimização.
- Ondulados amarelos: avisos sobre problemas de não bloqueio (por exemplo, tags ou parâmetros ausentes).
- Ondulados vermelhos: bloqueadores de implantação, como nomes de propriedades inválidos, campos obrigatórios ausentes ou incompatibilidades de esquema.

Etapa 4: navegar pelo modelo

Quando você clica em um arquivo de modelo no Explorador, o painel de Estrutura exibe automaticamente uma visão estruturada e hierárquica do seu modelo do CloudFormation. Ele

organiza o modelo em seções como **Parameters**, **Resources**, **Outputs** e **Mappings** e mostra cada tipo de recurso e ID lógico. Isso facilita a localização e a navegação rápida para recursos ou parâmetros específicos em grandes modelos.

Você pode usar `!r` para a definição para funções intrínsecas, como `GetAtt` e `Ref`, permitindo que você vá diretamente para o recurso ou parâmetro referenciado em seu modelo. Isso ajuda você a rastrear dependências, a entender as relações de recursos e a fazer edições com mais eficiência.

Etapa 5: validar e implantar

Quando estiver pronto para implantar seu modelo do CloudFormation, abra a paleta de comandos e execute **AWS CloudFormation: Validate and Deploy**. O comando valida seu modelo e, se nenhum erro de bloqueio for encontrado, ele prosseguirá para a fase de implantação. Antes de confirmar, o IDE mostra um resumo do [conjunto de alterações sensível a desvios](#) e uma visualização de diferenças para que você possa revisar todas as alterações propostas.

Como a validação funciona

O IDE executa automaticamente uma [verificação de validação antes da implantação](#) e valida seu modelo quanto a causas comuns de falha, incluindo:

- Sintaxe de propriedade inválida ou incompatibilidades de esquema: esses problemas geralmente são detectados pelo `cfn-lint` durante a criação, mas se um usuário continuar a implantar sem resolvê-los, a validação do tempo de implantação do CloudFormation mostrará os mesmos erros antes que a pilha seja criada ou atualizada.
- O nome do recurso está em conflito com os recursos existentes na sua conta.
- Restrições específicas do serviço, como conflitos de nomes de buckets do S3 ou falta de criptografia.

Se a validação detectar erros, o IDE destacará os problemas diretamente em seu modelo usando linhas onduladas vermelhas ou amarelas e listará os erros no painel **PROBLEMAS**. Cada problema inclui a propriedade ou o recurso específico que causou a falha, junto com uma sugestão de correção. Se não houver erros de bloqueio, você poderá prosseguir para a fase de implantação.

Se forem encontrados avisos (problemas de não bloqueio), uma caixa de diálogo será exibida permitindo que você continue com a implantação ou cancele e faça correções.

O IDE abre um [conjunto de alterações sensível a desvios](#) que exibe todas as diferenças entre seu modelo atual e a configuração da pilha implantada. Isso permite que você revise, confirme ou cancele o conjunto de alterações antes da execução.

Observação: conjuntos de alterações sensíveis a desvios aprimoram o processo de implantação do CloudFormation, permitindo que você gerencie o desvio de pilha com segurança. O desvio de pilha ocorre quando o estado real dos seus recursos é diferente do que está definido no seu modelo do CloudFormation, geralmente devido a alterações manuais feitas por meio do Console de gerenciamento da AWS, da CLI ou do SDK. O [conjunto de alterações sensível a desvios](#) do CloudFormation compara sua configuração de pilha processada com o estado ativo do recurso, e o IDE mostra essas diferenças para que você possa trazer os recursos de volta à conformidade antes da implantação.

Visualizar eventos de pilha

Quando a implantação começar, você poderá navegar até a guia CloudFormation no painel para monitorar o progresso em tempo real. Em Eventos de pilha, você verá uma lista das operações realizadas durante a implantação. Cada evento inclui detalhes como:

- Carimbo de data e hora: a hora em que o evento ocorreu
- Recurso: o recurso específico da AWS que está sendo criado, atualizado ou excluído
- Status: o estado atual da operação (por exemplo, CREATE_IN_PROGRESS, UPDATE_COMPLETE ou ROLLBACK_IN_PROGRESS)
- Motivo: contexto adicional ou mensagens de erro, se aplicável

Você também pode ver os recursos e as saídas da pilha nesse painel. Se você quiser abrir a pilha no Console de gerenciamento da AWS, use o ícone de link externo ao lado do nome da pilha. A visualização Eventos de pilha ajuda você a acompanhar o progresso da implantação, a identificar possíveis problemas e a confirmar quando sua pilha foi concluída com êxito.

Inicialização de um projeto do CloudFormation no IDE

Inicializar um projeto do CloudFormation no IDE ajuda você a configurar um espaço de trabalho estruturado com as pastas, a configuração do ambiente e as credenciais da AWS corretas para que você possa validar e implantar seus modelos de forma confiável. Você pode inicializar um novo projeto do CloudFormation diretamente do IDE para criar essa configuração recomendada.

Para inicializar um projeto CloudFormation:

- Abrir a paleta de comandos
 - No seu IDE, abra a paleta de comandos (Ctrl+Shift+P ou Cmd+Shift+P em macOS).
 - Escolha AWS CloudFormation: CFN Init: Initialize Project.
- Escolher um diretório de projeto
 - Por padrão, o IDE usará seu diretório de trabalho atual.
 - Você pode alterar esse caminho para qualquer pasta em que deseja armazenar seus modelos do CloudFormation.
- Selecione seu perfil de credencial da AWS
 - Você receberá um prompt para escolher um perfil de credencial da AWS. O perfil selecionado será usado para detecção, validações e implantações do ambiente.
- Configure o ambiente
 - Você receberá um prompt para criar ou selecionar um ambiente.
 - Os ambientes definem onde e como seus modelos serão implantados ou validados (por exemplo, desenvolvimento, versão beta ou produção). Você pode usar AWS CloudFormation: CFN Init: Add Environment para selecionar ou alterar seu ambiente.
 - Você pode usar AWS CloudFormation: CFN Init: Remove Environment para remover o ambiente selecionado.
- (Opcional) Importar arquivos de parâmetros
 - Se você já tiver arquivos de parâmetros existentes, o IDE permitirá importá-los durante a inicialização.
 - O IDE detecta automaticamente arquivos compatíveis e os vincula ao seu projeto para uso na validação e implantação de modelos.
- Nomear e finalizar o projeto
 - Forneça um nome de projeto, como ambiente beta, e conclua a configuração.
 - O IDE cria a estrutura inicial do projeto e o arquivo de configuração para você.

Você pode executar validações, visualizar implantações ou alternar entre ambientes diretamente do IDE.

Código aberto

O servidor de linguagens do AWS CloudFormation é de código aberto de acordo com a Licença do Apache-2.0, oferecendo aos clientes total transparência sobre como o diagnóstico de modelos,

a validação do esquema e a análise estática são realizados. Isso reduz o atrito de segurança e conformidade para clientes que precisam de visibilidade no nível da origem antes de adotar ferramentas.

A base de código está disponível publicamente no GitHub: <https://github.com/aws-cloudformation/cloudformation-languageserver/>.

Precisa de ajuda?

Experimente a [comunidade do CloudFormation](#) no AWS re:Post.

Gerar modelos a partir de recursos existentes com o gerador de IaC

Com a infraestrutura do CloudFormation como gerador de código (gerador de IaC), você pode gerar um modelo usando recursos da AWS provisionados na sua conta que ainda não são gerenciados pelo CloudFormation.

Estes são os benefícios do gerador de IaC:

- Coloque aplicações inteiras sob o gerenciamento do CloudFormation ou migre-as para uma aplicação do AWS CDK.
- Gere modelos sem que seja necessário descrever uma propriedade de recurso por propriedade e, depois, traduza-os na sintaxe JSON ou YAML.
- Use o modelo para replicar recursos em uma nova conta ou região.

O processo de geração de IaC consiste nas seguintes etapas:

1. Varredura de recursos: o primeiro passo é iniciar uma verificação dos recursos. Essa varredura abrange toda a região e expira 30 dias. Durante esse período, é possível criar vários modelos com base na mesma verificação.
2. Criação do modelo: para criar o modelo, você tem duas opções:
 - Criar um modelo do zero e adicionar os recursos verificados e recursos relacionados a ele.
 - Usar uma pilha existente do CloudFormation como ponto de partida e adicionar os recursos verificados pela varredura e os recursos relacionados ao modelo.
3. Importação de recursos: use o modelo para importar os recursos como uma pilha do CloudFormation ou migrá-los para uma aplicação do AWS CDK.

O recurso de gerador de IaC está disponível em todas as regiões comerciais e oferece suporte a muitos tipos de recursos comuns da AWS. Para obter uma lista completa de recursos suportados, consulte [Suporte a tipos de recursos](#).

Tópicos

- [Considerações](#)
- [Permissões do IAM necessárias para varrer recursos](#)
- [Comandos comumente usados para geração, exclusão e gerenciamento de modelos](#)
- [Migrar um modelo para o AWS CDK](#)
- [Iniciar uma verificação de recursos com o gerador de IaC do CloudFormation](#)
- [Visualize o resumo da verificação no console do CloudFormation](#)
- [Criar um modelo do CloudFormation a partir de recursos verificados com o gerador de IaC](#)
- [Criar uma pilha do CloudFormation com base nos recursos verificados](#)
- [Resolver propriedades somente gravação](#)

Considerações

É possível gerar modelos JSON ou YAML para recursos da AWS aos quais você tem acesso de leitura. Os modelos do recurso de gerador de IaC preparam recursos de nuvem de forma confiável e rápida, sem que seja necessário descrever um recurso propriedade por propriedade.

A tabela a seguir lista as cotas disponíveis para o recurso de geração de IaC.

Nome	Varredura completa	Varredura parcial
Número máximo de usuários que podem ser processados em uma verificação	100.000	100.000
Número de verificações por dia (para verificações com menos de 10.000 recursos)	10	10
Número de verificações por dia (para verificações com mais de 10.000 recursos)	1	1
Número simultâneo de modelos gerados por conta	5	5

Nome	Varredura completa	Varredura parcial
Número simultâneo de recursos modelados para uma geração de modelo	5	5
Número total de recursos que podem ser modelados em um modelo	500	500
Número máximo por conta de modelos gerados	1.000	1.000

 Important

O gerador de IaC só é compatível com recursos da AWS compatíveis com a API do Cloud Control na sua região. Para obter mais informações, consulte [Suporte a tipos de recursos](#).

Permissões do IAM necessárias para varrer recursos

Para varrer recursos com o gerador de IaC, a entidade principal do IAM (usuário, perfil ou grupo) deve ter:

- Permissões de varredura do CloudFormation
- Permissões de leitura nos serviços da AWS de destino

O escopo da varredura se limita aos recursos aos quais você tem acesso de leitura. Permissões ausentes não causarão falha na verificação, mas excluirão esses recursos.

Para obter um exemplo de política do IAM que concede permissões para varrer e gerenciar modelos, consulte [Permitir todas as operações do gerador de IaC](#).

Comandos comumente usados para geração, exclusão e gerenciamento de modelos

Os comandos comumente usados para trabalhar com o gerador de IaC incluem:

- [start-resource-scan](#) para iniciar uma verificação dos recursos na conta em uma Região da AWS.

- [describe-resource-scan](#) para monitorar o andamento de uma verificação de recursos.
- [list-resource-scans](#) para listar os escaneamentos de recursos em uma Região da AWS.
- [list-resource-scan-resources](#) para listar os recursos encontrados durante a verificação de recursos.
- [list-resource-scan-related-resources](#) para listar os recursos relacionados aos recursos verificados.
- [create-generated-template](#) para gerar um modelo do CloudFormation de um conjunto de recursos verificados.
- [update-generated-template](#) para atualizar o modelo gerado.
- [describe-generated-template](#) para retornar informações sobre um modelo gerado.
- [list-generated-templates](#) para listar todos os modelos gerados em sua conta e região atual.
- [delete-generated-template](#) para excluir um modelo gerado.

Migrar um modelo para o AWS CDK

O AWS Cloud Development Kit (AWS CDK) é um framework de desenvolvimento de software de código aberto que você pode usar para desenvolver, gerenciar e implantar recursos do CloudFormation usando linguagens de programação conhecidas.

A CLI do AWS CDK fornece uma integração com o gerador de IaC. Use o comando `cdk migrate` da CLI do AWS CDK para gerar um modelo do CloudFormation e criar uma aplicação do CDK que contenha os recursos. Depois, é possível usar o AWS CDK para gerenciar os recursos e implantar no CloudFormation.

Para obter mais informações, consulte [Migrar para o AWS CDK](#) no Guia do desenvolvedor do AWS Cloud Development Kit (AWS CDK).

Iniciar uma verificação de recursos com o gerador de IaC do CloudFormation

Antes de criar um modelo a partir dos recursos existentes, você deve iniciar uma verificação de recursos para descobrir os recursos atuais e seus relacionamentos.

Você pode iniciar uma verificação de recursos usando uma das seguintes opções. Para usuários iniciantes do gerador de IaC, recomendamos a primeira opção.

- Verificar todos os recursos (verificação completa): verifica todos os recursos existentes na conta e região atuais. Esse processo de verificação pode levar até 10 minutos para 1.000 recursos.

- Verificar recursos específicos (verificação parcial): selecione manualmente quais tipos de recursos serão verificados na conta e região atuais. Essa opção fornece um processo de verificação mais rápido e focado, tornando-a ideal para o desenvolvimento iterativo de modelos.

Concluída a verificação, você pode escolher quais recursos e os recursos relacionados devem ser incluídos ao gerar o modelo. Ao usar a verificação parcial, os recursos relacionados apenas estarão disponíveis durante a geração do modelo se:

- Você os tiver selecionado especificamente antes de iniciar a verificação, ou
- Tiverem sido necessários para descobrir os tipos de recursos selecionados.

Por exemplo, se você selecionar `AWS::EKS::Nodegroup` sem selecionar `AWS::EKS::Cluster`, o gerador de IaC incluirá automaticamente os recursos `AWS::EKS::Cluster` na verificação, pois a descoberta do grupo de nós requer a descoberta inicial do cluster. Em todos os outros casos, a verificação incluirá apenas os recursos que você selecionar especificamente.

Note

Antes de continuar, confirme que você tem as permissões necessárias para trabalhar com o gerador de IaC. Para obter mais informações, consulte [Permissões do IAM necessárias para varrer recursos](#).

Tópicos

- [Iniciar uma verificação de recursos \(console\)](#)
- [Iniciar uma verificação de recursos \(AWS CLI\)](#)

Iniciar uma verificação de recursos (console)

Para iniciar uma verificação de todos os tipos de recursos (verificação completa)

1. Abra a [página do gerador de IaC](#) do console do CloudFormation.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS que contém os recursos que devem ser verificados.
3. No painel Verificações, escolha Iniciar nova verificação e, em seguida, escolha Verificar todos os recursos.

Para iniciar uma verificação de recursos de tipos de recursos específicos (verificação parcial)

1. Abra a [página do gerador de laC](#) do console do CloudFormation.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS que contém os recursos que devem ser verificados.
3. No painel Verificações, escolha Iniciar nova verificação e escolha Verificar recursos específicos.
4. Na caixa de diálogo Iniciar verificação parcial, selecione até 100 tipos de recursos e escolha Iniciar verificação.

Iniciar uma verificação de recursos (AWS CLI)

Para iniciar uma verificação de todos os tipos de recursos (verificação completa)

Use o seguinte comando [start-resource-scan](#). Substitua *us-east-1* pela Região da AWS que contém os recursos a serem verificados.

```
aws cloudformation start-resource-scan --region us-east-1
```

Em caso de êxito, esse comando retornará o ARN da verificação. Anote o ARN na propriedade ResourceScanId. Ele será necessário para criar o modelo.

```
{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60"
}
```

Para iniciar uma verificação de recursos de tipos de recursos específicos (verificação parcial)

1. Use o seguinte comando [cat](#) para armazenar os tipos de recursos que deseja verificar em um arquivo JSON nomeado `config.json` no seu diretório pessoal. A seguir, um exemplo de configuração de verificação que examina instâncias do Amazon EC2, grupos de segurança e todos os recursos do Amazon S3.

```
$ cat > config.json
[
  {
    "Types": [
      "AWS::EC2::Instance",
```

```

        "AWS::EC2::SecurityGroup",
        "AWS::S3::*"
    ]
}
]

```

2. Use o comando [start-resource-scan](#) com a opção `--scan-filters`, junto com o arquivo `config.json` criado, para iniciar a verificação parcial. Substitua `us-east-1` pela Região da AWS que contém os recursos a serem verificados.

```

aws cloudformation start-resource-scan --scan-filters file://config.json --
region us-east-1

```

Em caso de êxito, esse comando retornará o ARN da verificação. Anote o ARN na propriedade `ResourceScanId`. Ele será necessário para criar o modelo.

```

{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60"
}

```

Para monitorar o andamento da verificação de um recurso

Use o comando [describe-resource-scan](#). Na opção `--resource-scan-id`, substitua o ARN de amostra pelo ARN real.

```

aws cloudformation describe-resource-scan --region us-east-1 \
--resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60

```

Se houver êxito, o comando gerará uma saída semelhante à seguinte:

```

{
  "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
  "Status": "COMPLETE",
  "StartTime": "2023-08-21T03:10:38.485000+00:00",
  "EndTime": "2023-08-21T03:20:28.485000+00:00",
  "PercentageCompleted": 100.0,
}

```

```

    "ResourceTypes": [
      "AWS::CloudFront::CachePolicy",
      "AWS::CloudFront::OriginRequestPolicy",
      "AWS::EC2::DHCPOptions",
      "AWS::EC2::InternetGateway",
      "AWS::EC2::KeyPair",
      "AWS::EC2::NetworkAcl",
      "AWS::EC2::NetworkInsightsPath",
      "AWS::EC2::NetworkInterface",
      "AWS::EC2::PlacementGroup",
      "AWS::EC2::Route",
      "AWS::EC2::RouteTable",
      "AWS::EC2::SecurityGroup",
      "AWS::EC2::Subnet",
      "AWS::EC2::SubnetCidrBlock",
      "AWS::EC2::SubnetNetworkAclAssociation",
      "AWS::EC2::SubnetRouteTableAssociation",
      ...
    ],
    "ResourcesRead": 676
  }

```

Para uma verificação parcial, a saída será semelhante à seguinte:

```

{
  "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
  "Status": "COMPLETE",
  "StartTime": "2025-03-06T18:24:19.542000+00:00",
  "EndTime": "2025-03-06T18:25:23.142000+00:00",
  "PercentageCompleted": 100.0,
  "ResourceTypes": [
    "AWS::EC2::Instance",
    "AWS::EC2::SecurityGroup",
    "AWS::S3::Bucket",
    "AWS::S3::BucketPolicy"
  ],
  "ResourcesRead": 65,
  "ScanFilters": [
    {
      "Types": [
        "AWS::EC2::Instance",
        "AWS::EC2::SecurityGroup",

```

```
    "AWS::S3:*"  
  ]  
}  
]  
}
```

Consulte uma descrição dos campos na saída em [DescribeResourceScan](#) na Referência da API do AWS CloudFormation.

Visualize o resumo da verificação no console do CloudFormation

Depois que a verificação for concluída, você poderá visualizar os recursos encontrados durante a verificação para ajudar a identificar a concentração de recursos em diferentes tipos de produtos.

Para visualizar informações sobre recursos encontrados durante a verificação

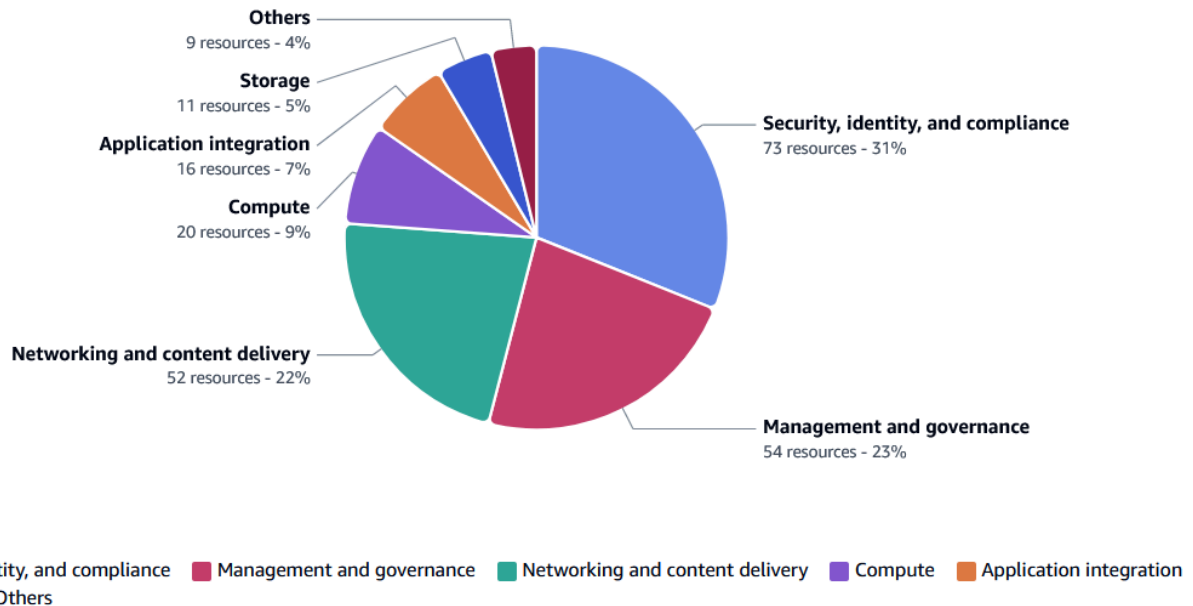
1. Abra a [página do gerador de IaC](#) do console do CloudFormation.
2. Na barra de navegação, na parte superior da tela, selecione a Região da AWS que contém a varredura de recursos a ser visualizada.
3. Na página de navegação, escolha Gerador de IaC.
4. Em Detalhamento dos recursos verificados, você encontrará um detalhamento visual dos recursos verificados por tipo de produto, por exemplo, Computação e Armazenamento.
5. Para personalizar o número de tipos de produtos exibidos, escolha Filtrar dados exibidos. Isso ajuda você a personalizar a visualização para se concentrar nos tipos de produtos nos quais tem mais interesse.
6. No lado direito da página está o painel Detalhes resumo da verificação. Para abrir o painel, selecione o ícone de painel aberto.

Scanned resources breakdown 235

View a visual breakdown of your scanned resources by product types. Select a product type for a further breakdown.

Filter displayed data

Filter by product type ▼



Criar um modelo do CloudFormation a partir de recursos verificados com o gerador de IaC

Este tópico explica como criar um modelo a partir de recursos que foram verificados pelo recurso de gerador de IaC.

Criar um modelo com base nos recursos verificados (console)

Para criar um modelo de pilha com recursos verificados

1. Abra a [página do gerador de IaC](#) do console do CloudFormation.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS que contém os recursos verificados.
3. No painel Modelos, escolha Criar modelo.
4. Selecione Iniciar de um novo modelo.
 - a. Em Nome do modelo, forneça um nome para o modelo.

- b. (Opcional) Configure sua Política de exclusão e a Política de substituição de atualização.
 - c. Escolha Próximo para adicionar recursos encontrados pela varredura ao modelo.
5. Em Adicionar recursos verificados, navegue pela lista de recursos verificados e selecione os recursos que deseja adicionar ao modelo. É possível filtrar os recursos por identificador, tipo de recurso ou tags. Os filtros são mutuamente inclusivos.
 6. Após adicionar todos os recursos necessários para o modelo, escolha Próximo para sair da página Adicionar recursos verificados e prossiga para a página Adicionar recursos relacionados.
 7. Revise uma lista recomendada de recursos relacionados. Recursos relacionados, como instâncias do Amazon EC2 e grupos de segurança, são interdependentes e normalmente pertencem à mesma workload. Selecione os usuários relacionados que deseja incluir no modelo gerado.

 Note

Sugerimos adicionar todos os recursos relacionados a esse modelo.

8. Revise os detalhes do modelo, recursos encontrados pela varredura e recursos relacionados.
9. Escolha Criar modelo para sair da página Revisar e criar e criar o modelo.

Criar um modelo com base nos recursos verificados (AWS CLI)

Para criar um modelo de pilha com recursos verificados

1. Use o comando [list-resource-scan-resources](#) para listar os recursos encontrados pela verificação, especificando a opção `--resource-identifier` para limitar a saída, se assim desejar. Na opção `--resource-scan-id`, substitua o ARN de amostra pelo ARN real.

```
aws cloudformation list-resource-scan-resources \
  --resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resource-identifier MyApp
```

Este é um exemplo de resposta, onde `ManagedByStack` indica se o CloudFormation já está gerenciando o recurso. Copie a saída. Você precisa dele para a próxima etapa.

```
{
  "Resources": [
```

```
{
  "ResourceType": "AWS::EKS::Cluster",
  "ResourceIdentifier": {
    "ClusterName": "MyAppClusterName"
  },
  "ManagedByStack": false
},
{
  "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
  "ResourceIdentifier": {
    "AutoScalingGroupName": "MyAppASGName"
  },
  "ManagedByStack": false
}
]
```

Consulte uma descrição dos campos na saída em [ScannedResource](#) na Referência de API do AWS CloudFormation.

2. Use o comando `cat` para armazenar os tipos e identificadores de recursos em um arquivo JSON chamado `resources.json` em seu diretório inicial. O exemplo a seguir é um JSON baseado na saída do exemplo da etapa anterior.

```
$ cat > resources.json
[
  {
    "ResourceType": "AWS::EKS::Cluster",
    "ResourceIdentifier": {
      "ClusterName": "MyAppClusterName"
    }
  },
  {
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
    "ResourceIdentifier": {
      "AutoScalingGroupName": "MyAppASGName"
    }
  }
]
```

3. Use o comando [list-resource-scan-related-resources](#), juntamente com o arquivo `resources.json` que você criou, para listar os recursos relacionados aos recursos verificados.

```
aws cloudformation list-resource-scan-related-resources \
  --resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resources file://resources.json
```

Este é um exemplo de resposta, onde `ManagedByStack` indica se o CloudFormation já está gerenciando o recurso. Adicione esses recursos ao arquivo JSON criado na etapa anterior. Ele será necessário para criar o modelo.

```
{
  "RelatedResources": [
    {
      "ResourceType": "AWS::EKS::Nodegroup",
      "ResourceIdentifier": {
        "NodegroupName": "MyAppNodegroupName"
      },
      "ManagedByStack": false
    },
    {
      "ResourceType": "AWS::IAM::Role",
      "ResourceIdentifier": {
        "RoleId": "arn:aws:iam::account-id:role/MyAppIAMRole"
      },
      "ManagedByStack": false
    }
  ]
}
```

Consulte uma descrição dos campos na saída em [ScannedResource](#) na Referência de API do AWS CloudFormation.

Note

A lista de entrada de recursos não pode exceder um tamanho de 100. Para listar recursos relacionados para mais de 100 recursos, execute o comando `list-resource-scan-related-resources` em lotes de 100 e consolide os resultados. Tenha em mente que a saída pode conter recursos duplicados na lista.

4. Use o comando [create-generated-template](#) para criar um novo modelo de pilha da seguinte forma, com as seguintes modificações:

- Substitua *us-east-1* pelo Região da AWS que contém os recursos verificados.
- Substitua *MyTemplate* pelo nome do modelo a ser criado.

```
aws cloudformation create-generated-template --region us-east-1 \  
--generated-template-name MyTemplate \  
--resources file://resources.json
```

Veja a seguir um exemplo de arquivo `resources.json`.

```
[  
  {  
    "ResourceType": "AWS::EKS::Cluster",  
    "LogicalResourceId": "MyCluster",  
    "ResourceIdentifier": {  
      "ClusterName": "MyAppClusterName"  
    }  
  },  
  {  
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",  
    "LogicalResourceId": "MyASG",  
    "ResourceIdentifier": {  
      "AutoScalingGroupName": "MyAppASGName"  
    }  
  },  
  {  
    "ResourceType": "AWS::EKS::Nodegroup",  
    "LogicalResourceId": "MyNodegroup",  
    "ResourceIdentifier": {  
      "NodegroupName": "MyAppNodegroupName"  
    }  
  },  
  {  
    "ResourceType": "AWS::IAM::Role",  
    "LogicalResourceId": "MyRole",  
    "ResourceIdentifier": {  
      "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"  
    }  
  }  
]
```

```
]
```

Em caso de êxito, o comando retornará a resposta a seguir.

```
{
  "Arn":
    "arn:aws:cloudformation:region:account-id:generatedtemplate/7fc8512c-d8cb-4e02-b266-d39c48344e48",
  "Name": "MyTemplate"
}
```

Criar uma pilha do CloudFormation com base nos recursos verificados

Após criar o modelo, você poderá visualizar o modelo gerado com o Infrastructure Composer antes de criar a pilha e importar os recursos verificados. Isso ajuda você a visualizar a arquitetura completa da aplicação com os recursos e seus relacionamentos. Para obter mais informações sobre o Infrastructure Composer, consulte [Crie modelos visualmente com o Infrastructure Composer](#)

Para criar a pilha e importar os recursos verificados

1. Abra a [página do gerador de IaC](#) do console do CloudFormation.
2. Na barra de navegação, na parte superior da tela, escolha a Região da AWS do modelo.
3. Escolha a guia Modelos e depois escolha o nome do modelo para ver mais informações.
4. Na guia Definição do modelo, na parte superior da seção Modelo, você pode alternar o modelo da sintaxe YAML para a JSON com base na sua preferência.
5. Revise os detalhes do modelo para garantir que tudo esteja configurado corretamente. Para facilitar a análise e a compreensão do modelo, você pode alternar da visualização de código padrão para uma visualização gráfica da infraestrutura descrita no modelo usando o Infrastructure Composer. Para fazer isso, em Modelo, escolha Tela em vez de Modelo.

Ações de tela

- Para focar nos detalhes de um recurso específico no seu modelo, clique duas vezes em um cartão para abrir o painel Propriedades do recurso.
- Para organizar e dispor visualmente os cartões na tela, escolha Organizar no canto superior esquerdo da tela.
- Para ampliar e reduzir a tela, use os controles de zoom no canto inferior direito da tela.

6. Para visualizar um recurso específico no console, escolha a guia Recursos do modelo e, em seguida, escolha a ID física do recurso que você deseja examinar. Isso leva você ao console desse recurso específico. Você também pode adicionar, remover e ressincronizar recursos na definição do modelo na guia Recursos do modelo.
7. Na guia Definição do modelo, o gerador de IaC pode emitir avisos sobre recursos que contêm propriedades somente gravação. Depois de revisar os avisos, é possível baixar o modelo gerado e fazer as alterações necessárias. Para obter mais informações, consulte [Resolver propriedades somente gravação](#).
8. Quando estiver satisfeito com a definição do modelo, na guia Definição do modelo, selecione Importar para pilha e, em seguida, selecione Avançar.
9. No painel Especificar pilha da página Especificar detalhes da pilha, insira o nome da pilha e escolha Próximo.
10. Revise e insira os parâmetros da pilha. Escolha Próximo.
11. Revise suas opções na página Revisar alterações e escolha Próximo.
12. Revise seus detalhes na página Revisar e importar e escolha Importar recursos.

Resolver propriedades somente gravação

Com o gerador de IaC do CloudFormation, é possível gerar um modelo usando recursos provisionados em sua conta que ainda não são gerenciados pelo CloudFormation. No entanto, certas propriedades de recursos são designadas como somente gravação, o que significa que elas podem ser gravadas, mas não podem ser lidas pelo CloudFormation, por exemplo, uma senha de banco de dados.

Ao gerar modelos do CloudFormation com base em recursos existentes, as propriedades somente gravação representam um desafio. Na maioria dos casos, o CloudFormation converte essas propriedades em parâmetros no modelo gerado. Isso permite a você inserir as propriedades como valores de parâmetros durante as operações de importação. No entanto, há cenários em que essa conversão não é possível, e o CloudFormation trata esses casos de forma diferente.

Propriedades mutuamente excludentes

Alguns recursos têm vários conjuntos de propriedades mutuamente excludentes, e pelo menos alguns deles são somente gravação. Nesses casos, o gerador de IaC não pode determinar qual conjunto de propriedades excludentes foi aplicado ao recurso durante a criação. Por exemplo,

você pode fornecer o código para um [AWS::Lambda::Function](#) usando um desses conjuntos de propriedades.

- Code/S3Bucket, Code/S3Key e, opcionalmente, Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

Todas essas propriedades são somente gravação. O gerador de IaC seleciona um dos conjuntos de propriedades exclusivos e os adiciona ao modelo gerado. Parâmetros são adicionados para cada uma das propriedades somente gravação. Os nomes dos parâmetros incluem OneOf e as descrições dos parâmetros indicam que a propriedade correspondente pode ser substituída por outras propriedades exclusivas. O gerador de IaC define um tipo de aviso de MUTUALLY_EXCLUSIVE_PROPERTIES para as propriedades incluídas.

Tipos mutuamente excludentes

Em alguns casos, uma propriedade somente gravação pode ser de vários tipos de dados. Por exemplo, a propriedade Body de [AWS::ApiGateway::RestApi](#) pode ser um object ou um string. Quando esse é o caso, o gerador de IaC inclui a propriedade no modelo gerado usando o tipo de string e define um tipo de alerta de MUTUALLY_EXCLUSIVE_TYPES.

Propriedades do Array

Se uma propriedade somente gravação tiver um tipo de array, o gerador de IaC não poderá incluí-la no modelo gerado porque os parâmetros só podem ser valores escalares. Nesse caso, a propriedade é omitida do modelo e um tipo de aviso de UNSUPPORTED_PROPERTIES é definido.

Propriedades opcionais

Para propriedades somente gravação opcionais, o gerador de IaC não consegue detectar se a propriedade foi usada ao configurar o recurso. Nesse caso, a propriedade é omitida do modelo gerado e um tipo de aviso de UNSUPPORTED_PROPERTIES é definido.

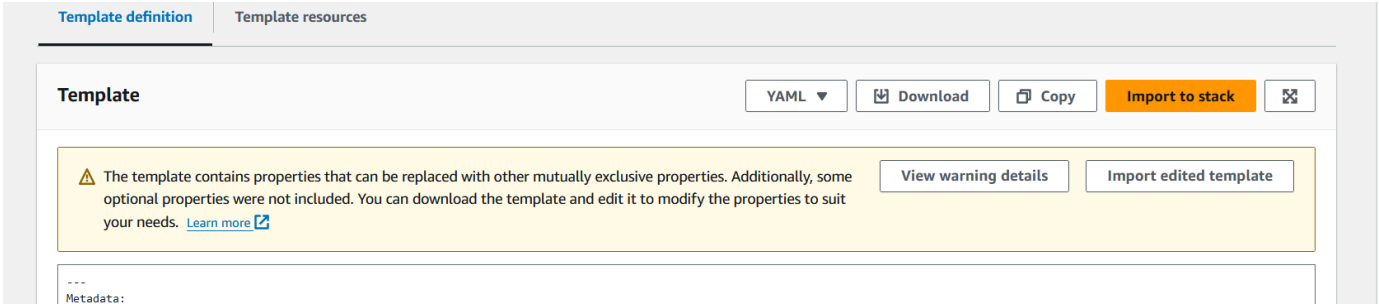
Avisos e próximas etapas

Para determinar quais propriedades são somente de gravação, é preciso observar os avisos retornados pelo console do gerador de IaC. A [Referência de propriedades e recursos da AWS](#) não indica se uma propriedade é somente gravação ou se é compatível com vários tipos.

Como alternativa, você pode ver quais propriedades são somente para gravação nos esquemas do provedor de recursos. Para baixar os esquemas do provedor de recursos, consulte os [Esquemas do provedor de recursos do CloudFormation](#).

Para resolver problemas com propriedades somente para gravação

1. Abra a [página do gerador de IaC](#) do console do CloudFormation.
2. Na barra de navegação, na parte superior da tela, escolha a Região da AWS do modelo.
3. Escolha a guia Modelos e, em seguida, escolha o nome do modelo que você criou.
4. Na guia Definição do modelo, quando o modelo gerado inclui recursos com propriedades somente de gravação, o console do gerador de IaC exibe um aviso com um resumo dos tipos de problemas. Por exemplo:



5. Selecione Exibir detalhes do aviso para obter mais detalhes. Os recursos com propriedades somente gravação são identificados pelo ID lógico usado no modelo gerado e no tipo de recurso.

Use a lista de avisos para identificar recursos com propriedades somente gravação e examine cada recurso para determinar quais alterações (se houver) precisam ser feitas no modelo gerado.

Resource warnings (1)

▼ LambdaFunction001aClambda002fD6K (AWS::Lambda::Function) [Learn more](#)

Unsupported properties

1. SnapStart/ApplyOn

Applying SnapStart setting on function resource type.Possible values: [PublishedVersions, None] This property can be replaced with other types

2. Code/S3ObjectVersion

For versioned objects, the version of the deployment package object to use. This property can be replaced with other exclusive properties

Mutually exclusive properties

1. Code/S3Bucket

An Amazon S3 bucket in the same AWS Region as your function. The bucket can be in a different AWS account. This property can be replaced with other exclusive properties

2. Code/S3Key

The Amazon S3 key of the deployment package. This property can be replaced with other exclusive properties

6. Caso o modelo precise ser atualizado para resolver problemas com propriedades somente de gravação, faça o seguinte:
 - a. Selecione Download para baixar uma cópia do modelo.
 - b. Edite o modelo.
 - c. Depois que as alterações forem concluídas, você poderá escolher o botão Importar modelo editado para continuar o processo de importação.

Como resolver problemas com propriedades somente gravação em recursos

AWS::ApiGateway::RestAPI

Este tópico explica como resolver problemas com propriedades somente gravação em recursos [AWS::ApiGateway::RestApi](#) ao usar o gerador de IaC.

Problema

Quando um modelo gerado contém recursos `AWS::ApiGateway::RestApi`, avisos são gerados para informar que as propriedades `Body`, `BodyS3Location` e `CloneFrom` são identificadas como `UNSUPPORTED_PROPERTIES`. Isso ocorre porque essas são propriedades opcionais somente gravação. O gerador de IaC não sabe se essas propriedades já foram aplicadas ao recurso. Portanto, ele omite essas propriedades no modelo gerado.

Resolução

Para definir a propriedade `Body` da API REST, atualize o modelo gerado.

1. Use a ação de API [GetExport](#) do Amazon API Gateway para baixar a API. Por exemplo, usando o comando [aws apigateway get-export](#) da AWS CLI. Para obter mais informações, consulte [Exportar uma API REST do API Gateway](#) no Guia do desenvolvedor do API Gateway.
2. Recupere a `Body` propriedade da resposta da ação da API `GetExport`. Carregue-a para um bucket do Amazon S3.
3. Baixe o modelo gerado.
4. Adicione as propriedades `BodyS3Location/Bucket` e `BodyS3Location/Key` ao modelo, especificando o nome do bucket e a chave em que `Body` está armazenado.
5. Abra o modelo gerado no console do gerador de IaC e escolha Importar modelo editado.

Como resolver problemas com propriedades somente gravação em recursos

AWS::Lambda::Function

Este tópico explica como resolver problemas com propriedades somente gravação em recursos [AWS::Lambda::Function](#) ao usar o gerador de IaC.

Problema

O recurso `AWS::Lambda::Function` tem três conjuntos de propriedades mutuamente excludentes para especificar o código Lambda:

- As propriedades `Code/S3Bucket` e `Code/S3Key`, e opcionalmente, a propriedade `Code/S3ObjectVersion`
- `Code/ImageUri` propriedade
- `Code/ZipFile` propriedade

Somente um desses conjuntos pode ser usado para um determinado recurso `AWS::Lambda::Function`.

O gerador de IaC não pode determinar qual conjunto de propriedades somente gravação exclusivas foi usado para criar ou atualizar o recurso. Como resultado, ele inclui somente o primeiro conjunto de propriedades no modelo gerado. As propriedades `Code/ImageUri` e `Code/ZipFile` são omitidas.

Além disso, o gerador de IaC emite os seguintes avisos:

- **MUTUALLY_EXCLUSIVE_PROPERTIES**: avisa que `Code/S3Bucket` e `Code/S3Key` são identificadas como propriedades mutuamente excludentes.
- **UNSUPPORTED_PROPERTIES**: avisa que não há suporte à propriedade `Code/S3ObjectVersion`.

Para incluir recursos `AWS::Lambda::Function` em um modelo gerado, você deve baixar e atualizar o modelo com as propriedades de código corretas.

Resolução

Se você armazenar seu código do Lambda em um bucket do Amazon S3 e não usar a propriedade **S3ObjectVersion**, poderá importar o modelo gerado sem nenhuma modificação. O gerador de IaC solicitará o bucket e a chave do Amazon S3 como parâmetros de modelo durante a operação de importação.

Se você armazenar seu código do Lambda como um repositório do Amazon ECR, poderá atualizar seu modelo usando as seguintes instruções:

1. Baixe o modelo gerado.
2. Remova as propriedades e os parâmetros correspondentes das propriedades Code/S3Bucket e Code/S3Key do modelo gerado.
3. Substitua as propriedades removidas no modelo gerado pela propriedade Code/ImageUri, especificando o URL do repositório do Amazon ECR.
4. Abra o modelo gerado no console do gerador de IaC e escolha o botão Importar modelo editado.

Se você armazenar seu código do Lambda em um arquivo ZIP, poderá atualizar seu modelo usando as seguintes instruções:

1. Baixe o modelo gerado.
2. Remova as propriedades e os parâmetros correspondentes das propriedades Code/S3Bucket e Code/S3Key do modelo gerado.
3. Substitua as propriedades removidas no modelo gerado pela propriedade Code/ZipFile.
4. Abra o modelo gerado no console do gerador de IaC e escolha o botão Importar modelo editado.

Se você não tiver uma cópia do seu código do Lambda, poderá atualizar seu modelo usando as seguintes instruções:

1. Use a ação da API [GetFunction](#) do AWS Lambda (por exemplo, usando o comando [aws lambda get-function](#) da AWS CLI).
2. Na resposta, o parâmetro RepositoryType será S3 se o código estiver em um bucket do Amazon S3 ou ECR se o código estiver em um repositório do Amazon ECR.
3. Na resposta, o parâmetro Location contém um URL pré-assinado que pode ser usado para baixar o pacote de implantação por 10 minutos. Baixe o código.
4. Carregue o código para um bucket do Amazon S3.
5. Execute uma operação de importação com o modelo gerado e forneça o nome do bucket e a chave como valores de parâmetros.

Obter valores armazenados em outros serviços usando referências dinâmicas

As referências dinâmicas fornecem uma maneira conveniente de especificar valores externos armazenados e gerenciados em outros serviços e dissociar informações confidenciais de seus modelos de infraestrutura como código. O CloudFormation recupera o valor da referência especificada quando necessário durante operações de pilha e conjunto de alterações.

Com as referências dinâmicas, é possível:

- Usar strings seguras: para dados confidenciais, sempre use parâmetros de strings seguros no AWS Systems Manager Parameter Store ou segredos no AWS Secrets Manager para garantir que seus dados sejam criptografados em repouso.
- Limitar o acesso: restrinja o acesso aos parâmetros do Parameter Store ou aos segredos do Secrets Manager somente a entidades principais e perfis autorizados.
- Faça a rotação das credenciais: faça regularmente a rotação dos seus dados confidenciais armazenados no Parameter Store ou no Secrets Manager para manter um alto nível de segurança.
- Automatize a rotação: use os recursos de rotação automática do Secrets Manager para atualizar e distribuir periodicamente seus dados confidenciais em seus ambientes e aplicações.

Considerações gerais

Antes de especificar referências dinâmicas em seus modelos do CloudFormation, leve em conta as seguintes considerações gerais:

- Evite referências dinâmicas, ou dados confidenciais, nas propriedades do recurso que fazem parte do identificador primário de um recurso. O CloudFormation pode usar o valor em texto simples real no identificador do recurso primário, o que pode ser um risco à segurança. Esse ID de recurso pode aparecer em quaisquer saídas ou destinos derivados.

Para determinar quais propriedades de recursos compreendem o identificador primário de um tipo de recurso, consulte a documentação de referência de recursos desse recurso na [Referência de tipos de propriedades e recursos da AWS](#). Na seção Return values (Valores de retorno), o valor de retorno da função Ref representa as propriedades do recurso que compõem o identificador primário do tipo de recurso.

- Você pode incluir até 60 referências dinâmicas em um modelo de pilha.

- Se você estiver usando transformações (como `AWS::Include` ou `AWS::Serverless`), o CloudFormation não resolverá referências dinâmicas antes de aplicar a transformação. Em vez disso, ele passa a string literal da referência dinâmica para a transformação e resolve as referências quando você executa o conjunto de alterações usando esse modelo.
- Referências dinâmicas não podem ser usadas para valores seguros (como os armazenados no Parameter Store ou no Secrets Manager) em recursos personalizados.
- Também não há suporte para referências dinâmicas em metadados `AWS::CloudFormation::Init` e propriedades `UserData` do Amazon EC2.
- Não crie uma referência dinâmica que termine com uma barra invertida (`\`). O CloudFormation não consegue resolver essas referências, o que fará com que as operações de pilha falhem.

Os tópicos a seguir fornecem informações e outras considerações sobre o uso de referências dinâmicas.

Tópicos

- [Obter um valor em texto simples do Systems Manager Parameter Store](#)
- [Obter um valor de string segura do Systems Manager Parameter Store](#)
- [Obtenha um segredo ou valor secreto do Secrets Manager](#)

Obter um valor em texto simples do Systems Manager Parameter Store

Ao criar um modelo do CloudFormation, talvez você queira usar valores de texto simples armazenados no Parameter Store. O Parameter Store é um recurso do AWS Systems Manager. Para ver uma introdução ao Parameter Store, consulte [AWS Systems Manager Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Para usar um valor de texto simples do Parameter Store em seu modelo, você usa uma referência dinâmica `ssm`. Essa referência permite acessar valores de parâmetros do tipo `String` ou `StringList` no Parameter Store.

Para verificar qual versão de uma referência dinâmica `ssm` será usada em uma operação de pilha, crie um conjunto de alterações para a operação de pilha. Em seguida, revise o modelo processado na guia Modelo. Para obter mais informações, consulte [Criar um conjunto de alterações para uma pilha do CloudFormation](#).

Ao usar referências dinâmicas do ssm, há algumas considerações importantes que você deve ter em mente:

- O CloudFormation não oferece suporte à detecção de desvio em referências dinâmicas. Para referências dinâmicas ssm em que você não especificou a versão do parâmetro, recomendamos que, se você atualizar a versão do parâmetro no Systems Manager, também execute uma operação de atualização de pilha em todas as pilhas que incluam a referência dinâmica ssm a fim de obter a versão mais recente do parâmetro.
- Para usar uma referência dinâmica ssm na seção `Parameters` do modelo do CloudFormation, é necessário incluir um número de versão. O CloudFormation não permite que você faça referência a um valor do Parameter Store sem um número de versão nessa seção. Como alternativa, é possível definir seu parâmetro como um tipo de parâmetro do Systems Manager em seu modelo. Ao fazer isso, você pode especificar uma chave de parâmetros do Systems Manager como o valor padrão para seu parâmetro. Em seguida, o CloudFormation recuperará a versão mais recente do valor do parâmetro no Parameter Store sem que você precise especificar um número de versão. Isso pode tornar seus modelos mais simples e fáceis de manter. Para obter mais informações, consulte [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#).
- Para recursos personalizados, o CloudFormation resolve referências dinâmicas ssm antes de enviar a solicitação ao recurso personalizado.
- O CloudFormation não oferece suporte ao uso de referências dinâmicas para fazer referência a um parâmetro compartilhado de outra Conta da AWS.
- O CloudFormation não oferece suporte ao uso de rótulos de parâmetros do Systems Manager em referências dinâmicas.

Permissões

Para especificar um parâmetro armazenado no Systems Manager Parameter Store, é necessário ter permissão para chamar [GetParameters](#) para o parâmetro especificado. Para saber como criar políticas do IAM que fornecem acesso a parâmetros específicos do Systems Manager, consulte [Restringir o acesso a parâmetros do Systems Manager usando políticas do IAM](#) no Guia do usuário do AWS Systems Manager.

Padrão de referência

Para fazer referência a um valor de texto simples armazenado no Parameter Store do Systems Manager em seu modelo do CloudFormation, use este padrão de referência ssm.

```
{{resolve:ssm:parameter-name:version}}
```

Sua referência deve aderir ao seguinte padrão de expressão regular para `parameter-name` e `version`:

```
{{resolve:ssm:[a-zA-Z0-9_.\-/+](?:\d+)?}}
```

`parameter-name`

O nome do parâmetro no Parameter Store. O nome do parâmetro faz distinção entre maiúsculas e minúsculas.

Obrigatório.

`version`

Um número inteiro que especifica a versão do parâmetro a ser usada. Se você não especificar a versão exata, o CloudFormation usará a versão mais recente do parâmetro sempre que você criar ou atualizar a pilha. Para obter mais informações, consulte [Trabalhar com versões de parâmetros](#) no Guia do usuário do AWS Systems Manager

Opcional.

Exemplos

Tópicos

- [Parâmetro do ID da AMI pública](#)
- [Parâmetro do ID da AMI personalizado](#)

Parâmetro do ID da AMI pública

O exemplo a seguir cria uma instância do EC2 que faz referência a um parâmetro da AMI pública. A referência dinâmica recupera o ID da AMI do Amazon Linux 2023 mais recente do parâmetro público. Para obter mais informações sobre parâmetros públicos, consulte [Descobrir parâmetros públicos no Parameter Store](#) no Guia do usuário do AWS Systems Manager.

JSON

```
{  
  "Resources": {  
    "MyInstance": {
```

```

        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
            "InstanceType": "t2.micro"
        }
    }
}

```

YAML

```

Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}'
      InstanceType: t2.micro

```

Parâmetro do ID da AMI personalizado

O exemplo a seguir cria um modelo de execução do EC2 que faz referência a um ID de AMI personalizada armazenada no Parameter Store. A referência dinâmica recupera o ID da AMI para a versão **2** do parâmetro *golden-ami* no momento em que uma instância é executada a partir do modelo de execução.

JSON

```

{
  "Resources": {
    "MyLaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Properties": {
        "LaunchTemplateName": {
          "Fn::Sub": "${AWS::StackName}-launch-template"
        },
        "LaunchTemplateData": {
          "ImageId": "{{resolve:ssm:golden-ami:2}}",
          "InstanceType": "t2.micro"
        }
      }
    }
  }
}

```



```
}  
}
```

YAML

```
Resources:  
  MyLaunchTemplate:  
    Type: AWS::EC2::LaunchTemplate  
    Properties:  
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template  
      LaunchTemplateData:  
        ImageId: '{{resolve:ssm:golden-ami:2}}'  
        InstanceType: t2.micro
```

Obter um valor de string segura do Systems Manager Parameter Store

No CloudFormation, é possível usar dados confidenciais, como senhas ou chaves de licença, sem expô-los diretamente em seus modelos, armazenando os dados confidenciais como uma "string segura" no AWS Systems Manager Parameter Store. Para ver uma introdução ao Parameter Store, consulte [AWS Systems Manager Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Para usar uma string segura do Parameter Store em seu modelo, use uma referência dinâmico `ssm-secure`. O CloudFormation nunca armazena o valor real do parâmetro. Em vez disso, ele armazena apenas a referência dinâmica literal, que contém o nome do parâmetro de texto simples da string segura.

Durante a criação ou as atualizações da pilha, o CloudFormation acessa o valor da string segura conforme necessário, sem expor o valor real. As strings seguras só podem ser usadas para propriedades de recursos que oferecem suporte ao padrão de referência dinâmica `ssm-secure`. Para obter mais informações, consulte [Recursos que oferecem suporte a padrões de parâmetros dinâmicos para strings seguras](#).

O CloudFormation não retorna o valor real do parâmetro para strings seguras em nenhuma chamada de API. Ele retorna apenas a referência dinâmica literal. Ao comparar alterações usando conjuntos de alterações, o CloudFormation compara somente a string de referência dinâmica literal. Ele não resolve nem compara os valores reais das strings seguras.

Ao usar referências dinâmicas do `ssm-secure`, há algumas considerações importantes que você deve ter em mente:

- O CloudFormation não pode acessar valores do Parameter Store de outras Contas da AWS.

- O CloudFormation não oferece suporte ao uso de rótulos de parâmetros do Systems Manager ou de parâmetros públicos em referências dinâmicas.
- O Systems Manager não oferece suporte a strings seguras nas regiões `cn-north-1` e `cn-northwest-1`.
- No momento, as referências dinâmicas para valores seguros, como `ssm-secure`, não são compatíveis com recursos personalizados.
- Se o CloudFormation precisar reverter uma atualização de pilha e a versão anteriormente especificada de um parâmetro de string segura não estiver mais disponível, essa operação de reversão falhará. Nesses casos, você tem duas opções:
 - Use `CONTINUE_UPDATE_ROLLBACK` para ignorar o recurso.
 - Recrie o parâmetro de string segura no Repositório de parâmetros do Systems Manager e atualize-o até que a versão do parâmetro atinja a versão usada no modelo. Em seguida, use `CONTINUE_UPDATE_ROLLBACK` sem ignorar o recurso.

Recursos que oferecem suporte a padrões de parâmetros dinâmicos para strings seguras

Os recursos que oferecem suporte ao padrão de referência dinâmica `ssm-secure` incluem:

Recurso	Tipo de propriedade	Propriedades
AWS::DirectoryService::MicrosoftAD		Password
AWS::DirectoryService::SimpleAD		Password
AWS::ElastiCache::ReplicationGroup		AuthToken
AWS::IAM::User	LoginProfile	Password
AWS::KinesisFirehose::DeliveryStream	RedshiftDestinationConfiguration	Password
AWS::OpsWorks::App	Fonte	Password

Recurso	Tipo de propriedade	Propriedades
AWS::OpsWorks::Stack	CustomCookbooksSource	Password
AWS::OpsWorks::Stack	RdsDbInstances	DbPassword
AWS::RDS::DBCluster		MasterUserPassword
AWS::RDS::DBInstance		MasterUserPassword
AWS::Redshift::Cluster		MasterUserPassword

Padrão de referência

Para fazer referência a um valor string segura do Parameter Store do Systems Manager em seu modelo do CloudFormation, use este padrão de referência `ssm-secure`.

```
{{resolve:ssm-secure:parameter-name:version}}
```

Sua referência deve aderir ao seguinte padrão de expressão regular para `parameter-name` e `version`:

```
{{resolve:ssm-secure:[a-zA-Z0-9_.\-/]+(:\d+)?}}
```

`parameter-name`

O nome do parâmetro no Parameter Store. O nome do parâmetro faz distinção entre maiúsculas e minúsculas.

Obrigatório.

`version`

Um número inteiro que especifica a versão do parâmetro a ser usada. Se você não especificar a versão exata, o CloudFormation usará a versão mais recente do parâmetro sempre que você criar ou atualizar a pilha. Para obter mais informações, consulte [Trabalhar com versões de parâmetros](#) no Guia do usuário do AWS Systems Manager

Opcional.

Exemplo

O exemplo a seguir usa uma referência dinâmica `ssm-secure` para definir a senha de um usuário do IAM como uma string segura armazenada no Parameter Store. Conforme especificado, CloudFormation usará a versão **10** do parâmetro ***IAMUserPassword*** para operações de pilha e conjunto de alterações.

JSON

```
"MyIAMUser": {
  "Type": "AWS::IAM::User",
  "Properties": {
    "UserName": "MyUserName",
    "LoginProfile": {
      "Password": "{{resolve:ssm-secure:IAMUserPassword:10}}"
    }
  }
}
```

YAML

```
MyIAMUser:
  Type: AWS::IAM::User
  Properties:
    UserName: 'MyUserName'
    LoginProfile:
      Password: '{{resolve:ssm-secure:IAMUserPassword:10}}'
```

Obtenha um segredo ou valor secreto do Secrets Manager

O Secrets Manager é um serviço que permite armazenar e gerenciar segredos com segurança, como credenciais de banco de dados, senhas e chaves de API de terceiros. Usando o Secrets Manager, você pode armazenar e controlar o acesso a esses segredos centralmente, para que você possa substituir credenciais codificadas no seu código (incluindo senhas) por uma chamada de API para o Secrets Manager recuperar o segredo de forma programática. Para obter mais informações, consulte [O que é o AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager.

Use referências dinâmicas `secretsmanager` para usar todos os segredos ou valores secretos armazenados no Secrets Manager em seus modelos do CloudFormation.

Práticas recomendadas

Siga estas práticas recomendadas ao usar as referências dinâmicas do Secrets Manager em seus modelos do CloudFormation:

- Use referências sem versão para seus modelos do CloudFormation: armazene credenciais no Secrets Manager e use referências dinâmicas sem especificar parâmetros `version-stage` ou `version-id` para suportar fluxos de trabalho adequados de rodízio de segredos.
- Aproveite o rodízio automático: use o atributo de rodízio automático do Secrets Manager com referências dinâmicas sem versão para gerenciamento de credenciais. Isso garante que suas credenciais sejam atualizadas regularmente sem exigir alterações no modelo. Para ter mais informações, consulte [Alternar segredos do AWS Secrets Manager](#).
- Use referências versionadas com moderação: especifique somente parâmetros explícitos `version-stage` ou `version-id` para cenários específicos, como situações de teste ou reversão.

Considerações

Ao usar referências dinâmicas do `secretsmanager`, existem considerações importantes que devem ser consideradas:

- O CloudFormation não rastreia qual versão de um segredo foi usada em implantações anteriores. Planeje cuidadosamente sua estratégia de gerenciamento de segredo antes de implementar referências dinâmicas. Use referências sem versão quando possível para aproveitar o rodízio automático de segredos. Monitore e valide as atualizações de recursos ao fazer alterações nas configurações de referência dinâmica, como ao fazer a transição de referências dinâmicas não versionadas para referências dinâmicas versionadas e vice-versa.
- Atualizar somente o valor do segredo no Secrets Manager não faz com que o CloudFormation recupere automaticamente o novo valor. O CloudFormation recupera o valor do segredo somente durante a criação de recursos ou atualizações que modificam o recurso que contém a referência dinâmica.

Por exemplo, suponha que seu modelo inclua um recurso [AWS::RDS::DBInstance](#) em que a propriedade `MasterPassword` esteja definida como uma referência dinâmica do Secrets Manager. Depois de criar uma pilha a partir desse modelo, você atualiza o valor do segredo no Secrets Manager. No entanto, a propriedade `MasterPassword` mantém o valor da senha antiga.

Para aplicar o novo valor de segredo, você precisará modificar o recurso

AWS::RDS::DBInstance em seu modelo do CloudFormation e realizar uma atualização da pilha.

Para evitar esse processo manual no futuro, considere usar o Secrets Manager para fazer o rodízio automático do segredo.

- No momento, as referências dinâmicas para valores seguros, como `secretsmanager`, não são compatíveis com recursos personalizados.
- A referência dinâmica `secretsmanager` pode ser usada em todas as propriedades de recursos. Usar a referência dinâmica `secretsmanager` indica que nem o Secrets Manager, nem o CloudFormation deve registrar ou persistir qualquer valor secreto resolvido. No entanto, o valor secreto pode aparecer no serviço cujo recurso está sendo usado. Revise seu uso para evitar o vazamento de dados secretos.

Permissões

Para especificar um segredo armazenado no Secrets Manager, você deve ter permissão para chamar [GetSecretValue](#) para o segredo.

Padrão de referência

Para fazer referência a segredos do Secrets Manager em seu modelo do CloudFormation, use este padrão de referência `secretsmanager`.

```
{{resolve:secretsmanager:secret-id:secret-string:json-key:version-stage:version-id}}
```

`secret-id`

O nome ou ARN completo do segredo.

Para acessar um segredo na sua Conta da AWS, é necessário especificar apenas o nome do segredo. Para acessar um segredo em uma Conta da AWS diferente, especifique o ARN completo do segredo.

Obrigatório.

`secret-string`

O único valor aceito é `SecretString`. O padrão é `SecretString`.

json-key

O nome da chave do par de chave-valor cujo valor você deseja recuperar. Se você não especificar um `json-key`, o CloudFormation recuperará todo o texto do segredo.

Esse segmento não pode incluir o caractere de dois pontos (:).

version-stage

O rótulo de preparação da versão do segredo a ser utilizada. O Secrets Manager usa rótulos de preparação para acompanhar diferentes versões durante o processo de alternância. Se você usar `version-stage`, não especifique `version-id`. Se você não especificar `version-stage` ou `version-id`, o padrão é a versão `AWSCURRENT`.

Esse segmento não pode incluir o caractere de dois pontos (:).

version-id

O identificador exclusivo da versão do segredo a usar. Se você especificar `version-id`, não especifique `version-stage`. Se você não especificar `version-stage` ou `version-id`, o padrão é a versão `AWSCURRENT`.

Esse segmento não pode incluir o caractere de dois pontos (:).

Exemplos

Tópicos

- [Recuperar valores de nome de usuário e senha de um segredo](#)
- [Recuperar todo o SecretString](#)
- [Recuperar um valor de uma versão específica de um segredo](#)
- [Recuperar segredos de outro Conta da AWS](#)

Recuperar valores de nome de usuário e senha de um segredo

O exemplo a seguir [AWS::RDS::DBInstance](#) recupera os valores de nome de usuário e senha armazenados no segredo *MySecret*. Este exemplo mostra o padrão recomendado para referências dinâmicas sem versão, que usa automaticamente a versão `AWSCURRENT` e dá suporte aos fluxos de trabalho de rodízio do Secrets Manager sem exigir alterações no modelo.

JSON

```
{
  "MyRDSInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName": "MyRDSInstance",
      "AllocatedStorage": "20",
      "DBInstanceClass": "db.t2.micro",
      "Engine": "mysql",
      "MasterUsername":
        "{{resolve:secretsmanager:MySecret:SecretString:username}}",
      "MasterUserPassword":
        "{{resolve:secretsmanager:MySecret:SecretString:password}}"
    }
  }
}
```

YAML

```
MyRDSInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName: MyRDSInstance
    AllocatedStorage: '20'
    DBInstanceClass: db.t2.micro
    Engine: mysql
    MasterUsername: '{{resolve:secretsmanager:MySecret:SecretString:username}}'
    MasterUserPassword: '{{resolve:secretsmanager:MySecret:SecretString:password}}'
```

Recuperar todo o SecretString

A referência dinâmica a seguir recupera o SecretString para *MySecret*.

```
{{resolve:secretsmanager:MySecret}}
```

Alternativa:

```
{{resolve:secretsmanager:MySecret:::}}
```


Recuperar um valor de uma versão específica de um segredo

A referência dinâmica a seguir recupera o valor *password* para a versão *AWSPREVIOUS* do *MySecret*.

```
{{resolve:secretsmanager:MySecret:SecretString:password:AWSPREVIOUS}}
```

Recuperar segredos de outro Conta da AWS

A referência dinâmica a seguir recupera o SecretString para *MySecret* que esteja em outra Conta da AWS. Você deve especificar o ARN do segredo completo para acessar segredos em outra Conta da AWS.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret}}
```

A referência dinâmica a seguir recupera o valor *password* para *MySecret* que esteja em outra Conta da AWS. Você deve especificar o ARN do segredo completo para acessar segredos em outra Conta da AWS.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret:SecretString:password}}
```

Obter valores da AWS usando pseudoparâmetros

Os pseudoparâmetros são variáveis integradas que fornecem acesso a informações importantes do ambiente da AWS, como IDs de contas, nomes de regiões e detalhes da pilha, que podem mudar entre implantações ou ambientes.

Você pode usar pseudoparâmetros em vez de valores com codificação rígida para tornar seus modelos mais portáteis e fáceis de serem reutilizados em diferentes regiões e Contas da AWS.

Sintaxe

Você pode referenciar pseudoparâmetros usando a função intrínseca Ref ou a Fn::Sub.

Ref.

A função intrínseca Ref usa a sintaxe geral a seguir. Para obter mais informações, consulte [Ref](#).

JSON

```
{ "Ref" : "AWS::PseudoParameter" }
```

YAML

```
!Ref AWS::PseudoParameter
```

Fn::Sub

A função intrínseca Fn::Sub usa um formato diferente que inclui a sintaxe `${}` em torno do pseudoparâmetro. Para obter mais informações, consulte [Fn::Sub](#).

JSON

```
{ "Fn::Sub" : "${AWS::PseudoParameter}" }
```

YAML

```
!Sub '${AWS::PseudoParameter}
```

Pseudoparâmetros disponíveis

AWS::AccountId

Devolve o ID da Conta da AWS da conta na qual a pilha está sendo criada, como 123456789012.

Esse pseudoparâmetro é comumente usado ao definir perfis, políticas e outras políticas de recursos do IAM que envolvem ARNs específicos da conta.

AWS::NotificationARNs

Retorna a lista de nomes de recursos da Amazon (ARNs) dos tópicos do Amazon SNS que recebem as notificações de eventos da pilha. Você pode especificar esses ARNs usando a opção `--notification-arns` na AWS CLI ou usando o console enquanto cria ou atualiza sua pilha.

Ao contrário de outros pseudoparâmetros que retornam um valor único, `AWS::NotificationARNs` retorna uma lista de ARNs. Para acessar um ARN específico na lista, use a função intrínseca `Fn::Select`. Para obter mais informações, consulte [Fn::Select](#).

AWS::NoValue

Remove a propriedade do recurso correspondente quando especificada como um valor de retorno na função intrínseca `Fn::If`. Para obter mais informações, consulte [Fn::If](#).

Esse pseudoparámetro é particularmente útil para criar propriedades de recursos condicionais que só devem ser incluídas em determinadas condições.

AWS::Partition

Retorna a partição onde o recurso se encontra. Para Regiões da AWS padrão, a partição é `aws`. Para recursos em outras partições, a partição é `aws-nomedapartição`. Por exemplo, a partição dos recursos na região da China (Pequim e Ningxia) é `aws-cn`, e a partição para recursos na região AWS GovCloud (EUA-Oeste) é `aws-us-gov`.

A partição faz parte do ARN dos recursos. Usar `AWS::Partition` garante que seus modelos funcionem corretamente em diferentes partições da AWS.

AWS::Region

Retorna uma sequência que representa a região na qual o recurso abrangente está sendo criado, como `us-west-2`.

Este é um dos pseudoparámetros mais usados, pois permite que os modelos se adaptem a diferentes Regiões da AWS sem modificação.

AWS::StackId

Retorna o ID (ARN) da pilha, como `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`.

AWS::StackName

Retorna o nome da pilha, como `teststack`.

O nome da pilha é normalmente usado para criar nomes de recursos exclusivos que são facilmente identificáveis como pertencentes a uma pilha específica.

AWS::URLSuffix

Retorna o sufixo para o domínio da AWS na Região da AWS onde a pilha está implantada. O sufixo é normalmente `amazonaws.com`, mas para a região China (Pequim), o sufixo é `amazonaws.com.cn`.

Esse parâmetro é particularmente útil ao criar URLs para endpoints de serviços da AWS.

Exemplos

Tópicos

- [Uso básico](#)
- [Usar o AWS::NotificationARNs](#)
- [Propriedades condicionais com AWS::NoValue](#)

Uso básico

Os exemplos a seguir criam dois recursos: um tópico do Amazon SNS e um alarme do CloudWatch que envia notificações para esse tópico. Eles usam `AWS::StackName`, `AWS::Region` e `AWS::AccountId` para inserir dinamicamente o nome da pilha, a Região da AWS atual e o ID da conta nos ARNs, nas descrições e nos nomes dos recursos.

JSON

```
{
  "Resources": {
    "MyNotificationTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "DisplayName": { "Fn::Sub": "Notifications for ${AWS::StackName}" }
      }
    },
    "CPUALarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": { "Fn::Sub": "Alarm for high CPU in ${AWS::Region}" },
        "AlarmName": { "Fn::Sub": "${AWS::StackName}-HighCPUALarm" },
        "MetricName": "CPUUtilization",
```

```

        "Namespace": "AWS/EC2",
        "Statistic": "Average",
        "Period": 300,
        "EvaluationPeriods": 1,
        "Threshold": 80,
        "ComparisonOperator": "GreaterThanThreshold",
        "AlarmActions": [{ "Fn::Sub": "arn:aws:sns:${AWS::Region}:
${AWS::AccountId}:${MyNotificationTopic}" }]
    }
}
}
}

```

YAML

```

Resources:
  MyNotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: !Sub Notifications for ${AWS::StackName}
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: !Sub Alarm for high CPU in ${AWS::Region}
      AlarmName: !Sub ${AWS::StackName}-HighCPUAlarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 300
      EvaluationPeriods: 1
      Threshold: 80
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - !Sub arn:aws:sns:${AWS::Region}:${AWS::AccountId}:${MyNotificationTopic}

```

Usar o AWS::NotificationARNs

Os exemplos a seguir configuram um grupo do Auto Scaling para enviar notificações de eventos de inicialização de instâncias e erros de inicialização. A configuração usa o pseudoparámetro `AWS::NotificationARNs`, que fornece uma lista de ARNs de tópicos do Amazon SNS que foram especificados durante a criação da pilha. A função `Fn::Select` escolhe o primeiro ARN dessa lista.

JSON

```
"myASG": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Properties": {
    "LaunchTemplate": {
      "LaunchTemplateId": { "Ref": "myLaunchTemplate" },
      "Version": { "Fn::GetAtt": [ "myLaunchTemplate", "LatestVersionNumber" ] }
    },
    "MaxSize": "1",
    "MinSize": "1",
    "VPCZoneIdentifier": [
      "subnetIdAz1",
      "subnetIdAz2",
      "subnetIdAz3"
    ],
    "NotificationConfigurations" : [{
      "TopicARN" : { "Fn::Select" : [ "0", { "Ref" : "AWS::NotificationARNs" } ] },
      "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH",
"autoscaling:EC2_INSTANCE_LAUNCH_ERROR" ]
    }]
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MinSize: '1'
    MaxSize: '1'
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    NotificationConfigurations:
      - TopicARN:
          Fn::Select:
            - '0'
          Ref: AWS::NotificationARNs
```

```
NotificationTypes:
- autoscaling:EC2_INSTANCE_LAUNCH
- autoscaling:EC2_INSTANCE_LAUNCH_ERROR
```

Propriedades condicionais com AWS::NoValue

Os exemplos a seguir criam uma instância de banco de dados do Amazon RDS que usará um snapshot apenas se o ID do snapshot for fornecido. Se a condição UseDBSnapshot for avaliada como verdadeira, o CloudFormation usará o valor do parâmetro DBSnapshotName para a propriedade DBSnapshotIdentifier. Se a condição for avaliada como falsa, o CloudFormation removerá a propriedade DBSnapshotIdentifier.

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
        "UseDBSnapshot",
        {"Ref" : "DBSnapshotName"},
        {"Ref" : "AWS::NoValue"}
      ]
    }
  }
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
```

```
Engine: MySQL
EngineVersion: '5.5'
MasterUsername:
  Ref: DBUser
MasterUserPassword:
  Ref: DBPassword
DBParameterGroupName:
  Ref: MyRDSParamGroup
DBSnapshotIdentifier:
  Fn::If:
    - UseDBSnapshot
    - Ref: DBSnapshotName
    - Ref: AWS::NoValue
```

Obter resultados exportados de uma pilha do CloudFormation implantada

Quando você tem várias pilhas na mesma região e Conta da AWS, talvez queira compartilhar informações entre elas. Útil quando uma pilha precisa usar recursos criados por outra pilha.

Por exemplo, você pode ter uma pilha que cria recursos de rede, como sub-redes e grupos de segurança, para servidores Web. Outras pilhas que criam os servidores Web reais podem então usar os recursos de rede criados pela primeira pilha. Você não precisa codificar IDs de recurso no modelo da pilha ou passar IDs como parâmetros de entrada.

Para compartilhar informações entre pilhas, exporte valores de saída de uma pilha e importe-os para outra pilha. Como isso funciona:

1. No modelo da primeira pilha (por exemplo, a pilha de rede), você define determinados valores para exportação usando o campo `Export` na seção `Outputs`. Para obter mais informações, consulte [Sintaxe de Outputs de modelo do CloudFormation](#).
2. Quando você criar ou atualizar essa pilha, o CloudFormation exportará os valores de saída, disponibilizando-os para outras pilhas na mesma região e Conta da AWS.
3. No modelo da outra pilha, você usa a função [Fn::ImportValue](#) para importar os valores exportados da primeira pilha.
4. Quando você criar ou atualizar a segunda pilha (por exemplo, a pilha do servidor web), o CloudFormation recuperará automaticamente os valores exportados da primeira pilha e os usará.

Para um passo a passo e modelos de amostra, consulte [Consultar saídas de recurso em outra pilha do CloudFormation](#).

Exportar valores de saída da pilha vs usar pilhas aninhadas

Pilha aninhada é uma pilha que você cria dentro de outra pilha usando o recurso

`AWS::CloudFormation::Stack`. Com pilhas aninhadas, você implanta e gerencia todos os recursos em uma única pilha. É possível usar saídas de uma pilha no grupo de pilhas aninhadas como entradas para outra pilha no grupo. Isso é diferente de exportar valores.

Caso você queira isolar o compartilhamento de informações dentro de um grupo de pilhas aninhadas, sugerimos usar pilhas aninhadas. Para compartilhar informações com outras pilhas (e não apenas dentro do grupo de pilhas aninhadas), exporte os valores. Por exemplo, você pode criar uma única pilha com uma sub-rede e exportar o ID. Outras pilhas podem usar essa sub-rede importando o ID. Não é necessário que cada pilha crie a própria sub-rede. Como as pilhas estão importando o ID de sub-rede, não é possível alterar nem excluí-lo.

Para saber mais sobre pilhas aninhadas, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).

Considerações

As restrições a seguir se aplicam a referências de pilha cruzada.

- Para cada Conta da AWS, os nomes de `Export` devem ser exclusivos em uma região.
- Não é possível criar referências entre pilhas nas diversas regiões. Você pode usar a função intrínseca `Fn::ImportValue` para importar somente valores que foram exportados na mesma região.
- Para saídas, o valor da propriedade `Name` de uma `Export` não pode utilizar funções `Ref` ou `GetAtt` que dependem de um recurso.

Da mesma forma, a função `ImportValue` não pode incluir funções `Ref` ou `GetAtt` que dependem de um recurso.

- Depois que outra pilha importar um valor de saída, não será possível excluir a pilha que está exportando o valor de saída ou modificar o valor de saída exportado. Todas as importações deverão ser removidas para que você possa excluir a pilha de exportação ou modificar o valor de saída.

Listar valores de saída exportados

Se você precisar consultar os valores de saída exportados de suas pilhas, use um dos seguintes métodos:

Para listar valores de saída exportados (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação esquerdo, escolha Exportações.

Para listar valores de saída exportados (AWS CLI)

Use o seguinte comando [list-exports](#): Substitua *us-east-1* por sua Região da AWS.

```
aws cloudformation list-exports --region us-east-1
```

O seguinte é um exemplo de saída.

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-a",
      "Value": "subnet-07b410xmplddcfa03"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-b",
      "Value": "subnet-075ed3xmplebd2fb1"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-vpcid",
      "Value": "vpc-011d7xmpl100e9841"
    }
  ]
}
```

O CloudFormation mostra os nomes e os valores das saídas exportadas da região atual e a pilha da qual elas foram exportadas. Para usar um valor de saída exportado em outro modelo de pilha, você pode fazer referência a ele usando o nome de exportação e a função `Fn::ImportValue`.

Listar pilhas que importam um valor de saída exportado

Para excluir ou alterar os valores de saída exportados, primeiro é necessário descobrir quais pilhas os estão importando.

Para ver as pilhas que importam um valor de saída exportado, use um dos seguintes métodos:

Para listar pilhas que importam um valor de saída exportado (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação esquerdo, escolha Exportações.
3. Para verificar quais pilhas importam um determinado valor de exportação, escolha Export Name (Nome da exportação) para esse valor de exportação. O CloudFormation mostra a página de detalhes da exportação, que lista todas as pilhas que estão importando esse valor.

Para listar pilhas que importam um valor de saída exportado (AWS CLI)

Use o comando [list-imports](#). Substitua *us-east-1* por sua Região da AWS e *private-vpc-vpcid* pelo nome do valor de saída exportado.

```
aws cloudformation list-imports --region us-east-1 \  
  --export-name private-vpc-vpcid
```

O CloudFormation retorna uma lista de pilhas que estão importando o valor.

```
{  
  "Imports": [  
    "my-app-stack"  
  ]  
}
```

Assim que souber quais pilhas estão importando um valor exportado específico, você precisará modificar essas pilhas para remover as funções `Fn::ImportValue` que fazem referência aos valores de saída. Você deve remover todas as importações que fazem referência a valores de saída exportados para poder excluir ou modificar os valores de saída exportados.

Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation

Ao criar seu modelo, você pode criar parâmetros que exijam que os usuários insiram identificadores de recursos da AWS existentes ou parâmetros do Systems Manager usando tipos de parâmetros especializados fornecidos pelo CloudFormation.

Tópicos

- [Visão geral](#)
- [Exemplo](#)
- [Considerações](#)
- [Tipos de parâmetros específicos da AWS compatíveis](#)
- [Tipos de parâmetros compatíveis do Systems Manager](#)
- [Tipos de parâmetros incompatíveis do Systems Manager](#)

Visão geral

No CloudFormation, é possível usar parâmetros para personalizar suas pilhas ao fornecer valores de entrada durante a criação ou atualização da pilha. Esse recurso torna seus modelos reutilizáveis e flexíveis em diferentes cenários.

Os parâmetros são definidos na seção `Parameters` de um modelo do CloudFormation. Cada parâmetro tem um nome e um tipo e pode ser configurado com atributos adicionais, como um valor padrão e valores permitidos. Para obter mais informações, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

O tipo de parâmetro determina o tipo de valor de entrada que o parâmetro pode aceitar. Por exemplo, `Number` aceita somente valores numéricos, enquanto `String` aceita entrada de texto.

O CloudFormation oferece vários tipos de parâmetros adicionais que você pode usar em seus modelos para referenciar recursos da AWS existentes e parâmetros do Systems Manager.

Esses tipos de parâmetro se encaixam em duas categorias:

- Tipos de parâmetros específicos da AWS: o CloudFormation fornece um conjunto de tipos de parâmetros que ajudam a capturar valores inválidos ao criar ou atualizar uma pilha. Ao usar esses

tipos de parâmetros, qualquer pessoa que usar seu modelo deverá especificar valores válidos da região e da Conta da AWS em que está criando a pilha.

Se eles usarem o Console de gerenciamento da AWS, o CloudFormation fornecerá uma lista pré-preenchida dos valores existentes de sua conta e região. Dessa forma, o usuário não precisa se lembrar e digitar corretamente um nome ou ID específico. Em vez disso, ele apenas seleciona valores em uma lista suspensa. Em alguns casos, é possível até mesmo pesquisar valores por ID, nome ou valor da tag Name.

- Tipos de parâmetros do Systems Manager: o CloudFormation também oferece tipos de parâmetros que correspondem a parâmetros existentes no Systems Manager Parameter Store. Quando você usa esses tipos de parâmetros, qualquer pessoa que usar seu modelo deverá especificar uma chave do repositório de parâmetros como o valor do tipo de parâmetro do Systems Manager. Em seguida, o CloudFormation recupera o valor mais recente no Parameter Store a ser usado em sua pilha. Isso pode ser útil quando você precisa atualizar com frequência os recursos da aplicação com novos valores de propriedades, por exemplo, IDs de imagem de máquina da Amazon (AMI). Consulte informações sobre o Parameter Store em [Systems Manager Parameter Store](#).

Após os parâmetros serem definidos na seção `Parameters`, você poderá referenciar os valores dos parâmetros em todo o seu modelo do CloudFormation usando a função `Ref`.

Exemplo

O exemplo mostra um modelo que usa os tipos de parâmetro a seguir.

- `AWS::EC2::VPC::Id`
- `AWS::EC2::Subnet::Id`
- `AWS::EC2::KeyPair::KeyName`
- `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`

Para criar uma pilha com base nesse modelo, você deve especificar um ID de VPC, um ID de sub-rede e um nome de par de chaves da sua conta. Você também pode especificar uma chave existente do Parameter Store que faça referência ao ID de AMI desejado ou manter o valor padrão de `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`. Esse parâmetro público é um alias para o ID de AMI regional para a AMI mais recente do Amazon Linux 2. Para obter mais informações sobre parâmetros públicos, consulte [Descobrir parâmetros públicos no Parameter Store](#) no Guia do usuário do AWS Systems Manager.

JSON

```
{
  "Parameters": {
    "VpcId": {
      "Description": "ID of an existing Virtual Private Cloud (VPC).",
      "Type": "AWS::EC2::VPC::Id"
    },
    "PublicSubnetId": {
      "Description": "ID of an existing public subnet within the specified VPC.",
      "Type": "AWS::EC2::Subnet::Id"
    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to enable SSH access to the instance.",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "AMIId": {
      "Description": "Name of a Parameter Store parameter that stores the ID of the Amazon Machine Image (AMI).",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    }
  },
  "Resources": {
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "VpcId": { "Ref": "VpcId" },
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    },
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
```

```

        "KeyName": { "Ref": "KeyName" },
        "ImageId": { "Ref": "AMIId" },
        "NetworkInterfaces": [
            {
                "AssociatePublicIpAddress": "true",
                "DeviceIndex": "0",
                "SubnetId": { "Ref": "PublicSubnetId" },
                "GroupSet": [{ "Ref": "InstanceSecurityGroup" }]
            }
        ]
    },
    "Outputs": {
        "InstanceId": {
            "Value": { "Ref": "Ec2Instance" }
        }
    }
}

```

YAML

```

Parameters:
  VpcId:
    Description: ID of an existing Virtual Private Cloud (VPC).
    Type: AWS::EC2::VPC::Id
  PublicSubnetId:
    Description: ID of an existing public subnet within the specified VPC.
    Type: AWS::EC2::Subnet::Id
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance.
    Type: AWS::EC2::KeyPair::KeyName
  AMIId:
    Description: Name of a Parameter Store parameter that stores the ID of the Amazon
    Machine Image (AMI).
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
Resources:
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access via port 22
      VpcId: !Ref VpcId

```

```

    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      KeyName: !Ref KeyName
      ImageId: !Ref AMIID
      NetworkInterfaces:
        - AssociatePublicIpAddress: "true"
          DeviceIndex: "0"
          SubnetId: !Ref PublicSubnetId
          GroupSet:
            - !Ref InstanceSecurityGroup
  Outputs:
    InstanceId:
      Value: !Ref Ec2Instance

```

Comando AWS CLI para criar a pilha

O comando [create-stack](#) a seguir cria uma pilha com base no modelo de exemplo.

```

aws cloudformation create-stack --stack-name MyStack \
  --template-body file://sampletemplate.json \
  --parameters \
    ParameterKey="VpcId",ParameterValue="vpc-a123baa3" \
    ParameterKey="PublicSubnetId",ParameterValue="subnet-123a351e" \
    ParameterKey="KeyName",ParameterValue="MyKeyName" \
    ParameterKey="AMIId",ParameterValue="MyParameterKey"

```

Para usar um tipo de parâmetro que aceite uma lista de strings, como `List<AWS::EC2::Subnet::Id>`, é necessário escapar as vírgulas dentro de `ParameterValue` com uma barra invertida dupla, conforme mostrado no exemplo a seguir.

```

--parameters ParameterKey="SubnetIDs",ParameterValue="subnet-5ea0c127\\, subnet-6194ea3b\\, subnet-c87f2be0"

```


Considerações

É altamente recomendável usar referências dinâmicas para restringir o acesso a definições de configuração confidenciais, como credenciais de terceiros. Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

Se desejar permitir que os usuários do modelo especifiquem valores de Contas da AWS diferentes, não use tipos de parâmetros específicos da AWS. Em vez disso, defina parâmetros do tipo `String` ou `CommaDelimitedList`.

Há algumas coisas que você deve ter em mente com os tipos de parâmetros do Systems Manager:

- É possível ver os valores de parâmetros resolvidos na guia Parâmetros da pilha no console ou executando [describe-stacks](#) ou [describe-change-set](#). Lembre-se: esses valores são definidos quando a pilha é criada ou atualizada e, portanto, podem ser diferentes dos últimos valores no Parameter Store.
- Para atualizações de pilha, ao usar a opção Usar valor existente (ou define `UsePreviousValue` como verdadeiro), isso significa que você deseja continuar usando a mesma chave do Parameter Store, não seu valor. O CloudFormation sempre recupera o valor mais recente.
- Se você especificar quaisquer valores permitidos ou outras restrições, o CloudFormation os validará com base nas chaves de parâmetros que você especificar, mas não em seus valores. Você deve validar os valores no próprio Parameter Store.
- Quando você cria ou atualiza pilhas e cria conjuntos de alterações, o CloudFormation usa quaisquer valores existentes no Parameter Store no momento em que a operação é realizada. Se não houver um parâmetro especificado no Parameter Store, sob a Conta da AWS do chamador, o CloudFormation retornará um erro de validação.
- Quando você executa um conjunto de alterações, o CloudFormation usa os valores especificados no conjunto de alterações. Você deve analisar esses valores antes de executar o conjunto de alterações, pois eles podem ser alterados no Armazenamento de parâmetros entre o momento em que você cria o conjunto de alterações e momento em que o executa.
- Para parâmetros do Parameter Store armazenados na mesma Conta da AWS, você deverá fornecer o nome do parâmetro. Para parâmetros do Parameter Store compartilhados por outra Conta da AWS, será necessário fornecer o ARN completo do parâmetro.

Tipos de parâmetros específicos da AWS compatíveis

O CloudFormation é compatível com os seguintes tipos específicos da AWS:

`AWS::EC2::AvailabilityZone::Name`

Uma Zona de disponibilidade, como `us-west-2a`.

`AWS::EC2::Image::Id`

Um ID de imagem do Amazon EC2, como `ami-0ff8a91507f77f867`. Observe que o console do CloudFormation não mostra uma lista suspensa de valores para esse tipo de parâmetro.

`AWS::EC2::Instance::Id`

Um ID de instância do Amazon EC2, como `i-1e731a32`.

`AWS::EC2::KeyPair::KeyName`

Um nome do par de chaves do Amazon EC2.

`AWS::EC2::SecurityGroup::GroupName`

Um nome de grupo de segurança da VPC padrão, como `my-sg-abc`.

`AWS::EC2::SecurityGroup::Id`

Um ID de security group, como `sg-a123fd85`.

`AWS::EC2::Subnet::Id`

Um ID de sub-rede, como `subnet-123a351e`.

`AWS::EC2::Volume::Id`

Um ID de volume do Amazon EBS, como `vol-3cdd3f56`.

`AWS::EC2::VPC::Id`

Um ID de VPC, como `vpc-a123baa3`.

`AWS::Route53::HostedZone::Id`

Um ID de zona hospedada do Amazon Route 53, como `Z23YXV40VPL04A`.

`List<AWS::EC2::AvailabilityZone::Name>`

Uma matriz de Zonas de disponibilidade de uma região, como `us-west-2a`, `us-west-2b`.

`List<AWS::EC2::Image::Id>`

Uma matriz de IDs de imagens do Amazon EC2, como `ami-0ff8a91507f77f867`, `ami-0a584ac55a7631c0c`. Observe que o console do CloudFormation não mostra uma lista suspensa de valores para esse tipo de parâmetro.

`List<AWS::EC2::Instance::Id>`

Uma matriz de IDs de instâncias do Amazon EC2, como `i-1e731a32`, `i-1e731a34`.

`List<AWS::EC2::SecurityGroup::GroupName>`

Uma matriz de nomes de grupos de segurança padrão da VPC, como `my-sg-abc`, `my-sg-def`.

`List<AWS::EC2::SecurityGroup::Id>`

Uma matriz de IDs de security groups, como `sg-a123fd85`, `sg-b456fd85`.

`List<AWS::EC2::Subnet::Id>`

Uma matriz de IDs de sub-rede, como `subnet-123a351e`, `subnet-456b351e`.

`List<AWS::EC2::Volume::Id>`

Uma matriz de IDs de volumes do Amazon EBS, como `vol-3cdd3f56`, `vol-4cdd3f56`.

`List<AWS::EC2::VPC::Id>`

Uma matriz de IDs de VPC, como `vpc-a123baa3`, `vpc-b456baa3`.

`List<AWS::Route53::HostedZone::Id>`

Uma matriz de IDs de zonas hospedadas do Amazon Route 53, como `Z23YXV40VPL04A`, `Z23YXV40VPL04B`.

Tipos de parâmetros compatíveis do Systems Manager

O CloudFormation é compatível com os seguintes tipos de parâmetros do Systems Manager:

`AWS::SSM::Parameter::Name`

O nome de uma chave de parâmetro do Systems Manager. Use esse tipo de parâmetro somente para verificar se um parâmetro necessário existe. O CloudFormation não recupera o valor real associado ao parâmetro.

`AWS::SSM::Parameter::Value<String>`

Um parâmetro do Systems Manager cujo valor é uma sequência. Corresponde ao tipo de parâmetro `String` no Armazenamento de parâmetros.

`AWS::SSM::Parameter::Value<List<String>>` ou
`AWS::SSM::Parameter::Value<CommaDelimitedList>`

Um parâmetro do Systems Manager cujo valor é uma lista de sequências. Corresponde ao tipo de parâmetro `StringList` no Armazenamento de parâmetros.

`AWS::SSM::Parameter::Value<AWS-specific parameter type>`

Um parâmetro do Systems Manager cujo valor é um tipo de parâmetro específico da AWS.

Por exemplo, o seguinte especifica o tipo `AWS::EC2::KeyPair::KeyName`:

- `AWS::SSM::Parameter::Value<AWS::EC2::KeyPair::KeyName>`

`AWS::SSM::Parameter::Value<List<AWS-specific parameter type>>`

Um parâmetro do Systems Manager cujo valor é uma lista de tipos de parâmetros específicos da AWS.

Por exemplo, o seguinte especifica uma lista de tipos `AWS::EC2::KeyPair::KeyName`:

- `AWS::SSM::Parameter::Value<List<AWS::EC2::KeyPair::KeyName>>`

Tipos de parâmetros incompatíveis do Systems Manager

O CloudFormation não é compatível com o seguinte tipo de parâmetro do Systems Manager:

- Listas de tipos de parâmetros do Systems Manager. Por exemplo:

`List<AWS::SSM::Parameter::Value<String>>`

Além disso, o CloudFormation não é compatível com a definição de parâmetros de modelo como tipos de parâmetros `SecureString` do Systems Manager. No entanto, você pode especificar strings seguras como valores de parâmetros para determinados recursos. Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

Orientações passo a passo do CloudFormation

Este documento fornece uma coleção de orientações detalhadas desenvolvidas para oferecer a você experiência prática em implantações de pilhas.

- [Consultar saídas de recurso em outra pilha do CloudFormation](#): essa demonstração mostra como referenciar as saídas de uma pilha do CloudFormation em outra pilha. Em vez de incluir todos os

recursos em uma única pilha, você pode criar recursos da AWS relacionados em pilhas separadas para criar modelos mais modulares e reutilizáveis.

- [Implantar aplicações no Amazon EC2](#): saiba como usar o CloudFormation para instalar, configurar e iniciar aplicações automaticamente em instâncias do Amazon EC2. Dessa forma, você pode duplicar facilmente implantações e atualizar instalações existentes sem se conectar diretamente às instâncias.
- [Atualizar uma pilha do CloudFormation](#): percorra uma progressão simples de atualizações de uma pilha em execução com o CloudFormation.
- [Criar uma aplicação dimensionada e com balanceamento de carga](#): descubra como usar o CloudFormation para criar uma aplicação escalável e com balanceamento de carga. Essa demonstração aborda a criação de um grupo do Auto Scaling, um balanceador de carga e outros recursos relacionados para garantir que sua aplicação possa lidar com cargas de tráfego variáveis e manter a alta disponibilidade.
- [Emparelhar com uma VPC em outra Conta da AWS](#): essa demonstração orienta você pelo processo de criação de uma conexão de emparelhamento de Nuvem Privada Virtual (VPC) entre duas VPCs em diferentes Contas da AWS. O emparelhamento de VPC ajuda você a rotear o tráfego entre as VPCs e acessar os recursos como se eles fizessem parte da mesma rede.
- [Executar implantações azuis/verdes do ECS por meio do CodeDeploy usando o CloudFormation](#): descubra como usar o CloudFormation para realizar implantações azuis/verdes do AWS CodeDeploy no Amazon ECS. Implantações azuis/verdes são uma forma de atualizar aplicações ou serviços com o mínimo de tempo de inatividade.

Consultar saídas de recurso em outra pilha do CloudFormation

Essa demonstração mostra como referenciar as saídas de uma pilha do CloudFormation em outra pilha para criar modelos mais modulares e reutilizáveis.

Em vez de incluir todos os recursos em uma única pilha, você cria recursos da AWS relacionados em pilhas separadas. Depois, você pode consultar as saídas de recursos necessárias de outras pilhas. Restringindo referências de pilha cruzada a saídas, você controla as partes de uma pilha referenciadas por outras pilhas.

Por exemplo, você pode ter uma pilha de rede com uma VPC, um grupo de segurança e uma sub-rede para aplicações web públicas e uma pilha de aplicações web públicas separadas. Para garantir que as aplicações Web usem o grupo de segurança e a sub-rede da pilha de rede, você cria uma referência de pilha cruzada que permite que a pilha de aplicações web referenciem saídas da pilha

de rede. Com uma referência de pilha cruzada, os proprietários das pilhas de aplicações web não precisam criar nem manter regras ou ativos de rede.

Para criar uma referência de pilha cruzada, use o campo de saída `Export` a fim de sinalizar o valor de uma saída de recurso para exportação. Em seguida, use a função intrínseca `Fn::ImportValue` para importar o valor. Para obter mais informações, consulte [Obter resultados exportados de uma pilha do CloudFormation implantada](#).

Note

O CloudFormation é um serviço gratuito. No entanto, você é cobrado pelos recursos da AWS incluídos nas pilhas segundo a taxa atual de cada um. Para obter mais informações sobre a definição de preço da AWS, consulte a [página de detalhes de cada produto](#).

Tópicos

- [Usar um modelo de exemplo para criar uma pilha de rede](#)
- [Usar um modelo de exemplo para criar uma pilha de aplicações Web](#)
- [Verificar se a pilha funciona conforme o previsto](#)
- [Solucionar problemas de mapeamento de AMIs](#)
- [Limpe os seus recursos](#)

Usar um modelo de exemplo para criar uma pilha de rede

Antes de iniciar este passo a passo, verifique se você tem permissões do IAM para usar todos os seguintes serviços: Amazon VPC, Amazon EC2 e CloudFormation.

A pilha de rede contém a VPC, o grupo de segurança e a sub-rede que você usará na pilha de aplicações web. Além desses recursos, a pilha de rede cria um gateway de Internet e tabelas de roteamento para permitir o acesso público.

Você deverá criar essa pilha antes de criar a pilha de aplicações web. Se você criar a pilha de aplicações web primeiro, ela não terá um grupo de segurança ou sub-rede.

O modelo da pilha está disponível no seguinte URL: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template>. Para ver os recursos que a pilha criará, escolha o link, que abre o modelo. Na seção `Outputs`, você pode ver os recursos de rede exportados pelo modelo de amostra. Os nomes dos recursos exportados são prefixados com o

nome da pilha caso você exporte recursos de rede de outras pilhas. Quando importam recursos de rede, os usuários podem especificar de qual pilha os recursos são importados.

Para criar a pilha de rede

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, escolha Criar pilha no canto superior direito e selecione Com novos recursos (padrão).
3. Escolha O modelo está pronto e, na seção Especificar modelo, escolha URL do Amazon S3.
4. Em URL do Amazon S3, cole o seguinte URL: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**.
5. Escolha Próximo.
6. Para Stack name (Nome da pilha), digite **SampleNetworkCrossStack** e selecione Next (Próximo).

 Note

Registre o nome dessa pilha. Você precisará do nome da pilha quando ativar a pilha de aplicações web.

7. Escolha Próximo. Para esta descrição, você não precisa adicionar tags nem especificar configurações avançadas.
8. Verifique se o nome da pilha e o URL do modelo estão corretos e escolha Create stack (Criar pilha).

Pode levar alguns minutos para que o CloudFormation crie a pilha. Aguarde até que todos os recursos tenham sido criados com êxito antes de continuar criando a pilha de aplicações web.

9. Para monitorar o progresso, visualize os eventos da pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Usar um modelo de exemplo para criar uma pilha de aplicações Web

A pilha de aplicações web cria uma instância EC2 que usa o grupo de segurança e a sub-rede da pilha de rede.

Você deve criar essa pilha na mesma Região da AWS que a pilha de rede.

O modelo da pilha está disponível no seguinte URL: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>. Para ver os recursos que a pilha criará, escolha o link, que abrirá o modelo. Na seção Resources, visualize as propriedades da instância do EC2. Você pode ver como os recursos de rede são importados de outra pilha usando a função Fn::ImportValue.

Para criar a pilha de aplicações web

1. Na página Pilhas, escolha Criar pilha no canto superior direito e depois Com novos recursos (padrão).
2. Escolha O modelo está pronto e, na seção Especificar modelo, escolha URL do Amazon S3.
3. Em URL do Amazon S3, cole o seguinte URL: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**.
4. Escolha Próximo.
5. Para Nome da pilha, digite **SampleWebAppCrossStack**. Na seção Parâmetros, use o valor padrão do parâmetro NetworkStackName e escolha Próximo.

O modelo de amostra usa o valor do parâmetro para especificar de qual pilha importar valores.

6. Escolha Próximo. Para esta descrição, você não precisa adicionar tags nem especificar configurações avançadas.
7. Verifique se o nome da pilha e o URL do modelo estão corretos e escolha Create stack (Criar pilha).

Pode levar alguns minutos para que o CloudFormation crie a pilha.

Verificar se a pilha funciona conforme o previsto

Após a criação da pilha, visualize os recursos e observe o ID da instância. Para obter mais informações sobre como visualizar recursos de pilha, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Para verificar o grupo de segurança da instância e a sub-rede, visualize as propriedades da instância no [console do Amazon EC2](#). Caso a instância use o grupo de segurança e a sub-rede da pilha SampleNetworkCrossStack, você criou com êxito uma referência de pilha cruzada.

Use o console para visualizar as saídas da pilha e o URL do site de exemplo para verificar se a aplicação web está em execução. Para obter mais informações, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Solucionar problemas de mapeamento de AMIs

Se você receber o erro `Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64`, o modelo não inclui um mapeamento de AMIs para sua Região da AWS. Em vez de atualizar o mapeamento, recomendamos usar os parâmetros públicos do Systems Manager para referenciar dinamicamente as AMIs mais recentes:

1. Baixe o modelo `SampleWebAppCrossStack` para sua máquina local em:
<https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>.
2. Exclua toda a seção de mapeamento `AWSRegionArch2AMI`.
3. Adicione o seguinte parâmetro do Systems Manager:

```
"LatestAmiId": {
  "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
  "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
  "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
}
```

4. Substitua a referência de `ImageId` existente:

```
"ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" } ,
  "HVM64" ] },
```

Por:

```
"ImageId": { "Ref": "LatestAmiId" },
```

Esse parâmetro é automaticamente resolvido para a AMI mais recente do Amazon Linux 2 para a região em que você implanta a pilha.

Para outras distribuições Linux, use o caminho de parâmetro apropriado. Para obter mais informações, consulte [Descobrir parâmetros públicos no Parameter Store](#) no Guia do usuário do AWS Systems Manager.

5. Carregue o modelo modificado em um bucket do S3 em sua conta:

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. Ao criar a pilha, especifique o URL do seu modelo no S3 em vez do URL de exemplo.

Limpe os seus recursos

Para se certificar de que você não seja cobrado por serviços indesejados, exclua as pilhas.

Para excluir as pilhas

1. No console do CloudFormation, selecione a pilha SampleWebAppCrossStack.
2. Escolha Actions (Ações) e selecione Delete stack (Excluir pilha).
3. Na mensagem de confirmação, selecione Delete (Excluir).
4. Depois que a pilha tiver sido excluída, repita as mesmas etapas para a pilha SampleNetworkCrossStack.

Note

Aguarde até o CloudFormation excluir por completo a pilha SampleWebAppCrossStack. Se a instância do EC2 ainda estiver em execução na VPC, o CloudFormation não excluirá a VPC na pilha SampleNetworkCrossStack.

Implantar aplicações no Amazon EC2

É possível usar o CloudFormation para instalar, configurar e iniciar aplicações automaticamente em instâncias do Amazon EC2. Isso permite que você duplique implantações e atualize instalações existentes facilmente sem se conectar diretamente à instância, o que pode economizar muito tempo e esforço.

O CloudFormation inclui um conjunto de scripts auxiliares (cfn-init, cfn-signal, cfn-get-metadata e cfn-hup) baseados em cloud-init. Você chama esses scripts auxiliares de seus modelos do CloudFormation para instalar, configurar e atualizar os aplicativos em instâncias do Amazon EC2 que estão no mesmo modelo. Para obter mais informações, consulte [Referência de scripts auxiliares do CloudFormation](#) no Guia de referência de modelos do CloudFormation.

No [tutorial de conceitos básicos](#), você criou um servidor Web simples usando UserData com um script bash básico. Embora isso tenha funcionado para uma página simples “Hello World”, as aplicações reais geralmente precisam de uma configuração mais sofisticada, incluindo:

- Vários pacotes de software instalados na ordem correta.
- Arquivos de configuração complexos criados com conteúdo específico.
- Serviços iniciados e configurados para serem executados automaticamente.
- Tratamento de erros e validação do processo de configuração.

Os scripts auxiliares do CloudFormation fornecem uma maneira mais robusta e sustentável de configurar instâncias do EC2 em comparação com os scripts bash básicos em UserData. O script auxiliar `cfn-init` lê os dados de configuração dos metadados do seu modelo e os aplica sistematicamente à sua instância.

Neste tutorial, você vai aprender como usar o script auxiliar `cfn-init` e como monitorar o processo de bootstrap.

Note

O CloudFormation é gratuito, mas os recursos do Amazon EC2 que você criar serão cobrados. Porém, se você for novo na AWS, poderá aproveitar o [nível gratuito](#) para minimizar ou eliminar os custos durante este processo de aprendizagem.

Tópicos

- [Pré-requisitos](#)
- [Como entender os conceitos de bootstrap](#)
- [Começar com um exemplo simples de bootstrap](#)
- [Como adicionar arquivos e comandos](#)
- [Como adicionar segurança](#)
- [O modelo completo de bootstrap](#)
- [Criar uma pilha usando o console](#)
- [Monitorar o processo de bootstrap](#)
- [Testar o servidor Web bootstrapped](#)
- [Solucionar problemas de bootstrap](#)

- [Limpar os recursos](#)
- [Próximas etapas](#)

Pré-requisitos

- Você deve ter concluído o tutorial [Criar sua primeira pilha](#) ou ter uma experiência equivalente com os conceitos básicos do CloudFormation.
- Você deve ter acesso a uma Conta da AWS com um usuário ou perfil do IAM que tenha permissões para usar o Amazon EC2 e o CloudFormation ou ter acesso de usuário administrativo.
- Você deve ter uma nuvem privada virtual (VPC) que tenha acesso à internet. Este modelo de tutorial requer uma VPC padrão, que vem automaticamente com as Contas da AWS mais recentes. Se você não tiver uma VPC padrão, ou se ela tiver sido excluída, consulte a seção de solução de problemas no tutorial [Criar sua primeira pilha](#) para obter soluções alternativas.

Como entender os conceitos de bootstrap

Vamos entender os principais conceitos que fazem o bootstrapping funcionar antes de criar o modelo.

O script de ajuda **cfn-init**

O CloudFormation fornece os scripts auxiliares do Python que podem ser usados para instalar software e iniciar serviços em uma instância do Amazon EC2. O script `cfn-init` lê os metadados dos recursos do seu modelo e aplica a configuração à sua instância.

O processo funciona da seguinte maneira:

1. Você define a configuração na seção Metadata do seu recurso do EC2.
2. Você chama `cfn-init` no script UserData.
3. O script `cfn-init` lê os metadados e aplica a configuração.
4. A instância é configurada de acordo com suas especificações.

Estrutura de metadados

A configuração é definida em uma estrutura específica dentro da instância do EC2.

Resources:

```
EC2Instance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
        files:
        commands:
        services:
```

O script `cfn-init` processa essas seções na seguinte ordem: pacotes, grupos, usuários, origens, arquivos, comandos e serviços.

Começar com um exemplo simples de bootstrap

Vamos começar com um exemplo mínimo de bootstrap que apenas instala e inicia o Apache.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            # Install Apache web server
          yum:
            httpd: []
          services:
            # Start Apache and enable it to start on boot
            systemd:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64
        # Script that runs when instance starts
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
          region ${AWS::Region}
```

Este exemplo simples demonstra os principais conceitos:

- A seção `packages` instala o pacote `httpd` usando `yum`. Isso funciona no Amazon Linux e em outras distribuições Linux que usam `yum`.
- A seção `services` garante que o `httpd` inicie e seja executado automaticamente.
- O `UserData` instala as ferramentas de `bootstrap` mais recentes e chama o `cfn-init`.

Como adicionar arquivos e comandos

Agora, vamos aprimorar nosso exemplo adicionando uma página da Web personalizada e um arquivo de log no diretório `/var/log` na instância do EC2.

Criar arquivos

A seção `files` permite que você crie arquivos na instância com conteúdo específico. A barra vertical (`|`) permite que você passe um bloco literal de texto (código HTML) como conteúdo do arquivo (`/var/www/html/index.html`).

```
files:
  /var/www/html/index.html:
    content: |
      <body>
        <h1>Congratulations, you have successfully launched the AWS CloudFormation
sample.</h1>
      </body>
```

Executar comandos

A seção `commands` permite que você execute comandos shell durante o processo de `bootstrap`. Esse comando cria um arquivo de log em `/var/log/welcome.txt` na instância do EC2. Para visualizá-lo, você precisa de um par de chaves do Amazon EC2 para usar no acesso por SSH e um intervalo de endereços IP que possa ser usado para o SSH na instância (não abordados aqui).

```
commands:
  createWelcomeLog:
    command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

Como adicionar segurança

Como estamos configurando um servidor Web, precisamos permitir que o tráfego da Web (HTTP) chegue à nossa instância do EC2. Para isso, criaremos um grupo de segurança que permita tráfego de entrada na porta 80 do seu endereço IP. As instâncias do EC2 também precisam enviar tráfego

para a internet, por exemplo, para instalar atualizações de pacotes. Por padrão, os grupos de segurança permitem todo o tráfego de saída. Vamos então associar esse grupo de segurança à nossa instância do EC2 usando a propriedade `SecurityGroupIds`.

```
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow HTTP access from my IP address
    SecurityGroupIngress:
      - IpProtocol: tcp
        Description: HTTP
        FromPort: 80
        ToPort: 80
        CidrIp: !Ref MyIP
```

O modelo completo de bootstrap

Agora, vamos juntar todas as peças. Confira o modelo completo que combina todos os conceitos que discutimos.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
```

```
MaxLength: 18
Default: 0.0.0.0/0
AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\/\d{1,2}$'
ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
```

Resources:**WebServerSecurityGroup:**

```
Type: AWS::EC2::SecurityGroup
```

Properties:

```
GroupDescription: Allow HTTP access from my IP address
```

SecurityGroupIngress:

```
- IpProtocol: tcp
  Description: HTTP
  FromPort: 80
  ToPort: 80
  CidrIp: !Ref MyIP
```

WebServer:

```
Type: AWS::EC2::Instance
```

Metadata:

```
AWS::CloudFormation::Init:
```

config:**packages:****yum:**

```
httpd: []
```

files:

```
/var/www/html/index.html:
```

```
content: |
```

```
<body>
```

```
<h1>Congratulations, you have successfully launched the AWS
```

```
CloudFormation sample.</h1>
```

```
</body>
```

commands:**createWelcomeLog:**

```
command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

services:**systemd:****httpd:**

```
enabled: true
```

```
ensureRunning: true
```

Properties:

```
ImageId: !Ref LatestAmiId
```

```
InstanceType: !Ref InstanceType
```

```
SecurityGroupIds:
```



```
- !Ref WebServerSecurityGroup
UserData: !Base64
  Fn::Sub: |
    #!/bin/bash
    yum install -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
  Tags:
    - Key: Name
      Value: Bootstrap Tutorial Web Server

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServer.PublicDnsName}'
    Description: EC2 instance public DNS name
```

Criar uma pilha usando o console

O procedimento a seguir envolve o upload do modelo de pilha de amostra a partir de um arquivo. Abra um editor de texto na sua máquina local e adicione o modelo. Salve o arquivo com o nome `samplelinux2stack.template`.

Para executar o modelo de pilha

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Escolha Create stack (Criar pilha), With new resources (Com novos recursos (padrão)).
3. Em Especificar modelo, escolha Carregar um arquivo de modelo, Escolher arquivo para carregar o arquivo `samplelinux2stack.template`.
4. Escolha Próximo.
5. Na página Especificar detalhes da pilha, digite **BootstrapTutorialStack** como o nome da pilha.
6. Em Parâmetros, faça o seguinte:
 - LatestAmild: deixe o valor padrão.
 - InstanceType: escolha t2.micro ou t3.micro como o tipo de instância do EC2.
 - MyIP: insira seu endereço IP público com um sufixo /32.
7. Escolha Avançar duas vezes e, em seguida, Enviar para criar a pilha.

Monitorar o processo de bootstrap

Os processos de bootstrap demoram mais do que simples execuções do EC2 porque software adicional está sendo instalado e configurado.

Para monitorar o progresso do bootstrap

1. No console do CloudFormation, selecione sua pilha e abra a guia Eventos.
2. Fique atento ao evento `WebServer CREATE_IN_PROGRESS`. O processo de bootstrap começa após a inicialização da instância.
3. Normalmente, o processo de bootstrap leva alguns minutos. Você verá `WebServer CREATE_COMPLETE` quando tiver terminado.

Caso queira ver o que está acontecendo durante o processo de bootstrap, você pode verificar os logs da instância.

Para visualizar os logs de bootstrap (opcional)

1. Abra o [console do EC2](#) e encontre sua instância.
2. Selecione a instância e escolha Ações, Monitorar e solucionar problemas e Obter log do sistema para ver o progresso do bootstrap.
3. Se você não visualizar os logs de imediato, aguarde e atualize a página.

Testar o servidor Web bootstrapped

Quando sua pilha mostrar `CREATE_COMPLETE`, teste seu servidor Web.

Para testar o servidor Web

1. No console do CloudFormation, acesse a guia Saídas da sua pilha.
2. Clique no valor `WebsiteURL` para abrir o servidor Web em uma nova guia.
3. Você verá sua página da Web personalizada com a mensagem `Congratulations, you have successfully launched the AWS CloudFormation sample`.

 Note

Se a página não for carregada de imediato, aguarde um minuto e tente novamente. O processo de bootstrap ainda pode estar sendo concluído mesmo após a pilha mostrar CREATE_COMPLETE.

Solucionar problemas de bootstrap

Se o processo de bootstrap falhar ou se o servidor Web não estiver funcionando, confira abaixo os problemas e soluções comuns.

Problemas comuns

- Falha na criação da pilha: verifique a guia Eventos para ver mensagens de erro específicas.
- Servidor Web não acessível: verifique se o endereço IP está correto no parâmetro MyIP. Lembre-se de incluir /32 no final.
- Falha no processo de bootstrap: a instância pode ser inicializada, mas o `cfn-init` falha. Verifique os logs do sistema conforme descrito na seção de monitoramento.

Limpar os recursos

Para evitar cobranças contínuas, você pode fazer a limpeza excluindo a pilha e seus recursos.

Para excluir a pilha e seus recursos

1. Abra o [console do CloudFormation](#).
2. Na página Pilhas, selecione a opção ao lado do nome da pilha que você criou (**BootstrapTutorialStack**) e escolha Excluir.
3. Quando a confirmação for solicitada, escolha Excluir.
4. Monitore o andamento do processo de exclusão da pilha na guia Evento. O status de **BootstrapTutorialStack** é alterado para DELETE_IN_PROGRESS. Quando o CloudFormation conclui a exclusão da pilha, ela é removida da lista.

Próximas etapas

Parabéns! Você aprendeu com êxito como fazer bootstrap de instâncias do EC2 com o CloudFormation. Agora você já sabe:

- Como usar scripts `cfn-init` auxiliares
- Como estruturar metadados para bootstrapping
- Como instalar pacotes, criar arquivos, executar comandos e gerenciar serviços
- Como monitorar problemas de bootstrap

Para continuar aprendendo:

- Saiba como atualizar uma pilha em execução e usar o script auxiliar `cfn-hup`. Para obter mais informações, consulte [Atualizar uma pilha do CloudFormation](#).
- Aprenda a fazer bootstrap de uma pilha do Windows. Para obter mais informações, consulte [Fazer o bootstrap de pilhas do CloudFormation baseadas em Windows](#).
- Conheça cenários de bootstrap mais complexos com vários conjuntos de configurações. Para obter mais informações, consulte [cfn-init](#) e [AWS::CloudFormation::Init](#) no Guia de referência de modelos do CloudFormation.
- Saiba mais sobre o script `cfn-signal` para relatar o status de conclusão do bootstrap. Para obter mais informações, consulte [cfn-signal](#) no Guia de referência de modelos do CloudFormation.

Atualizar uma pilha do CloudFormation

Note

Este tutorial se baseia nos conceitos apresentados no tutorial [Implantar aplicações no Amazon EC2](#). Se você ainda não concluiu esse tutorial, recomendamos que o faça primeiro para entender a inicialização do EC2 com o CloudFormation.

Este tópico demonstra uma progressão simples de atualizações para uma pilha em execução. Demonstraremos as seguintes etapas:

1. Criar a pilha inicial: crie uma pilha usando uma AMI básica do Amazon Linux 2, instalando o Apache Web Server e uma aplicação PHP simples usando os scripts auxiliares do CloudFormation.
2. Atualizar a aplicação: atualize um dos arquivos na aplicação e implante o software usando o CloudFormation.
3. Adicionar um par de chaves: adicione um par de chaves do Amazon EC2 à instância e, em seguida, atualize o grupo de segurança para permitir acesso SSH à instância.
4. Atualizar o tipo de instância: altere o tipo de instância da instância subjacente do Amazon EC2.
5. Atualizar a AMI: altere a imagem de máquina da Amazon (AMI) para a instância do Amazon EC2 na sua pilha.

 Note

O CloudFormation é gratuito, mas os recursos do Amazon EC2 que você criar serão cobrados. Porém, se você for novo na AWS, poderá aproveitar o [nível gratuito](#) para minimizar ou eliminar os custos durante este processo de aprendizagem.

Tópicos

- [Etapa 1: criar a pilha inicial](#)
- [Etapa 2: atualizar a aplicação](#)
- [Etapa 3: adicionar o acesso por SSH com um par de chaves](#)
- [Etapa 4: atualizar o tipo de instância](#)
- [Etapa 5: atualizar a AMI](#)
- [Considerações sobre disponibilidade e impacto](#)
- [Recursos relacionados](#)

Etapa 1: criar a pilha inicial

Vamos começar com a criação de uma pilha que possamos usar em todo o restante desta seção. Fornecemos um modelo simples que executa uma única aplicação web de instância PHP hospedada no Apache Web Server e em execução em uma AMI do Amazon Linux 2.

O Apache Web Server, o PHP e a aplicação PHP simples são instalados pelos scripts auxiliares do CloudFormation instalados por padrão na AMI do Amazon Linux 2. O trecho do modelo a seguir

mostra os metadados que descrevem os pacotes e os arquivos a instalar, neste caso o Apache Web Server e a infraestrutura PHP do repositório Yum da AMI do Amazon Linux 2. O trecho também mostra a seção `Services`, que garante que o Apache Web Server esteja em execução.

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
            php: []
        files:
          /var/www/html/index.php:
            content: |
              <?php
              echo '<h1>Hello World!</h1>';
              ?>
            mode: '000644'
            owner: apache
            group: apache
        services:
          systemd:
            httpd:
              enabled: true
              ensureRunning: true
```

A aplicação em si é um exemplo de “Hello World” totalmente definida dentro do modelo. Para um aplicativo do mundo real, os arquivos podem ser armazenados no Amazon S3, no GitHub ou em outro repositório e referenciados a partir do modelo. O CloudFormation pode baixar pacotes (como RPMs ou RubyGems) nem referenciar arquivos individuais e expandir arquivos `.zip` e `.tar` para criar os artefatos da aplicação na instância do Amazon EC2.

O modelo habilita e configura o daemon `cfn-hup` para receber alterações na configuração definida nos metadados da instância do Amazon EC2. Usando o daemon `cfn-hup`, você pode atualizar o software de aplicação, como a versão do Apache ou do PHP, ou o próprio arquivo da aplicação PHP no CloudFormation. O trecho a seguir do mesmo recurso do Amazon EC2 no modelo mostra os itens necessários para configurar o `cfn-hup` para chamar o `cfn-init` a cada dois minutos para observar e aplicar atualizações nos metadados. Caso contrário, o `cfn-init` só é executado uma vez na inicialização.

```

files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack=${AWS::StackId}
      region=${AWS::Region}
      # The interval used to check for changes to the resource metadata in minutes.
      Default is 15
      interval=2
      mode: '000400'
      owner: root
      group: root
  /etc/cfn/hooks.d/cfn-auto-reloader.conf:
    content: !Sub |
      [cfn-auto-reloader-hook]
      triggers=post.update
      path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
      action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --region
      ${AWS::Region}
      runas=root
services:
  systemd:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf

```

Para concluir a pilha, na seção **Properties** da definição da instância do Amazon EC2, a propriedade **UserData** contém o script `cloud-init` que chama o `cfn-init` a fim de instalar os pacotes e arquivos. Para obter mais informações, consulte [Referência de scripts auxiliares do CloudFormation](#) no Guia de referência de modelos do CloudFormation. O modelo também cria um grupo de segurança do Amazon EC2.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t3.micro

AllowedValues:

- t3.nano
- t3.micro
- t3.small
- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

Resources:**WebServerInstance:**

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

yum update -y aws-cfn-bootstrap


```

    # Run cfn-init
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region} || error_exit 'Failed to run cfn-init'

    # Start up the cfn-hup daemon to listen for changes to the EC2 instance
metadata
    /opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
    # Signal success or failure
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
          mode: '000400'
          owner: root
          group: root
        /etc/cfn/hooks.d/cfn-auto-reloader.conf:
          content: !Sub |
            [cfn-auto-reloader-hook]
            triggers=post.update
            path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}

```

```
    runas=root
  services:
    systemd:
      httpd:
        enabled: true
        ensureRunning: true
      cfn-hup:
        enabled: true
        ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  CreationPolicy:
    ResourceSignal:
      Timeout: PT5M

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
    Description: URL of the web application
```

Para iniciar uma pilha desse modelo

1. Copie o modelo e salve-o localmente no sistema como um arquivo de texto. Anote o local, pois será necessário usar o arquivo em uma etapa posterior.
2. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
3. Escolha Criar pilha, com novos recursos (padrão).
4. Escolha Escolher um modelo existente.
5. Em Especificar modelo, escolha Carregar um arquivo de modelo e navegue até o arquivo que você criou na primeira etapa e, em seguida, escolha Avançar.

6. Na página Especificar detalhes da pilha, insira **UpdateTutorial** como o nome da pilha.
7. Em Parâmetros, mantenha todos os parâmetros iguais e escolha Avançar duas vezes.
8. Na tela Revisar e criar, escolha Enviar.

Depois que o status da pilha for CREATE_COMPLETE, a guia Saída exibirá o URL de seu site. Se você clicar no valor da saída WebsiteURL, verá sua nova aplicação PHP funcionando.

Etapa 2: atualizar a aplicação

Agora que implantamos a pilha, vamos atualizar o aplicativo. Vamos fazer uma alteração simples no texto impresso pela aplicação. Para fazer isso, vamos adicionar um comando echo ao arquivo index.php como mostrado neste trecho de modelo:

```
files:
  /var/www/html/index.php:
    content: |
      <?php
      echo "<h1>Hello World!</h1>";
      echo "<p>This is an updated version of our application.</p>";
      ?>
    mode: '000644'
    owner: apache
    group: apache
```

Use um editor de texto para editar manualmente o arquivo de modelo que você salvou localmente.

Agora, atualize a pilha.

Para atualizar a pilha com seu modelo atualizado

1. No console do CloudFormation, selecione a pilha **UpdateTutorial**.
2. Escolha Atualizar, fazer uma atualização direta.
3. Escolha Substituir modelo existente.
4. Em Especificar modelo, escolha Carregar um arquivo de modelo e carregue o arquivo de modelo modificado, e depois escolha Avançar.
5. Na página Especificar detalhes da pilha, mantenha todos os parâmetros iguais e escolha Avançar.

6. Na página Revisar, analise suas alterações. Em Alterações, você verá que o CloudFormation atualizará o recurso `WebServerInstance`.
7. Selecione Enviar.

Quando a pilha estiver no estado `UPDATE_COMPLETE`, será possível escolher o valor da saída `WebsiteURL` novamente para verificar se as alterações da aplicação tiveram efeito. O daemon `cfn-hup` é executado a cada dois minutos e, portanto, pode levar até dois minutos para que a aplicação seja alterada após a atualização da pilha.

Para ver o conjunto de recursos que foram atualizados, vá para o console do CloudFormation. Na guia Eventos, examine os eventos da pilha. Neste caso específico, os metadados de `WebServerInstance` da instância do Amazon EC2 foram atualizados, o que fez o CloudFormation reavaliar também os outros recursos (`WebServerSecurityGroup`) para garantir que não havia outras alterações. Nenhum outro recurso da pilha foi modificado. O CloudFormation só atualiza os recursos da pilha que são afetados por alterações feitas na pilha. Tais alterações podem ser diretas, como alterações de propriedades ou metadados, ou podem ser devido a dependências ou fluxos de dados por meio de `Ref`, `GetAtt` ou outras funções intrínsecas do modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

Essa atualização simples ilustra o processo. No entanto, você pode fazer alterações muito mais complexas nos arquivos e pacotes implantados nas instâncias do Amazon EC2. Por exemplo, você pode decidir que precisa adicionar o MySQL à instância, junto com suporte PHP para MySQL. Para fazer isso, basta adicionar outros pacotes e arquivos juntamente com todos os serviços adicionais à configuração e, em seguida, atualizar a pilha para implantar as alterações.

```
packages:
  yum:
    httpd: []
    php: []
    mysql: []
    php-mysql: []
    mysql-server: []
    mysql-libs: []

    ...

services:
  systemd:
    httpd:
```

```
    enabled: true
    ensureRunning: true
  cfn-hup:
    enabled: true
    ensureRunning: true
    files:
      - /etc/cfn/cfn-hup.conf
      - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  mysqld:
    enabled: true
    ensureRunning: true
```

Você pode atualizar os metadados do CloudFormation a fim de atualizar para novas versões dos pacotes usados pela aplicação. Nos exemplos anteriores, a propriedade de versão de cada pacote está vazia, indicando que o `cfn-init` deve instalar a versão mais recente do pacote.

```
packages:
  yum:
    httpd: []
    php: []
```

Opcionalmente, você pode especificar uma string de versão para um pacote. Se você alterar a string de versão em chamadas de pilha de atualização subsequentes, a nova versão do pacote será implantada. Veja um exemplo de como usar os números de versão para pacotes do RubyGems. Qualquer pacote compatível com o versionamento pode ter versões específicas.

```
packages:
  rubygems:
    mysql: []
    rubygems-update:
      - "1.6.2"
    rake:
      - "0.8.7"
    rails:
      - "2.3.11"
```

Etapa 3: adicionar o acesso por SSH com um par de chaves

Também é possível atualizar um recurso no modelo para adicionar propriedades que não foram originalmente especificadas no modelo. Para ilustrar, vamos adicionar um par de chaves do Amazon

EC2 a uma instância existente do EC2 e, em seguida, abrir a porta 22 no grupo de segurança do Amazon EC2 para que você possa usar o Secure Shell (SSH) para acessar a instância.

Para adicionar acesso SSH a uma instância Amazon EC2 existente

1. Adicione outros dois parâmetros ao modelo para transmitir o nome de um par de chaves do Amazon EC2 existente e a localização do SSH.

Parameters:

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

2. Adicione a propriedade KeyName à instância Amazon EC2.

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref WebServerSecurityGroup

3. Adicione a porta 22 e a localização do SSH às regras de entrada referentes ao security group do Amazon EC2.

WebServerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

```
GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
SecurityGroupIngress:
  - IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 0.0.0.0/0
  - IpProtocol: tcp
    FromPort: 22
    ToPort: 22
    CidrIp: !Ref SSHLocation
```

4. Atualize a pilha usando as mesmas etapas explicadas em [Etapa 2: atualizar a aplicação](#).

Etapa 4: atualizar o tipo de instância

Agora, vamos demonstrar como atualizar a infraestrutura subjacente alterando o tipo de instância.

A pilha criada até o momento usa uma instância t3.micro do Amazon EC2. Vamos supor que o site recém-criado esteja recebendo cada vez mais tráfego do que uma instância t3.micro pode processar, e agora você deseja mudar para um tipo de instância m5.large do Amazon EC2. Se a arquitetura do tipo de instância for alterada, a instância será criada com uma AMI diferente. No entanto, tanto a t3.micro quanto a m5.large usam as mesmas arquiteturas de CPU e executam AMIs do Amazon Linux 2 (x86_64). Para obter mais informações, consulte [Compatibilidade para alterar o tipo de instância](#) no Guia do usuário do Amazon EC2.

Vamos usar o modelo que modificamos na etapa anterior para alterar o tipo de instância. Como InstanceType era um parâmetro de entrada para o modelo, não é necessário modificá-lo. É possível alterar o valor do parâmetro na página Especificar detalhes da pilha.

Para atualizar a pilha com um novo valor de parâmetro

1. No console do CloudFormation, selecione a pilha **UpdateTutorial**.
2. Escolha Atualizar, fazer uma atualização direta.
3. Escolha Usar modelo existente e depois Avançar.
4. Na página Especificar detalhes da pilha, altere o valor da caixa de texto InstanceType de t3.micro para m5.large. Em seguida, escolha Avançar duas vezes.
5. Na página Revisar, analise suas alterações. Em Alterações, você verá que o CloudFormation atualizará o recurso WebServerInstance.
6. Selecione Enviar.

Você pode alterar dinamicamente o tipo de instância de uma instância do Amazon EC2 baseada no EBS iniciando e parando a instância. O CloudFormation tenta otimizar a alteração atualizando o tipo de instância e reiniciando a instância, de modo que o ID de instância não seja alterado. Quando a instância é reiniciada, no entanto, o endereço IP público da instância é alterado. Para garantir que o endereço IP elástico esteja vinculado corretamente depois da alteração, o CloudFormation também o atualiza. Você pode ver as alterações no console do CloudFormation na guia Eventos.

Para verificar o tipo de instância a partir do Console de gerenciamento da AWS, abra o console do Amazon EC2 e localize sua instância.

Etapa 5: atualizar a AMI

Agora, vamos atualizar nossa pilha para usar o Amazon Linux 2023, que é a próxima geração do Amazon Linux.

A atualização da AMI é uma grande mudança que exige a substituição da instância. Não é possível simplesmente iniciar e interromper a instância para modificar a AMI. O CloudFormation considera isso uma alteração em uma propriedade imutável do recurso. Para alterar uma propriedade imutável, o CloudFormation precisa iniciar um recurso substituto, neste caso, uma nova instância do Amazon EC2 executando a nova AMI.

Vamos conferir como podemos atualizar nosso modelo de pilha para usar o Amazon Linux 2023. As principais mudanças incluem a atualização do parâmetro da AMI e a mudança do yum para o gerenciador de pacotes dnf.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2023 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

```
    Type: String
```

```
    Default: t3.micro
```

```
    AllowedValues:
```

- t3.nano
- t3.micro
- t3.small
- t3.medium

- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

```

- !Ref WebServerSecurityGroup
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    # Get the latest CloudFormation package
    dnf update -y aws-cfn-bootstrap
    # Run cfn-init
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region} || error_exit 'Failed to run cfn-init'

    # Start up the cfn-hup daemon to listen for changes to the EC2 instance
metadata
    /opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
    # Signal success or failure
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        dnf:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            echo "<p>This is an updated version of our application.</p>";
            echo "<p>Running on Amazon Linux 2023!</p>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
            mode: '000400'
            owner: root

```

```
    group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
      content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
      runas=root
  services:
    systemd:
      httpd:
        enabled: true
        ensureRunning: true
      cfn-hup:
        enabled: true
        ensureRunning: true
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  CreationPolicy:
    ResourceSignal:
      Timeout: PT5M

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
    Description: URL of the web application
```

Atualize a pilha usando as mesmas etapas explicadas em [Etapa 2: atualizar a aplicação](#).

Depois que a nova instância está em execução, o CloudFormation atualiza os outros recursos da pilha para que apontem para o novo recurso. Quando todos os novos recursos são criados, o recurso antigo é excluído. Esse processo é conhecido como UPDATE_CLEANUP. Desta vez, você notará que o ID da instância e o URL do aplicativo da instância na pilha foram alterados como resultado da atualização. Os eventos na tabela Evento contêm uma descrição “A atualização solicitada tem uma alteração em uma propriedade imutável e, portanto, está criando um novo recurso físico” para indicar que um recurso foi substituído.

Como alternativa: se você tem um código de aplicação escrito na AMI que deseja atualizar, pode usar o mesmo mecanismo de atualização de pilha para atualizar a AMI para que ela carregue a nova aplicação.

Para atualizar a AMI com o código personalizado da aplicação

1. Crie a nova AMI contendo as alterações da aplicação ou do sistema operacional. Para mais informações, consulte [Criar uma AMI](#) com suporte do Amazon EBS na Guia do Usuário do Amazon EC2.
2. Atualize o modelo para incorporar o novo ID da AMI.
3. Atualize a pilha usando as mesmas etapas explicadas em [Etapa 2: atualizar a aplicação](#).

Ao atualizar a pilha, o CloudFormation detecta que o ID da AMI foi alterado e, em seguida, aciona uma atualização da pilha da mesma maneira que iniciamos a atualização demonstrada acima.

Considerações sobre disponibilidade e impacto

Diferentes propriedades têm diferentes impactos nos recursos na pilha. Você pode usar o CloudFormation para atualizar qualquer propriedade. No entanto, antes de fazer qualquer alteração, você deve considerar estas perguntas:

1. Como a atualização afeta o recurso em si? Por exemplo, a atualização de um limite de alarme deixará o alarme inativo durante a atualização. Como vimos, a alteração do tipo de instância exige que a instância seja parada e reiniciada. O CloudFormation usa as ações de atualizar ou modificar nos recursos subjacentes para alterá-los. Para compreender o impacto de atualizações, é necessário verificar a documentação dos recursos específicos.
2. A alteração é mutável ou imutável? Algumas alterações nas propriedades de recursos, como alteração da AMI em uma instância do Amazon EC2, não são compatíveis com os serviços

subjacentes. No caso de alterações mutáveis, o CloudFormation usará APIs do tipo Atualizar ou Modificar para os recursos subjacentes. Para alterações de propriedade imutáveis, o CloudFormation criará novos recursos com as propriedades atualizadas e, em seguida, os vinculará à pilha antes de excluir os recursos antigos. Embora o CloudFormation tente reduzir o tempo de inatividade dos recursos de pilha, a substituição de um recurso é um processo de várias etapas e levará tempo. Durante a reconfiguração da pilha, seu aplicativo não estará totalmente operacional. Por exemplo, ele pode não ser capaz de atender a solicitações ou acessar um banco de dados.

Recursos relacionados

Para obter mais informações sobre como usar o CloudFormation para iniciar aplicações e sobre a integração com outros serviços de configuração e implantação, como o Puppet e o Opscode Chef, consulte os seguintes whitepapers:

- [Bootstrapping applications via CloudFormation](#)
- [Integrating CloudFormation with Opscode Chef](#)
- [Integrating CloudFormation with Puppet](#)

Criar uma aplicação dimensionada e com balanceamento de carga

Neste passo a passo, você cria uma pilha que ajuda a configurar uma aplicação dimensionada e com balanceamento de carga. O passo a passo fornece um modelo de exemplo que você usa para criar a pilha. O modelo de exemplo provisiona um grupo do Auto Scaling, um Application Load Balancer, grupos de segurança que controlam o tráfego para o balanceador de carga e para o grupo do Auto Scaling e uma configuração de notificação do Amazon SNS para publicar notificações sobre atividades de escalabilidade.

Esse modelo cria uma ou mais instâncias do Amazon EC2 e um Application Load Balancer. Você será cobrado pelos recursos da AWS usados se criar uma pilha com base nesse modelo.

Modelo de pilha completo

Vamos começar com o modelo.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
```

Parameters:**InstanceType:**

Description: The EC2 instance type

Type: String

Default: t3.micro

AllowedValues:

- t3.micro
- t3.small
- t3.medium

KeyName:

Description: Name of an existing EC2 key pair to allow SSH access to the instances

Type: AWS::EC2::KeyPair::KeyName

LatestAmiId:

Description: The latest Amazon Linux 2 AMI from the Parameter Store

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

OperatorEmail:

Description: The email address to notify when there are any scaling activities

Type: String

SSHLocation:

Description: The IP address range that can be used to SSH to the EC2 instances

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Subnets:

Type: 'List<AWS::EC2::Subnet::Id>'

Description: At least two public subnets in different Availability Zones in the selected VPC

VPC:

Type: AWS::EC2::VPC::Id

Description: A virtual private cloud (VPC) that enables resources in public subnets to connect to the internet

Resources:**ELBSecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: ELB Security Group

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80

```
    CidrIp: 0.0.0.0/0
EC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Security Group
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId:
          Fn::GetAtt:
            - ELBSecurityGroup
            - GroupId
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
EC2TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 30
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 15
    HealthyThresholdCount: 5
    Matcher:
      HttpCode: '200'
    Name: EC2TargetGroup
    Port: 80
    Protocol: HTTP
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: '20'
    UnhealthyThresholdCount: 3
    VpcId: !Ref VPC
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref EC2TargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP
```

```
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    Subnets: !Ref Subnets
    SecurityGroups:
      - !GetAtt ELBSecurityGroup.GroupId
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      SecurityGroupIds:
        - !Ref EC2SecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd
          systemctl enable httpd
          echo "<h1>Hello World!</h1>" > /var/www/html/index.html
NotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
WebServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref LaunchTemplate
      Version: !GetAtt LaunchTemplate.LatestVersionNumber
    MaxSize: '3'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref NotificationTopic
        NotificationTypes: ['autoscaling:EC2_INSTANCE_LAUNCH',
'autoscaling:EC2_INSTANCE_LAUNCH_ERROR', 'autoscaling:EC2_INSTANCE_TERMINATE',
'autoscaling:EC2_INSTANCE_TERMINATE_ERROR']
```



```
TargetGroupARNs:
  - !Ref EC2TargetGroup
VPCZoneIdentifier: !Ref Subnets
```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "InstanceType": {
      "Description": "The EC2 instance type",
      "Type": "String",
      "Default": "t3.micro",
      "AllowedValues": [
        "t3.micro",
        "t3.small",
        "t3.medium"
      ]
    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to allow SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "LatestAmiId": {
      "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "OperatorEmail": {
      "Description": "The email address to notify when there are any scaling activities",
      "Type": "String"
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": 9,
      "MaxLength": 18,
      "Default": "0.0.0.0/0",
      "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  },
```

```
"Subnets":{
  "Type":"List<AWS::EC2::Subnet::Id>",
  "Description":"At least two public subnets in different Availability Zones in the
selected VPC"
},
"VPC":{
  "Type":"AWS::EC2::VPC::Id",
  "Description":"A virtual private cloud (VPC) that enables resources in public
subnets to connect to the internet"
}
},
"Resources":{
  "ELBSecurityGroup":{
    "Type":"AWS::EC2::SecurityGroup",
    "Properties":{
      "GroupDescription":"ELB Security Group",
      "VpcId":{
        "Ref":"VPC"
      },
    },
    "SecurityGroupIngress":[
      {
        "IpProtocol":"tcp",
        "FromPort":80,
        "ToPort":80,
        "CidrIp":"0.0.0.0/0"
      }
    ]
  },
  "EC2SecurityGroup":{
    "Type":"AWS::EC2::SecurityGroup",
    "Properties":{
      "GroupDescription":"EC2 Security Group",
      "VpcId":{
        "Ref":"VPC"
      },
    },
    "SecurityGroupIngress":[
      {
        "IpProtocol":"tcp",
        "FromPort":80,
        "ToPort":80,
        "SourceSecurityGroupId":{
          "Fn::GetAtt":[
            "ELBSecurityGroup",
```

```

        "GroupId"
      ]
    }
  },
  {
    "IpProtocol": "tcp",
    "FromPort": 22,
    "ToPort": 22,
    "CidrIp": {
      "Ref": "SSHLocation"
    }
  }
]
}
},
"EC2TargetGroup": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "HealthCheckIntervalSeconds": 30,
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeoutSeconds": 15,
    "HealthyThresholdCount": 5,
    "Matcher": {
      "HttpCode": "200"
    },
    "Name": "EC2TargetGroup",
    "Port": 80,
    "Protocol": "HTTP",
    "TargetGroupAttributes": [
      {
        "Key": "deregistration_delay.timeout_seconds",
        "Value": "20"
      }
    ],
    "UnhealthyThresholdCount": 3,
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"ALBListener": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [

```

```

        {
            "Type": "forward",
            "TargetGroupArn": {
                "Ref": "EC2TargetGroup"
            }
        }
    ],
    "LoadBalancerArn": {
        "Ref": "ApplicationLoadBalancer"
    },
    "Port": 80,
    "Protocol": "HTTP"
}
},
"ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
        "Scheme": "internet-facing",
        "Subnets": {
            "Ref": "Subnets"
        },
        "SecurityGroups": [
            {
                "Fn::GetAtt": [
                    "ELBSecurityGroup",
                    "GroupId"
                ]
            }
        ]
    }
}
},
"LaunchTemplate": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
        "LaunchTemplateName": {
            "Fn::Sub": "${AWS::StackName}-launch-template"
        },
        "LaunchTemplateData": {
            "ImageId": {
                "Ref": "LatestAmiId"
            },
            "InstanceType": {
                "Ref": "InstanceType"
            }
        }
    }
}

```

```

      "KeyName":{
        "Ref":"KeyName"
      },
      "SecurityGroupIds":[
        {
          "Ref":"EC2SecurityGroup"
        }
      ],
      "UserData":{
        "Fn::Base64":{
          "Fn::Join":[
            "",
            [
              "#!/bin/bash\n",
              "yum update -y\n",
              "yum install -y httpd\n",
              "systemctl start httpd\n",
              "systemctl enable httpd\n",
              "echo \"<h1>Hello World!</h1>\n" > /var/www/html/index.html"
            ]
          ]
        }
      }
    },
    "NotificationTopic":{
      "Type":"AWS::SNS::Topic",
      "Properties":{
        "Subscription":[
          {
            "Endpoint":{
              "Ref":"OperatorEmail"
            },
            "Protocol":"email"
          }
        ]
      }
    },
    "WebServerGroup":{
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "LaunchTemplate":{
          "LaunchTemplateId":{

```

```
        "Ref": "LaunchTemplate"
    },
    "Version": {
        "Fn::GetAtt": [
            "LaunchTemplate",
            "LatestVersionNumber"
        ]
    },
    "MaxSize": "3",
    "MinSize": "1",
    "NotificationConfigurations": [
        {
            "TopicARN": {
                "Ref": "NotificationTopic"
            },
            "NotificationTypes": [
                "autoscaling:EC2_INSTANCE_LAUNCH",
                "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
                "autoscaling:EC2_INSTANCE_TERMINATE",
                "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
            ]
        }
    ],
    "TargetGroupARNs": [
        {
            "Ref": "EC2TargetGroup"
        }
    ],
    "VPCZoneIdentifier": {
        "Ref": "Subnets"
    }
}
}
```

Demonstração do modelo

A primeira parte desse modelo especifica o `Parameters`. A cada parâmetro deve receber um valor em tempo de execução para que o CloudFormation provisione a pilha com êxito. Os recursos especificados mais tarde nesse modelo fazem referência a esses valores e usam os dados.

- **InstanceType**: o tipo de instância do EC2 provisionado pelo Amazon EC2 Auto Scaling. Se não especificado, um padrão de `t3.micro` será usado.
- **KeyName**: um par de chaves do EC2 existente para permitir o acesso SSH às instâncias.
- **LatestAmiId**: a imagem de máquina da Amazon (AMI) para as instâncias. Se não for especificada, suas instâncias serão iniciadas com uma AMI do Amazon Linux 2, usando um parâmetro público AWS Systems Manager mantido pelo AWS. Para obter mais informações, consulte [Localizar parâmetros públicos](#), no Guia do usuário do AWS Systems Manager.
- **OperatorEmail**: o endereço de e-mail para o qual você deseja enviar notificações de atividade de escalabilidade.
- **SSHLocation**: o intervalo de endereços IP que pode ser usado para aplicar SSH nas instâncias.
- **Subnets**: pelo menos duas sub-redes públicas devem estar em diferentes zonas de disponibilidade.
- **VPC**: uma nuvem privada virtual (VPC) na sua conta que permite aos recursos em sub-redes públicas se conectarem à Internet.

 Note

Você pode usar a VPC padrão e as sub-redes padrão para permitir que as instâncias acessem a Internet. Nesse último caso, certifique-se de que a VPC tenha uma sub-rede mapeada para cada zona de disponibilidade da região na qual você está trabalhando. No mínimo, é necessário ter duas sub-redes públicas disponíveis para criar o balanceador de carga.

A próxima parte desse modelo especifica o **Resources**. Esta seção especifica os recursos da pilha e suas propriedades.

[AWS::EC2::SecurityGroup](#) do recurso **ELBSecurityGroup**

- **SecurityGroupIngress** contém uma regra de entrada TCP que permite acesso de todos os endereços IP (`"CidrIp": "0.0.0.0/0"`) na porta 80.

[AWS::EC2::SecurityGroup](#) do recurso **EC2SecurityGroup**

- **SecurityGroupIngress** contém duas regras de entrada: 1) uma regra de entrada TCP que permite acesso SSH (porta 22) a partir do intervalo de endereços IP que você fornece ao

parâmetro de entrada `SSHLocation` e 2) uma regra de entrada TCP que permite acesso do balanceador de carga especificando o grupo de segurança do balanceador de carga. A função [GetAtt](#) é usada para obter o ID do grupo de segurança com o nome lógico `ELBSecurityGroup`.

Recurso [AWS::ElasticLoadBalancingV2::TargetGroup](#) `EC2TargetGroup`

- `Port`, `Protocol` e `HealthCheckProtocol` especificam a porta da instância (80) e o protocolo (HTTP) para onde o `ApplicationLoadBalancer` roteia o tráfego e que o Elastic Load Balancing utiliza para verificar a integridade das instâncias do EC2.
- `HealthCheckIntervalSeconds` especifica que as instâncias do EC2 têm um intervalo de 30 segundos entre verificações de integridade. O `HealthCheckTimeoutSeconds` é definido como o tempo que Elastic Load Balancing espera por uma resposta do destino da verificação de integridade (15 segundos neste exemplo). Após o tempo limite, o Elastic Load Balancing marca essa verificação de integridade da instância do EC2 como não íntegra. Quando uma instância do EC2 falha em três verificações de integridade consecutivas (`UnhealthyThresholdCount`), o Elastic Load Balancing interrompe o roteamento do tráfego para essa instância do EC2 até que ela tenha cinco verificações de integridade positivas consecutivas (`HealthyThresholdCount`). Nesse ponto, o Elastic Load Balancing considera a instância íntegra e começa a rotear o tráfego para ela novamente.
- `TargetGroupAttributes` atualiza o valor do atraso de cancelamento do registro do grupo de destino para 20 segundos. Por padrão, o Elastic Load Balancing aguarda 300 segundos antes de concluir o processo de cancelamento do registro.

Recurso [AWS::ElasticLoadBalancingV2::Listener](#) `ALBListener`

- `DefaultActions` especifica a porta na qual o balanceador de carga escuta, o grupo de destino ao qual o balanceador de carga encaminha solicitações e o protocolo usado para rotear as solicitações.

Recurso [AWS::ElasticLoadBalancingV2::LoadBalancer](#) `ApplicationLoadBalancer`

- `Subnets` usa o valor do parâmetro de entrada `Subnets` como a lista de sub-redes públicas nas quais os nós do balanceador de carga serão criados.
- `SecurityGroup` obtém o ID do grupo de segurança que atua como um firewall virtual para seus nós de balanceador de carga a fim de controlar o tráfego de entrada. A função [GetAtt](#) é usada para obter o ID do grupo de segurança com o nome lógico `ELBSecurityGroup`.

Recurso [AWS::EC2::LaunchTemplate](#) LaunchTemplate

- ImageId usa o valor do parâmetro de entrada LatestAmiId como a AMI a ser usada.
- KeyName usa o valor do parâmetro de entrada KeyName como o par de chaves do EC2 a ser usado.
- SecurityGroupIds obtém o ID do grupo de segurança com o nome lógico EC2SecurityGroup que atua como um firewall virtual para suas instâncias do EC2 a fim de controlar o tráfego de entrada.
- UserData é um script de configuração executado depois que a instância está ativa e em execução. Neste exemplo, o script instala o Apache e cria um arquivo index.html.

Recurso [AWS::SNS::Topic](#) NotificationTopic

- Subscription usa o valor do parâmetro de entrada OperatorEmail como o endereço de e-mail do destinatário das notificações quando há atividades de escalabilidade.

Recurso [AWS::AutoScaling::AutoScalingGroup](#) WebServerGroup

- MinSize e MaxSize definem os números mínimo e máximo de instâncias do EC2 no grupo do Auto Scaling.
- TargetGroupARNs usa o ARN do grupo de destino com o nome lógico EC2TargetGroup. À medida que esse grupo do Auto Scaling é escalado, ele automaticamente registra e cancela o registro de instâncias nesse grupo de destino.
- VPCZoneIdentifier usa o valor do parâmetro de entrada Subnets como a lista de sub-redes públicas nas quais as instâncias do EC2 podem ser criadas.

Etapa 1: Iniciar a pilha

Para iniciar a pilha, verifique se você tem permissões do AWS Identity and Access Management (IAM) para usar todos estes serviços: Amazon EC2, Amazon EC2 Auto Scaling, AWS Systems Manager, Elastic Load Balancing, Amazon SNS e CloudFormation.

O procedimento a seguir envolve o upload do modelo de pilha de amostra a partir de um arquivo. Abra um editor de texto na máquina local e adicione um dos modelos. Salve o arquivo com o nome `sampleloadbalancedappstack.template`.

Para executar o modelo de pilha

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Escolha Create stack (Criar pilha), With new resources (Com novos recursos (padrão)).
3. Em Especificar modelo, escolha Carregar um arquivo de modelo, Escolher arquivo para carregar o arquivo `sampleloadbalancedappstack.template`.
4. Escolha Próximo.
5. Na página Specify stack details (Especificar detalhes da pilha), digite um nome para a pilha (por exemplo, **SampleLoadBalancedAppStack**).
6. Em Parameters (Parâmetros), analise os parâmetros da pilha e forneça valores para todos os parâmetros que não tenham valores padrão, incluindo OperatorEmail, SSHLocation, KeyName, VPC e Subnets.
7. Escolha Next (Próximo) duas vezes.
8. Na página Revisar, verifique e confirme as configurações.
9. Selecione Enviar.

Você pode visualizar o status da pilha no console do CloudFormation, na coluna Status. Quando o CloudFormation criar a pilha, você receberá o status CREATE_COMPLETE.

Note

Depois de criar a pilha, você deve confirmar a assinatura para que o endereço de e-mail possa começar a receber notificações. Para obter mais informações, consulte [Obter notificações do Amazon SNS quando o grupo do Auto Scaling é dimensionado](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Etapa 2: limpar recursos de amostra

Para ter certeza de que você não será cobrado por recursos de amostra não utilizados, exclua a pilha.

Para excluir a pilha

1. No console do CloudFormation, selecione a pilha SampleLoadBalancedAppStack.
2. Escolha Excluir.

3. Na mensagem de confirmação, escolha Delete stack (Excluir pilha).

O status de SampleLoadBalancedAppStack muda para DELETE_IN_PROGRESS. Quando o CloudFormation conclui a exclusão da pilha, ela é removida da lista.

Use o modelos de exemplo deste passo a passo para criar seus próprios modelos de pilha.

Para obter mais informações, consulte [Tutorial: Configurar uma aplicação dimensionada e com balanceamento de carga](#), no Guia do usuário do Amazon EC2 Auto Scaling.

Emparelhar com uma VPC em outra Conta da AWS

Você pode emparelhar com uma nuvem privada virtual (VPC) em outra Conta da AWS usando [AWS::EC2::VPCPeeringConnection](#). Isso cria uma conexão de rede entre duas VPCs que permite rotear o tráfego entre elas, de maneira que elas possam se comunicar como se estivessem na mesma rede. uma conexão de pareamento da VPC pode ajudar a facilitar o acesso e a transferência dos dados.

Para estabelecer uma conexão de emparelhamento da VPC, você precisa autorizar duas Contas da AWS separadas dentro de uma única pilha do CloudFormation.

Consulte mais informações sobre o emparelhamento de VPCs e suas limitações no [Guia de emparelhamento da Amazon VPC](#).

Pré-requisitos

1. Você precisa de um ID da VPC pareada, um ID de Conta da AWS e uma [função de acesso de conta cruzada](#) para a conexão de pareamento.

Note

Esta descrição se refere a duas contas: a primeira é uma conta que permite o pareamento de contas cruzadas (a conta de aceitação). A segunda é uma conta que solicita a conexão de pareamento (a conta solicitante).

2. Para aceitar a conexão de pareamento da VPC, a função de acesso de conta cruzada deve ser assumível por você. O recurso se comporta da mesma maneira que um recurso de conexão de pareamento da VPC na mesma conta. Para obter mais informações sobre como um administrador do IAM concede aos usuários permissões para assumir o perfil entre contas, consulte [Conceder permissões a um usuário para alternar perfis](#) no Guia do usuário do IAM.

Etapa 1: Criar uma VPC e uma função de conta cruzada

Nesta etapa, você criará a VPC e a função na conta de aceitação.

Para criar uma VPC e um perfil de acesso entre contas

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, escolha Criar pilha no canto superior direito e depois Com novos recursos (padrão).
3. Em Pré-requisito: preparar modelo, escolha Escolher um modelo existente e depois Carregar um arquivo de modelo, Escolher arquivo.
4. Abra um editor de texto na máquina local e adicione um dos modelos a seguir. Salve o arquivo e volte ao console para selecioná-lo como o arquivo de modelo.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.1.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
```

```

        {
            "Principal": {
                "AWS": {
                    "Ref": "PeerRequesterAccountId"
                }
            },
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow"
        }
    ],
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "root",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "ec2:AcceptVpcPeeringConnection",
                        "Resource": "*"
                    }
                ]
            }
        }
    ]
},
"Outputs": {
    "VPCId": {
        "Value": {
            "Ref": "vpc"
        }
    },
    "RoleARN": {
        "Value": {
            "Fn::GetAtt": [
                "peerRole",
                "Arn"
            ]
        }
    }
}

```

```
    }  
  }  
}  
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09  
Description: Create a VPC and an assumable role for cross account VPC peering.  
Parameters:  
  PeerRequesterAccountId:  
    Type: String  
Resources:  
  vpc:  
    Type: AWS::EC2::VPC  
    Properties:  
      CidrBlock: 10.1.0.0/16  
      EnableDnsSupport: false  
      EnableDnsHostnames: false  
      InstanceTenancy: default  
  peerRole:  
    Type: AWS::IAM::Role  
    Properties:  
      AssumeRolePolicyDocument:  
        Statement:  
          - Principal:  
              AWS: !Ref PeerRequesterAccountId  
            Action:  
              - 'sts:AssumeRole'  
            Effect: Allow  
      Path: /  
      Policies:  
        - PolicyName: root  
          PolicyDocument:  
            Version: 2012-10-17  
            Statement:  
              - Effect: Allow  
                Action: 'ec2:AcceptVpcPeeringConnection'  
                Resource: '*'  
Outputs:  
  VPCId:  
    Value: !Ref vpc  
  RoleARN:
```

```
Value: !GetAtt
  - peerRole
  - Arn
```

5. Escolha Próximo.
6. Dê um nome à pilha (por exemplo, **VPC-owner**) e insira o ID da Conta da AWS; da conta solicitante no campo PeerRequesterAccountId.
7. Aceite os padrões e escolha Próximo.
8. Escolha Reconheço que o CloudFormation pode criar recursos do IAM e escolha Criar pilha.

Etapa 2: Criar um modelo que inclua **AWS::EC2::VPCPeeringConnection**

Agora que já criou a VPC e a função de conta cruzada, você pode parear com a VPC usando outra Conta da AWS (a conta solicitante).

Para criar um modelo que inclua o recurso [AWS::EC2::VPCPeeringConnection](#)

1. Volte para a página inicial do console do CloudFormation.
2. Na página Pilhas, escolha Criar pilha no canto superior direito e depois Com novos recursos (padrão).
3. Em Pré-requisito: preparar modelo, escolha Escolher um modelo existente e depois Carregar um arquivo de modelo, Escolher arquivo.
4. Abra um editor de texto na máquina local e adicione um dos modelos a seguir. Salve o arquivo e volte ao console para selecioná-lo como o arquivo de modelo.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and a VPC Peering connection using the PeerRole to accept.",
  "Parameters": {
    "PeerVPCAccountId": {
      "Type": "String"
    },
    "PeerVPCId": {
      "Type": "String"
    },
    "PeerRoleArn": {
```

```
    "Type": "String"
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.2.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "vpcPeeringConnection": {
      "Type": "AWS::EC2::VPCPeeringConnection",
      "Properties": {
        "VpcId": {
          "Ref": "vpc"
        },
        "PeerVpcId": {
          "Ref": "PeerVPCId"
        },
        "PeerOwnerId": {
          "Ref": "PeerVPCAccountId"
        },
        "PeerRoleArn": {
          "Ref": "PeerRoleArn"
        }
      }
    }
  },
  "Outputs": {
    "VPCId": {
      "Value": {
        "Ref": "vpc"
      }
    },
    "VPCPeeringConnectionId": {
      "Value": {
        "Ref": "vpcPeeringConnection"
      }
    }
  }
}
```



```
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and a VPC Peering connection using the PeerRole to
  accept.
Parameters:
  PeerVPCAccountId:
    Type: String
  PeerVPCId:
    Type: String
  PeerRoleArn:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.2.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  vpcPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref vpc
      PeerVpcId: !Ref PeerVPCId
      PeerOwnerId: !Ref PeerVPCAccountId
      PeerRoleArn: !Ref PeerRoleArn
Outputs:
  VPCId:
    Value: !Ref vpc
  VPCPeeringConnectionId:
    Value: !Ref vpcPeeringConnection
```

5. Escolha Próximo.
6. Dê um nome à pilha (por exemplo, **VPC-peering-connection**).
7. Aceite os padrões e escolha Próximo.
8. Escolha Reconheço que o CloudFormation pode criar recursos do IAM e escolha Criar pilha.

Criar um modelo com uma política altamente restritiva

Convém criar uma política de pareamento altamente restritiva para a VPC usando outra Conta da AWS.

O modelo de exemplo a seguir mostra como alterar o modelo do proprietário pareado da VPC (a conta de aceitação criada na Etapa 1 acima), de maneira que ele seja mais restritivo.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  },
  "Resources": {
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Action": [
                "sts:AssumeRole"
              ],
              "Effect": "Allow",
              "Principal": {
                "AWS": {
                  "Ref": "PeerRequesterAccountId"
                }
              }
            }
          ]
        }
      }
    },
    "Path": "/",
    "Policies": [
      {
        "PolicyDocument": {
          "Statement": [
            {

```

```

        "Action": "ec2:acceptVpcPeeringConnection",
        "Effect": "Allow",
        "Resource": {
            "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}"
        }
    },
    {
        "Action": "ec2:acceptVpcPeeringConnection",
        "Condition": {
            "StringEquals": {
                "ec2:AcceptorVpc": {
                    "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}"
                }
            }
        },
        "Effect": "Allow",
        "Resource": {
            "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-
peering-connection/*"
        }
    }
],
"Version": "2012-10-17"
},
"PolicyName": "root"
}
]
}
},
"vpc": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
        "CidrBlock": "10.1.0.0/16",
        "EnableDnsHostnames": false,
        "EnableDnsSupport": false,
        "InstanceTenancy": "default"
    }
}
},
"Outputs": {
    "RoleARN": {
        "Value": {
            "Fn::GetAtt": [

```

```

        "peerRole",
        "Arn"
    ]
}
},
"VPCId":{
    "Value":{
        "Ref":"vpc"
    }
}
}
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Action:
              - 'sts:AssumeRole'
            Effect: Allow
            Principal:
              AWS:
                Ref: PeerRequesterAccountId
      Path: /
    Policies:
      - PolicyDocument:
          Statement:
            - Action: 'ec2:acceptVpcPeeringConnection'
              Effect: Allow
              Resource:
                'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}'
            - Action: 'ec2:acceptVpcPeeringConnection'
              Condition:
                StringEquals:

```

```

        'ec2:AccepterVpc':
            'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}'
            Effect: Allow
            Resource:
                'Fn::Sub': >-
                    arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-peering-
connection/*
            Version: 2012-10-17
            PolicyName: root
    vpc:
        Type: AWS::EC2::VPC
        Properties:
            CidrBlock: 10.1.0.0/16
            EnableDnsHostnames: false
            EnableDnsSupport: false
            InstanceTenancy: default
    Outputs:
        RoleARN:
            Value:
                'Fn::GetAtt':
                    - peerRole
                    - Arn
        VPCId:
            Value:
                Ref: vpc

```

Para acessar o VPC, você pode usar o mesmo modelo solicitante como na Etapa 2 acima.

Para obter mais informações, consulte [Identity and Access Management para emparelhamento de VPC](#) no Guia de emparelhamento de Amazon VPC.

Executar implantações azuis/verdes do ECS por meio do CodeDeploy usando o CloudFormation

Para atualizar uma aplicação em execução no Amazon Elastic Container Service (Amazon ECS), é possível usar uma estratégia de implantação azul/verde do CodeDeploy. Essa estratégia ajuda a minimizar as interrupções causadas pela alteração das versões da aplicação.

Em uma implantação azul/verde, você cria um novo ambiente de aplicação (chamado de verde) junto com seu ambiente ativo atual (chamado de azul). Isso permite que você monitore e teste o ambiente verde antes de encaminhar o tráfego ao vivo do ambiente azul para o ambiente verde.

Depois que o ambiente verde estiver servindo tráfego ao vivo, você poderá encerrar o ambiente azul com segurança.

Para permitir que o CloudFormation execute implantações azuis/verdes em uma pilha, inclua as seguintes informações no modelo de pilha:

- Uma seção `Hooks` que descreve um hook `AWS::CodeDeploy::BlueGreen`.
- Uma seção `Transform` que especifica a transformação `AWS::CodeDeployBlueGreen`.

Os tópicos a seguir orientam você ao longo da configuração de um modelo do CloudFormation para uma implantação azul/verde no ECS.

Tópicos

- [Sobre implantações azuis/verdes](#)
- [Considerações ao gerenciar implantações azul/verde do ECS usando o CloudFormation](#)
- [Sintaxe do hook `AWS::CodeDeploy::BlueGreen`](#)
- [Exemplo de modelo de implantação azul/verde](#)

Sobre implantações azuis/verdes

Este tópico fornece uma visão geral de como realizar implantações azuis/verdes com o CloudFormation. Ele também explica como preparar seu modelo do CloudFormation para implantações azuis/verdes.

Tópicos

- [Como funciona](#)
- [Atualizações de recursos que iniciam implantações verdes](#)
- [Preparar seu modelo para executar implantações azul/verde do ECS](#)
- [Modelar sua implantação azul/verde usando recursos do CloudFormation](#)
- [Conjuntos de alterações](#)
- [Monitorar eventos da pilha](#)
- [Permissões do IAM para implantações azuis/verdes](#)

Como funciona

Ao usar o CloudFormation para executar implantações azuis/verdes do ECS via CodeDeploy, você começa criando um modelo de pilha que define os recursos para seus ambientes de aplicativo azul e verde, incluindo a especificação das configurações de roteamento de tráfego e de estabilização a serem usadas. Em seguida, você cria uma pilha com base nesse modelo. Isso gera sua aplicação azul (atual). O CloudFormation cria somente os recursos azuis durante a criação da pilha. Os recursos para uma implantação verde não são criados até que sejam necessários.

Depois, se em uma futura atualização de pilha você atualizar os recursos de definição de tarefa ou do conjunto de tarefas no aplicativo azul, o CloudFormation fará o seguinte:

- Gerará todos os recursos de ambiente do aplicativo verde necessários
- Deslocará o tráfego com base nos parâmetros de roteamento de tráfego especificados
- Excluirá os recursos azuis

Se ocorrer um erro em qualquer ponto antes da implantação verde ser finalizada com êxito, o CloudFormation reverterá a pilha para seu estado anterior ao início de toda a implantação verde.

Atualizações de recursos que iniciam implantações verdes

Quando você executa uma atualização de pilha que atualiza determinadas propriedades de recursos do ECS, o CloudFormation inicia um processo de implantação verde. Os recursos que iniciam esse processo são:

- [AWS::ECS::TaskDefinition](#)
- [AWS::ECS::TaskSet](#)

No entanto, se as atualizações desses recursos não envolverem alterações de propriedade que exijam substituição, uma implantação ecológica não será iniciada. Para obter mais informações, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

É importante observar que não é possível combinar atualizações para os recursos acima com atualizações para outros recursos na mesma operação de atualização de pilha. Se precisar atualizar os recursos listados e outros recursos na mesma pilha, você terá duas opções:

- Execute duas operações de atualização de pilha separadas: uma que inclua somente as atualizações para os recursos acima e uma atualização de pilha separada que inclua alterações em quaisquer outros recursos.

- Remova as seções Transform e Hooks do modelo e execute a atualização da pilha. Neste caso, o CloudFormation não executará uma implantação verde.

Preparar seu modelo para executar implantações azul/verde do ECS

Para permitir implantações azul/verde em sua pilha, inclua as seções a seguir no modelo de pilha antes de executar uma atualização dela.

- Adicione uma referência à transformação AWS::CodeDeployBlueGreen ao modelo:

```
"Transform": [  
  "AWS::CodeDeployBlueGreen"  
],
```

- Adicione uma seção Hooks que invoque o hook AWS::CodeDeploy::BlueGreen e especifique as propriedades para a implantação. Para obter mais informações, consulte [Sintaxe do hook AWS::CodeDeploy::BlueGreen](#).
- Na seção Resources, defina os recursos azuis e verdes para a implantação.

É possível adicionar essas seções ao criar o modelo pela primeira vez (ou seja, antes de criar a pilha) ou adicioná-las a um modelo existente antes de executar uma atualização de pilha. Se você especificar a implantação azul/verde para uma nova pilha, o CloudFormation criará somente os recursos azuis durante a criação. Os recursos para a implantação verde não serão criados até que sejam necessários durante uma atualização da pilha.

Modelar sua implantação azul/verde usando recursos do CloudFormation

Para executar a implantação azul/verde do CodeDeploy no ECS, o modelo do CloudFormation deverá incluir os recursos que modelam sua implantação, como um serviço do Amazon ECS e um balanceador de carga. Para mais detalhes sobre o que esses recursos representam, consulte [Antes de iniciar uma implantação do Amazon ECS](#) no Guia do usuário do AWS CodeDeploy.

Requisito	Recurso	Obrigatório/opcional	Inicia a implantação azul/verde se substituído?
Cluster do Amazon ECS	AWS::ECS::Cluster	Opcional. O cluster padrão pode ser usado.	Não
Serviço do Amazon ECS	AWS::ECS::Service	Obrigatório.	Não
Application ou Network Load Balancer	AWS::ECS::ServiceLoadBalancer	Obrigatório.	Não
Listener de produção	AWS::ElasticLoadBalancingV2::Listener	Obrigatório.	Não
Listener de teste	AWS::ElasticLoadBalancingV2::Listener	Opcional.	Não
Dois grupos de destino	AWS::ElasticLoadBalancingV2::TargetGroup	Obrigatório.	Não
Definição de tarefa do Amazon ECS	AWS::ECS::TaskDefinition	Obrigatório.	Sim
Contêiner para seu aplicativo do Amazon ECS	AWS::ECS::TaskDefinition Container Definition Name	Obrigatório.	Não
Porta para o conjunto de tarefas de substituição	AWS::ECS::TaskDefinition PortMapping ContainerPort	Obrigatório.	Não

Conjuntos de alterações

É altamente recomendável que você crie um conjunto de alterações antes de executar uma atualização de pilha que iniciará uma implantação verde. Isso permite que você veja as alterações que serão feitas na pilha antes da execução da atualização da pilha. Esteja ciente de que as alterações de recursos podem não ser listadas na ordem em que serão executadas durante a atualização da pilha. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

Monitorar eventos da pilha

É possível visualizar os eventos da pilha gerados em cada etapa da implantação do ECS na guia Events (Eventos) da página Stack (Pilha) e usando a AWS CLI. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Permissões do IAM para implantações azuis/verdes

Para que o CloudFormation execute com sucesso as implantações azul/verde, você deve ter as seguintes permissões do CodeDeploy:

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

Para obter mais informações, consulte [Ações, recursos e chaves de condição do CodeDeploy](#) na Referência de autorização do serviço.

Considerações ao gerenciar implantações azul/verde do ECS usando o CloudFormation

O processo de usar o CloudFormation para realizar implantações azuis/verdes do ECS por meio do CodeDeploy é diferente de uma implantação padrão do ECS que usa somente o CodeDeploy. Para obter uma compreensão detalhada dessas diferenças, consulte [Diferenças entre implantações azuis/verdes do Amazon ECS via CodeDeploy e CloudFormation](#) no Guia do usuário do AWS CodeDeploy.

Ao gerenciar sua implantação azul/verde usando o CloudFormation, há certas limitações e considerações que você deve ter em mente:

- Somente atualizações de determinados recursos acionarão uma implantação verde. Para obter mais informações, consulte [Atualizações de recursos que iniciam implantações verdes](#).

- Não é possível incluir atualizações de recursos que iniciam implantações verdes e atualizações de outros recursos na mesma atualização de pilha. Para obter mais informações, consulte [Atualizações de recursos que iniciam implantações verdes](#).
- Só é possível especificar um único serviço do ECS como o destino de implantação.
- Os parâmetros cujos valores forem ofuscados pelo CloudFormation não poderão ser atualizados pelo CodeDeploy durante uma implantação verde e causarão um erro e uma falha na atualização da pilha. Isso inclui:
 - Parâmetros definidos com o atributo NoEcho.
 - Parâmetros que usam referências dinâmicas para recuperar seus valores de serviços externos. Para obter mais informações sobre referências dinâmicas, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).
- Para cancelar uma implantação verde que ainda está em andamento, cancele a atualização da pilha no CloudFormation, e não no CodeDeploy ou ECS. Para obter mais informações, consulte [Cancelar uma atualização de pilha](#). Após a conclusão de uma atualização, não será possível cancelá-la. No entanto, é possível atualizar uma pilha novamente com quaisquer configurações anteriores.
- No momento, os seguintes recursos do CloudFormation não são compatíveis com modelos que definem implantações azuis/verdes do ECS:
 - Declarar [saídas](#) ou usar [Fn::ImportValue](#) para importar valores de outras pilhas.
 - Importar recursos. Para obter mais informações sobre a importação de recursos, consulte [Importar recursos da AWS para uma pilha do CloudFormation](#).
 - Usar o hook `AWS::CodeDeploy::BlueGreen` em um modelo que inclua recursos de pilha aninhados. Para saber mais sobre pilhas aninhadas, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).
 - Usar o hook `AWS::CodeDeploy::BlueGreen` em uma pilha aninhada.

Sintaxe do hook **AWS::CodeDeploy::BlueGreen**

A sintaxe a seguir descreve a estrutura de um hook `AWS::CodeDeploy::BlueGreen` para implantações azuis/verdes do ECS.

Sintaxe

```
"Hooks": {  
  "Logical ID": {
```

```

    "Type": "AWS::CodeDeploy::BlueGreen",
    "Properties": {
      "TrafficRoutingConfig": {
        "Type": "Traffic routing type",
        "TimeBasedCanary": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        },
        "TimeBasedLinear": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        }
      },
      "AdditionalOptions": {"TerminationWaitTimeInMinutes": Integer},
      "LifecycleEventHooks": {
        "BeforeInstall": "FunctionName",
        "AfterInstall": "FunctionName",
        "AfterAllowTestTraffic": "FunctionName",
        "BeforeAllowTraffic": "FunctionName",
        "AfterAllowTraffic": "FunctionName"
      },
      "ServiceRole": "CodeDeployServiceRoleName",
      "Applications": [
        {
          "Target": {
            "Type": "AWS::ECS::Service",
            "LogicalID": "Logical ID of AWS::ECS::Service"
          },
          "ECSAttributes": {
            "TaskDefinitions": [
              "Logical ID of AWS::ECS::TaskDefinition (Blue)",
              "Logical ID of AWS::ECS::TaskDefinition (Green)"
            ],
            "TaskSets": [
              "Logical ID of AWS::ECS::TaskSet (Blue)",
              "Logical ID of AWS::ECS::TaskSet (Green)"
            ],
            "TrafficRouting": {
              "ProdTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
                "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener  
(Production)"
              },
              "TestTrafficRoute": {

```

```
(Test)"
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "LogicalID": ""Logical ID of AWS::ElasticLoadBalancingV2::Listener"",
  },
  "TargetGroups": [
    ""Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Blue)"",
    ""Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Green)""
  ]
}
}
```

Propriedades

ID lógico (também chamado de nome lógico)

O ID lógico de um hook declarado na seção Hooks do modelo. O ID lógico deve ser alfanumérico (A-Z a-z 0-9) e exclusivo no modelo.

Obrigatório: Sim

Type

O tipo do hook. `AWS::CodeDeploy::BlueGreen`

Obrigatório: Sim

Properties

Propriedades do hook.

Obrigatório: Sim

TrafficRoutingConfig

Definição das configurações de roteamento de tráfego.

Obrigatório: não

A configuração padrão é o deslocamento de tráfego canário baseado em tempo, com uma porcentagem de 15% de etapa e um tempo de incorporação de cinco minutos.

Type

O tipo de deslocamento de tráfego usado pela configuração de implantação.

Valores válidos: AllAtOnce | TimeBasedCanary | TimeBasedLinear

Obrigatório: Sim

TimeBasedCanary

Especifica uma configuração que desloca o tráfego de uma versão da implantação para outra em incrementos de dois.

Necessário: Condicional: se você especificar TimeBasedCanary como o tipo de roteamento de tráfego, deverá incluir o parâmetro TimeBasedCanary.

StepPercentage

A porcentagem de tráfego a ser deslocado no primeiro incremento de uma implantação TimeBasedCanary. A porcentagem da etapa deve ser 14% ou maior.

Obrigatório: não

BakeTimeMins

O número de minutos entre o primeiro e o segundo deslocamento de tráfego de uma implantação TimeBasedCanary.

Obrigatório: não

TimeBasedLinear

Especifica uma configuração que desloca o tráfego de uma versão da implantação para outra em incrementos iguais, com um número igual de minutos entre cada incremento.

Necessário: Condicional: se você especificar TimeBasedLinear como o tipo de roteamento de tráfego, deverá incluir o parâmetro TimeBasedLinear.

StepPercentage

A porcentagem de tráfego deslocado no início de cada incremento de uma implantação TimeBasedLinear. A porcentagem da etapa deve ser 14% ou maior.

Obrigatório: não

`BakeTimeMins`

O número de minutos entre cada deslocamento de tráfego incremental de uma implantação `TimeBasedLinear`.

Obrigatório: não

`AdditionalOptions`

Opções adicionais para a implantação azul/verde.

Obrigatório: não

`TerminationWaitTimeInMinutes`

Especifica o tempo de espera, em minutos, antes de encerrar os recursos azuis.

Obrigatório: não

`LifecycleEventHooks`

Use hooks de eventos de ciclo de vida para especificar uma função do Lambda que o CodeDeploy possa chamar para validar uma implantação. É possível usar a mesma função ou uma diferente para os eventos de ciclo de vida de implantação. Após a conclusão dos testes de validação, a função do Lambda `AfterAllowTraffic` chama de volta o CodeDeploy e entrega um resultado de `Succeeded` ou `Failed`. Para mais informações, consulte a [seção “Hooks” do AppSpec](#) no Guia do usuário do AWS CodeDeploy.

Obrigatório: não

`BeforeInstall`

Função a ser usada para executar tarefas antes que o conjunto de tarefas de substituição seja criado.

Obrigatório: não

`AfterInstall`

Função a ser usada para executar tarefas depois que o conjunto de tarefas de substituição for criado e um dos grupos de destino for associado a ele.

Obrigatório: não

AfterAllowTestTraffic

Função a ser usada para executar tarefas depois que o listener de teste distribuir o tráfego para o conjunto de tarefas de substituição.

Obrigatório: não

BeforeAllowTraffic

Função a ser usada para executar tarefas depois que o segundo grupo de destino for associado ao conjunto de tarefas de substituição, mas antes que o tráfego seja deslocado para o conjunto de tarefas de substituição.

Obrigatório: não

AfterAllowTraffic

Função a ser usada para executar tarefas depois que o segundo grupo de destino distribuir o tráfego para o conjunto de tarefas de substituição.

Obrigatório: não

ServiceRole

A função de execução a ser usada pelo CloudFormation para executar as implantações azul/verde. Para obter uma lista das permissões necessárias, consulte [Permissões do IAM para implantações azuis/verdes](#).

Obrigatório: não

Applications

Especifica as propriedades do aplicativo do Amazon ECS.

Obrigatório: sim

Target

Obrigatório: sim

Type

O tipo de recurso.

Obrigatório: sim

LogicalID

O ID lógico do recurso.

Obrigatório: Sim

ECSAttributes

Os recursos que representam os vários requisitos de sua implantação de aplicativos do Amazon ECS.

Obrigatório: Sim

TaskDefinitions

O ID lógico do recurso [AWS::ECS::TaskDefinition](#) para executar o contêiner do Docker que contém sua aplicação do Amazon ECS.

Obrigatório: sim

TaskSets

Os IDs lógicos dos recursos [AWS::ECS::TaskSet](#) a serem usados como conjuntos de tarefas para a aplicação.

Obrigatório: sim

TrafficRouting

Especifica os recursos usados para roteamento de tráfego.

Obrigatório: sim

ProdTrafficRoute

O listener é usado pelo load balancer para direcionar o tráfego para seus grupos de destino.

Obrigatório: sim

Type

O tipo do recurso. `AWS::ElasticLoadBalancingV2::Listener`

Obrigatório: sim

`LogicalID`

O ID lógico do recurso.

Obrigatório: sim

`TestTrafficRoute`

O listener é usado pelo load balancer para direcionar o tráfego para seus grupos de destino.

Obrigatório: sim

`Type`

O tipo do recurso. `AWS::ElasticLoadBalancingV2::Listener`

Obrigatório: sim

`LogicalID`

O ID lógico do recurso.

Obrigatório: não

`TargetGroups`

ID lógico dos recursos a serem usados como grupos de destino para rotear o tráfego para o destino registrado.

Obrigatório: sim

Exemplo de modelo de implantação azul/verde

O modelo de exemplo a seguir configura uma implantação azul/verde do CodeDeploy no ECS com um progresso de roteamento de tráfego de 15% por etapa e um período de estabilização de 5 minutos entre cada etapa.

Criar uma pilha com o modelo provisionará a configuração inicial da implantação. Se você fez alguma alteração nas propriedades do recurso `BlueTaskSet` que exija que esse recurso seja substituído, o CloudFormation inicializará uma implantação verde como parte da atualização da pilha.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": { "Type": "AWS::EC2::VPC::Id" },
    "Subnet1": { "Type": "AWS::EC2::Subnet::Id" },
    "Subnet2": { "Type": "AWS::EC2::Subnet::Id" }
  },
  "Transform": [ "AWS::CodeDeployBlueGreen" ],
  "Hooks": {
    "CodeDeployBlueGreenHook": {
      "Type": "AWS::CodeDeploy::BlueGreen",
      "Properties": {
        "TrafficRoutingConfig": {
          "Type": "TimeBasedCanary",
          "TimeBasedCanary": {
            "StepPercentage": 15,
            "BakeTimeMins": 5
          }
        }
      },
    },
    "Applications": [
      {
        "Target": {
          "Type": "AWS::ECS::Service",
          "LogicalID": "ECSDemoService"
        },
        "ECSAttributes": {
          "TaskDefinitions": [ "BlueTaskDefinition", "GreenTaskDefinition" ],
          "TaskSets": [ "BlueTaskSet", "GreenTaskSet" ],
          "TrafficRouting": {
            "ProdTrafficRoute": {
              "Type": "AWS::ElasticLoadBalancingV2::Listener",
              "LogicalID": "ALBListenerProdTraffic"
            },
            "TargetGroups": [ "ALBTargetGroupBlue", "ALBTargetGroupGreen" ]
          }
        }
      }
    ]
  },
  "Resources": {
```

```
"ExampleSecurityGroup":{
  "Type":"AWS::EC2::SecurityGroup",
  "Properties":{
    "GroupDescription":"Security group for ec2 access",
    "VpcId":{"Ref":"Vpc" },
    "SecurityGroupIngress":[
      {
        "IpProtocol":"tcp",
        "FromPort":80,
        "ToPort":80,
        "CidrIp":"0.0.0.0/0"
      },
      {
        "IpProtocol":"tcp",
        "FromPort":8080,
        "ToPort":8080,
        "CidrIp":"0.0.0.0/0"
      },
      {
        "IpProtocol":"tcp",
        "FromPort":22,
        "ToPort":22,
        "CidrIp":"0.0.0.0/0"
      }
    ]
  }
},
"ALBTargetGroupBlue":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{"HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{"Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{"Ref":"Vpc" }
  }
}
```

```

},
"ALBTargetGroupGreen":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{ "HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{ "Ref":"Vpc" }
  }
},
"ExampleALB":{
  "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties":{
    "Scheme":"internet-facing",
    "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
    "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }],
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "Type":"application",
    "IpAddressType":"ipv4"
  }
},
"ALBListenerProdTraffic":{
  "Type":"AWS::ElasticLoadBalancingV2::Listener",
  "Properties":{
    "DefaultActions":[
      {
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {
              "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
              "Weight":1
            }
          ]
        }
      }
    ]
  }
}

```

```

    }
  ],
  "LoadBalancerArn":{ "Ref":"ExampleALB" },
  "Port":80,
  "Protocol":"HTTP"
}
},
"ALBListenerProdRule":{
  "Type":"AWS::ElasticLoadBalancingV2::ListenerRule",
  "Properties":{
    "Actions":[
      {
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {
              "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
              "Weight":1
            }
          ]
        }
      }
    ],
    "Conditions":[
      {
        "Field":"http-header",
        "HttpHeaderConfig":{
          "HttpHeaderName":"User-Agent",
          "Values":[ "Mozilla" ]
        }
      }
    ]
  },
  "ListenerArn":{ "Ref":"ALBListenerProdTraffic" },
  "Priority":1
}
},
"ECSTaskExecutionRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "AssumeRolePolicyDocument":{
      "Version": "2012-10-17",
      "Statement":[
        {
          "Sid":"","

```

```

        "Effect": "Allow",
        "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
},
    "ManagedPolicyArns": [ "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" ]
},
    "BlueTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "ExecutionRoleArn": {
                "Fn::GetAtt": [ "ECSTaskExecutionRole", "Arn" ]
            },
            "ContainerDefinitions": [
                {
                    "Name": "DemoApp",
                    "Image": "nginxdemos/hello:latest",
                    "Essential": true,
                    "PortMappings": [
                        {
                            "HostPort": 80,
                            "Protocol": "tcp",
                            "ContainerPort": 80
                        }
                    ]
                }
            ]
        }
    },
    "RequiresCompatibilities": [ "FARGATE" ],
    "NetworkMode": "awsvpc",
    "Cpu": "256",
    "Memory": "512",
    "Family": "ecs-demo"
},
    "ECSDemoCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {}
    },
    "ECSDemoService": {

```

```

    "Type": "AWS::ECS::Service",
    "Properties": {
      "Cluster": { "Ref": "ECSDemoCluster" },
      "DesiredCount": 1,
      "DeploymentController": { "Type": "EXTERNAL" }
    }
  },
  "BlueTaskSet": {
    "Type": "AWS::ECS::TaskSet",
    "Properties": {
      "Cluster": { "Ref": "ECSDemoCluster" },
      "LaunchType": "FARGATE",
      "NetworkConfiguration": {
        "AwsVpcConfiguration": {
          "AssignPublicIp": "ENABLED",
          "SecurityGroups": [ { "Ref": "ExampleSecurityGroup" } ],
          "Subnets": [ { "Ref": "Subnet1" }, { "Ref": "Subnet2" } ]
        }
      }
    },
    "PlatformVersion": "1.4.0",
    "Scale": {
      "Unit": "PERCENT",
      "Value": 100
    },
    "Service": { "Ref": "ECSDemoService" },
    "TaskDefinition": { "Ref": "BlueTaskDefinition" },
    "LoadBalancers": [
      {
        "ContainerName": "DemoApp",
        "ContainerPort": 80,
        "TargetGroupArn": { "Ref": "ALBTargetGroupBlue" }
      }
    ]
  },
  "PrimaryTaskSet": {
    "Type": "AWS::ECS::PrimaryTaskSet",
    "Properties": {
      "Cluster": { "Ref": "ECSDemoCluster" },
      "Service": { "Ref": "ECSDemoService" },
      "TaskSetId": { "Fn::GetAtt": [ "BlueTaskSet", "Id" ] }
    }
  }
}

```



```
}  
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09  
Parameters:  
  Vpc:  
    Type: AWS::EC2::VPC::Id  
  Subnet1:  
    Type: AWS::EC2::Subnet::Id  
  Subnet2:  
    Type: AWS::EC2::Subnet::Id  
Transform:  
  - 'AWS::CodeDeployBlueGreen'  
Hooks:  
  CodeDeployBlueGreenHook:  
    Type: AWS::CodeDeploy::BlueGreen  
    Properties:  
      TrafficRoutingConfig:  
        Type: TimeBasedCanary  
        TimeBasedCanary:  
          StepPercentage: 15  
          BakeTimeMins: 5  
      Applications:  
        - Target:  
            Type: AWS::ECS::Service  
            LogicalID: ECSDemoService  
        ECSAttributes:  
          TaskDefinitions:  
            - BlueTaskDefinition  
            - GreenTaskDefinition  
          TaskSets:  
            - BlueTaskSet  
            - GreenTaskSet  
          TrafficRouting:  
            ProdTrafficRoute:  
              Type: AWS::ElasticLoadBalancingV2::Listener  
              LogicalID: ALBListenerProdTraffic  
            TargetGroups:  
              - ALBTargetGroupBlue  
              - ALBTargetGroupGreen  
Resources:
```

```
ExampleSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for ec2 access
    VpcId: !Ref Vpc
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 8080
        ToPort: 8080
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
ALBTargetGroupBlue:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 2
    HealthyThresholdCount: 2
    Matcher:
      HttpCode: '200'
    Port: 80
    Protocol: HTTP
    Tags:
      - Key: Group
        Value: Example
    TargetType: ip
    UnhealthyThresholdCount: 4
    VpcId: !Ref Vpc
ALBTargetGroupGreen:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
```

```
HealthCheckTimeoutSeconds: 2
HealthyThresholdCount: 2
Matcher:
  HttpStatusCode: '200'
Port: 80
Protocol: HTTP
Tags:
  - Key: Group
    Value: Example
TargetType: ip
UnhealthyThresholdCount: 4
VpcId: !Ref Vpc
ExampleALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    SecurityGroups:
      - !Ref ExampleSecurityGroup
    Subnets:
      - !Ref Subnet1
      - !Ref Subnet2
    Tags:
      - Key: Group
        Value: Example
    Type: application
    IpAddressType: ipv4
ALBListenerProdTraffic:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref ALBTargetGroupBlue
              Weight: 1
    LoadBalancerArn: !Ref ExampleALB
    Port: 80
    Protocol: HTTP
ALBListenerProdRule:
  Type: AWS::ElasticLoadBalancingV2::ListenerRule
  Properties:
    Actions:
      - Type: forward
        ForwardConfig:
```

```
    TargetGroups:
      - TargetGroupArn: !Ref ALBTargetGroupBlue
        Weight: 1
    Conditions:
      - Field: http-header
        HttpHeaderConfig:
          HttpHeaderName: User-Agent
          Values:
            - Mozilla
    ListenerArn: !Ref ALBListenerProdTraffic
    Priority: 1
  ECSTaskExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Sid: ''
            Effect: Allow
            Principal:
              Service: ecs-tasks.amazonaws.com
            Action: 'sts:AssumeRole'
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy'
  BlueTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      ExecutionRoleArn: !GetAtt
        - ECSTaskExecutionRole
        - Arn
      ContainerDefinitions:
        - Name: DemoApp
          Image: 'nginxdemos/hello:latest'
          Essential: true
          PortMappings:
            - HostPort: 80
              Protocol: tcp
              ContainerPort: 80
      RequiresCompatibilities:
        - FARGATE
      NetworkMode: awsvpc
      Cpu: '256'
      Memory: '512'
      Family: ecs-demo
```

```
ECSDemoCluster:
  Type: AWS::ECS::Cluster
  Properties: {}
ECSDemoService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref ECSDemoCluster
    DesiredCount: 1
    DeploymentController:
      Type: EXTERNAL
BlueTaskSet:
  Type: AWS::ECS::TaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups:
          - !Ref ExampleSecurityGroup
        Subnets:
          - !Ref Subnet1
          - !Ref Subnet2
    PlatformVersion: 1.4.0
    Scale:
      Unit: PERCENT
      Value: 100
    Service: !Ref ECSDemoService
    TaskDefinition: !Ref BlueTaskDefinition
    LoadBalancers:
      - ContainerName: DemoApp
        ContainerPort: 80
        TargetGroupArn: !Ref ALBTargetGroupBlue
PrimaryTaskSet:
  Type: AWS::ECS::PrimaryTaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    Service: !Ref ECSDemoService
    TaskSetId: !GetAtt
      - BlueTaskSet
      - Id
```

Trechos de modelo do CloudFormation

Esta seção fornece uma série de exemplos de cenários que você pode usar para entender como declarar várias partes de modelo do CloudFormation. Você também pode usar os trechos como ponto de partida para seções de seus modelos personalizados.

Note

Como os modelos do CloudFormation devem ser compatíveis com JSON, não há provisão para um caractere de continuação de linha. A quebra dos trechos desse documento poderá ser aleatória se a linha tiver mais de 68 caracteres.

Tópicos

- [Trechos de modelos gerais](#)
- [Snippets de modelos do CloudFormation para ajuste de escala automático](#)
- [Snippets de modelo do Console de faturamento da AWS](#)
- [Trechos de modelo do CloudFormation](#)
- [Trechos de modelo do Amazon CloudFront](#)
- [Trechos de modelo do Amazon CloudWatch](#)
- [Trechos de modelo do Amazon CloudWatch Logs](#)
- [Trechos de modelo do Amazon DynamoDB](#)
- [Snippets de modelo do CloudFormation do Amazon EC2](#)
- [Modelos de exemplo do Amazon Elastic Container Service](#)
- [Modelo de exemplo do Amazon Elastic File System](#)
- [Trechos de modelo do Elastic Beanstalk](#)
- [Trechos de modelo do Elastic Load Balancing](#)
- [Trechos de modelo do AWS Identity and Access Management](#)
- [AWS LambdaModelo](#)
- [Trechos de modelo do Amazon Redshift](#)
- [Trechos de modelo do Amazon RDS](#)
- [Trechos de modelos do Route 53](#)

- [Trechos de modelo do Amazon S3](#)
- [Trechos de modelo do Amazon SNS](#)
- [Trechos de modelo do Amazon SQS](#)
- [Trechos de modelo do Amazon Timestream](#)

Trechos de modelos gerais

Os exemplos a seguir mostram diferentes recursos de modelos do CloudFormation que não são específicos de um serviço da AWS

Tópicos

- [Propriedade UserData codificada em Base64](#)
- [Propriedade UserData codificada em Base64 com AccessKey e SecretKey](#)
- [Seção Parameters com um parâmetro de string literal](#)
- [Seção Parameters com parâmetro string com restrição de expressão regular](#)
- [Seção Parameters com parâmetros numéricos com restrições MinValue e MaxValue](#)
- [Seção Parameters com parâmetro numérico com a restrição AllowedValues](#)
- [Seção Parameters com um parâmetro CommaDelimitedList literal](#)
- [Seção Parameters com valor de parâmetro baseado em pseudoparâmetro](#)
- [Seção Mapping com três mapeamentos](#)
- [Description com base em string literal](#)
- [Seção Outputs com uma saída de string literal](#)
- [Seção Outputs com uma saída de referência de recurso e uma saída de referência de pseudoparâmetro](#)
- [Seção Outputs com uma saída com base em uma função, uma string literal, uma referência e um pseudoparâmetro](#)
- [Versão de formato do modelo](#)
- [Propriedade Tags da AWS](#)

Propriedade UserData codificada em Base64

Este exemplo mostra a montagem de uma propriedade UserData usando as funções `Fn::Base64` e `Fn::Join`. As referências `MyValue` e `MyName` são parâmetros que devem ser definidos na seção

Parameters do modelo. A sequência literal `Hello World` é apenas outro valor que este exemplo passa como parte do `UserData`.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ ",", [
      { "Ref" : "MyValue" },
      { "Ref" : "MyName" },
      "Hello World" ] ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    Ref: MyValue
    Ref: MyName
    Hello World
```

Propriedade `UserData` codificada em Base64 com `AccessKey` e `SecretKey`

Este exemplo mostra a montagem de uma propriedade `UserData` usando as funções `Fn::Base64` e `Fn::Join`. Inclui as informações de `AccessKey` e `SecretKey`. As referências `AccessKey` e `SecretKey` são parâmetros que devem ser definidos na seção `Parameters` do modelo.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ "", [
      "ACCESS_KEY=", { "Ref" : "AccessKey" },
      "SECRET_KEY=", { "Ref" : "SecretKey" } ] ]
  }
}
```

YAML

```
UserData:
```



```
Fn::Base64: !Sub |
  ACCESS_KEY=${AccessKey}
  SECRET_KEY=${SecretKey}
```

Seção Parameters com um parâmetro de string literal

O exemplo a seguir descreve uma declaração válida da seção Parameters na qual um único parâmetro de tipo String é declarado.

JSON

```
"Parameters" : {
  "UserName" : {
    "Type" : "String",
    "Default" : "nonadmin",
    "Description" : "Assume a vanilla user if no command-line spec provided"
  }
}
```

YAML

```
Parameters:
  UserName:
    Type: String
    Default: nonadmin
    Description: Assume a vanilla user if no command-line spec provided
```

Seção Parameters com parâmetro string com restrição de expressão regular

O exemplo a seguir descreve uma declaração válida da seção Parameters na qual um único parâmetro de tipo String é declarado. O parâmetro AdminUserAccount tem um valor padrão de admin. O valor do parâmetro deve ter um tamanho mínimo de 1, um tamanho máximo de 16 e conter caracteres alfabéticos e numéricos, mas deve começar com um caractere alfabético.

JSON

```
"Parameters" : {
  "AdminUserAccount": {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "The admin account user name",
```

```
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "16",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
  }
}
```

YAML

```
Parameters:
  AdminUserAccount:
    Default: admin
    NoEcho: true
    Description: The admin account user name
    Type: String
    MinLength: 1
    MaxLength: 16
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
```

Seção Parameters com parâmetros numéricos com restrições MinValue e MaxValue

O exemplo a seguir descreve uma declaração válida da seção Parameters na qual um único parâmetro de tipo Number é declarado. O parâmetro WebServerPort tem um padrão de 80, um valor mínimo de 1 e um valor máximo de 65535.

JSON

```
"Parameters" : {
  "WebServerPort": {
    "Default": "80",
    "Description" : "TCP/IP port for the web server",
    "Type": "Number",
    "MinValue": "1",
    "MaxValue": "65535"
  }
}
```

YAML

```
Parameters:
  WebServerPort:
    Default: 80
```

```
Description: TCP/IP port for the web server
Type: Number
MinValue: 1
MaxValue: 65535
```

Seção Parameters com parâmetro numérico com a restrição AllowedValues

O exemplo a seguir descreve uma declaração válida da seção Parameters na qual um único parâmetro de tipo Number é declarado. O parâmetro WebServerPort tem um padrão de 80 e permite apenas valores de 80 e 8888.

JSON

```
"Parameters" : {
  "WebServerPortLimited": {
    "Default": "80",
    "Description" : "TCP/IP port for the web server",
    "Type": "Number",
    "AllowedValues" : ["80", "8888"]
  }
}
```

YAML

```
Parameters:
  WebServerPortLimited:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    AllowedValues:
      - 80
      - 8888
```

Seção Parameters com um parâmetro CommaDelimitedList literal

O exemplo a seguir descreve uma declaração válida da seção Parameters na qual um único parâmetro do tipo CommaDelimitedList é declarado. A propriedade NoEcho é definida como TRUE, o que mascara seu valor com asteriscos (*****) na saída describe-stacks, exceto pelas informações armazenadas nos locais especificados abaixo.

 Important

O uso do atributo NoEcho não mascara informações armazenadas no seguinte:

- A seção de modelo de Metadata. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção Metadata. Para obter mais informações, consulte [Metadados](#).
- A seção de modelo de Outputs. Para obter mais informações, consulte [Saídas](#).
- O atributo Metadata de uma definição de recurso. Para obter mais informações, consulte o atributo [Metadata](#).

É altamente recomendável não usar esses mecanismos para incluir informações confidenciais, como senhas ou segredos.

 Important

Em vez de incorporar informações confidenciais diretamente em modelos do CloudFormation, recomendamos usar os parâmetros dinâmicos no modelo da pilha para fazer referência a informações confidenciais que são armazenadas e gerenciadas de forma externa ao CloudFormation, como no AWS Systems Manager Parameter Store ou no AWS Secrets Manager.

Para obter mais informações, consulte a prática recomendada [Não incorporar credenciais nos seus modelos](#).

JSON

```
"Parameters" : {
  "UserRoles" : {
    "Type" : "CommaDelimitedList",
    "Default" : "guest,newhire",
    "NoEcho" : "TRUE"
  }
}
```

YAML

```
Parameters:
  UserRoles:
    Type: CommaDelimitedList
    Default: "guest,newhire"
    NoEcho: true
```

Seção Parameters com valor de parâmetro baseado em pseudoparâmetro

O exemplo a seguir mostra comandos nos dados do usuário do EC2 que usam os pseudoparâmetros `AWS::StackName` e `AWS::Region`. Para obter mais informações sobre pseudoparâmetros, consulte [Obter valores da AWS usando pseudoparâmetros](#).

JSON

```
"UserData"      : { "Fn::Base64" : { "Fn::Join" : ["", [
  "#!/bin/bash -xe\n",
  "yum install -y aws-cfn-bootstrap\n",

  "/opt/aws/bin/cfn-init -v ",
  "      --stack ", { "Ref" : "AWS::StackName" },
  "      --resource LaunchConfig ",
  "      --region ", { "Ref" : "AWS::Region" }, "\n",

  "/opt/aws/bin/cfn-signal -e $? ",
  "      --stack ", { "Ref" : "AWS::StackName" },
  "      --resource WebServerGroup ",
  "      --region ", { "Ref" : "AWS::Region" }, "\n"
]]}}
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    yum update -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource LaunchConfig --
region ${AWS::Region}
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource WebServerGroup
--region ${AWS::Region}
```

Seção Mapping com três mapeamentos

O exemplo a seguir descreve uma declaração válida de seção Mapping que contém três mapeamentos. O mapa, quando correspondido a uma chave de mapeamento de Stop, SlowDown ou Go fornece os valores RGB atribuídos ao atributo RGBColor correspondente.

JSON

```
"Mappings" : {
  "LightColor" : {
    "Stop" : {
      "Description" : "red",
      "RGBColor" : "RED 255 GREEN 0 BLUE 0"
    },
    "SlowDown" : {
      "Description" : "yellow",
      "RGBColor" : "RED 255 GREEN 255 BLUE 0"
    },
    "Go" : {
      "Description" : "green",
      "RGBColor" : "RED 0 GREEN 128 BLUE 0"
    }
  }
}
```

YAML

```
Mappings:
  LightColor:
    Stop:
      Description: red
      RGBColor: "RED 255 GREEN 0 BLUE 0"
    SlowDown:
      Description: yellow
      RGBColor: "RED 255 GREEN 255 BLUE 0"
    Go:
      Description: green
      RGBColor: "RED 0 GREEN 128 BLUE 0"
```

Description com base em string literal

O exemplo a seguir descreve uma declaração válida da seção `Description`, onde o valor é baseado em uma string literal. Este trecho pode ser para modelos, parâmetros, recursos, propriedades ou saídas.

JSON

```
"Description" : "Replace this value"
```

YAML

```
Description: "Replace this value"
```

Seção Outputs com uma saída de string literal

Este exemplo mostra uma atribuição de saída com base em uma sequência literal.

JSON

```
"Outputs" : {  
  "MyPhone" : {  
    "Value" : "Please call 555-5555",  
    "Description" : "A random message for aws cloudformation describe-stacks"  
  }  
}
```

YAML

```
Outputs:  
  MyPhone:  
    Value: Please call 555-5555  
    Description: A random message for aws cloudformation describe-stacks
```

Seção Outputs com uma saída de referência de recurso e uma saída de referência de pseudoparâmetro

Este exemplo mostra uma seção `Outputs` com duas atribuições de saída. Uma é baseada em um recurso, e a outra é baseada em uma pseudoreferência.

JSON

```
"Outputs" : {
  "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
  "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```

Seção Outputs com uma saída com base em uma função, uma string literal, uma referência e um pseudoparâmetro

Este exemplo mostra uma seção Outputs com uma atribuição de saída. A função Join é usada para concatenar o valor usando um sinal de porcentagem como delimitador.

JSON

```
"Outputs" : {
  "MyOutput" : {
    "Value" : { "Fn::Join" :
      [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
    }
  }
}
```

YAML

```
Outputs:
  MyOutput:
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ] ]
```

Versão de formato do modelo

O trecho a seguir descreve uma declaração válida da seção `AWSTemplateFormatVersion`.

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

Propriedade Tags da AWS

Este exemplo mostra uma propriedade Tags da AWS. Você pode especificar essa propriedade na seção Properties de um recurso. Quando o recurso é criado, ele será marcado com as tags que você declara.

JSON

```
"Tags" : [  
  {  
    "Key" : "keyname1",  
    "Value" : "value1"  
  },  
  {  
    "Key" : "keyname2",  
    "Value" : "value2"  
  }  
]
```

YAML

```
Tags:  
-  
  Key: "keyname1"  
  Value: "value1"  
-  
  Key: "keyname2"  
  Value: "value2"
```

Snippets de modelos do CloudFormation para ajuste de escala automático

Com o Amazon EC2 Auto Scaling, é possível escalar automaticamente instâncias do Amazon EC2 seja com políticas de escalabilidade ou com escalabilidade programada. Grupos do Auto Scaling

são conjuntos de instâncias do Amazon EC2 que habilitam recursos de escalabilidade automática e gerenciamento de frotas, como políticas de escalabilidade, ações programadas, verificações de integridade, ganchos do ciclo de vida e balanceamento de carga.

O Aplicativo Auto Scaling fornece escalabilidade automática de diferentes recursos além do Amazon EC2, com políticas de escalabilidade ou com escalabilidade programada.

É possível criar e configurar grupos de Auto Scaling, políticas de escalabilidade, ações programadas e outros recursos de ajuste de escala automático com parte de sua infraestrutura usando modelos do CloudFormation. Os modelos facilitam o gerenciamento e a automação da implantação de recursos de ajuste de escala automático de forma consistente e repetível.

Os snippets de modelos de exemplo a seguir descrevem os recursos ou componentes do CloudFormation para o Amazon EC2 Auto Scaling e o ajuste de escala automático de aplicações. Esses snippets foram projetados para serem integrados a um modelo e não devem ser executados de forma independente.

Categorias de snippets

- [Configurar os recursos do Amazon EC2 Auto Scaling com o CloudFormation](#)
- [Configurar recursos de ajuste de escala automático de aplicações com o CloudFormation](#)

Configurar os recursos do Amazon EC2 Auto Scaling com o CloudFormation

Os exemplos a seguir mostram snippets diferentes a serem incluídos em modelos para uso com o Amazon EC2 Auto Scaling.

Categorias de snippets

- [Criar um grupo do Auto Scaling com uma única instância](#)
- [Criar um grupo do Auto Scaling com balanceador de carga anexado](#)
- [Criar um grupo do Auto Scaling com notificações](#)
- [Criar um grupo do Auto Scaling que usa um CreationPolicy e um UpdatePolicy](#)
- [Criar uma política de escalabilidade em etapas](#)
- [Exemplos de grupos de instâncias mistas](#)
- [Exemplos de configuração de ativação](#)

Criar um grupo do Auto Scaling com uma única instância

Este exemplo mostra um recurso [AWS::AutoScaling::AutoScalingGroup](#) com uma única instância para ajudar você a começar. A propriedade `VPCZoneIdentifier` do grupo do Auto Scaling especifica uma lista de sub-redes existentes em três diferentes zonas de disponibilidade. É necessário especificar os IDs de sub-rede aplicáveis da conta antes de criar a pilha. A propriedade `LaunchTemplate` faz referência a um recurso [AWS::EC2::LaunchTemplate](#) com o nome lógico `myLaunchTemplate` definido em outra parte do modelo.

Note

Para obter exemplos de modelos inicialização, consulte [Criar modelos de execução com o CloudFormation](#) na seção de snippets do Amazon EC2 e a seção [Exemplos](#) no recurso `AWS::EC2::LaunchTemplate`.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
      },
      "Version" : {
        "Fn::GetAtt" : [
          "myLaunchTemplate",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '1'
    MinSize: '1'
```

Criar um grupo do Auto Scaling com balanceador de carga anexado

Este exemplo mostra um recurso [AWS::AutoScaling::AutoScalingGroup](#) para balanceamento de carga em diversos servidores. Ele especifica os nomes lógicos dos recursos da AWS declarados em outro lugar no mesmo modelo.

1. A propriedade `VPCZoneIdentifier` especifica os nomes lógicos de dois recursos [AWS::EC2::Subnet](#) em que as instâncias do EC2 do grupo do Auto Scaling serão criadas: `myPublicSubnet1` e `myPublicSubnet2`.
2. A propriedade `LaunchTemplate` especifica um recurso [AWS::EC2::LaunchTemplate](#) com o nome lógico `myLaunchTemplate`.
3. A propriedade `TargetGroupARNs` lista os grupos de destino de um Application Load Balancer ou um Network Load Balancer usado para rotear o tráfego para o grupo de Auto Scaling. Neste exemplo, um grupo de destino é especificado, um recurso [AWS::ElasticLoadBalancingV2::TargetGroup](#) com o nome lógico `myTargetGroup`.

JSON

```
"myServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ { "Ref" : "myPublicSubnet1" }, { "Ref" :
"myPublicSubnet2" } ],
    "LaunchTemplate" : {
```

```

    "LaunchTemplateId" : {
      "Ref" : "myLaunchTemplate"
    },
    "Version" : {
      "Fn::GetAtt" : [
        "myLaunchTemplate",
        "LatestVersionNumber"
      ]
    }
  },
  "MaxSize" : "5",
  "MinSize" : "1",
  "TargetGroupARNs" : [ { "Ref" : "myTargetGroup" } ]
}
}

```

YAML

```

myServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref myPublicSubnet1
      - !Ref myPublicSubnet2
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    TargetGroupARNs:
      - !Ref myTargetGroup

```

Consulte também

Para obter um exemplo detalhado que cria um grupo de Auto Scaling com uma política de escalabilidade de rastreamento de destino baseada na métrica `ALBRequestCountPerTarget` predefinida para o Application Load Balancer, consulte a seção [Exemplos](#) no recurso `AWS::AutoScaling::ScalingPolicy`.

Criar um grupo do Auto Scaling com notificações

Este exemplo mostra um recurso [AWS::AutoScaling::AutoScalingGroup](#) que envia notificações do Amazon SNS quando eventos especificados ocorrem. A propriedade

NotificationConfigurations especifica o tópico do SNS em que o CloudFormation envia uma notificação e os eventos farão com que o CloudFormation envie notificações. Quando os eventos especificados por NotificationTypes ocorrem, o CloudFormation enviará uma notificação para o tópico do SNS especificado pelo TopicARN. Quando você inicializa a pilha, o CloudFormation cria um recurso [AWS::SNS::Subscription](#) (snsTopicForAutoScalingGroup) que é declarado no mesmo modelo.

A propriedade VPCZoneIdentifier do grupo do Auto Scaling especifica uma lista de sub-redes existentes em três diferentes zonas de disponibilidade. É necessário especificar os IDs de sub-rede aplicáveis da conta antes de criar a pilha. A propriedade LaunchTemplate faz referência ao nome lógico de um recurso [AWS::EC2::LaunchTemplate](#) declarado em outro lugar no mesmo modelo.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": [
    "snsTopicForAutoScalingGroup"
  ],
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "logicalName"
      },
    },
    "Version" : {
      "Fn::GetAtt" : [
        "logicalName",
        "LatestVersionNumber"
      ]
    },
    "MaxSize" : "5",
    "MinSize" : "1",
    "NotificationConfigurations" : [
      {
        "TopicARN" : { "Ref" : "snsTopicForAutoScalingGroup" },
        "NotificationTypes" : [
          "autoscaling:EC2_INSTANCE_LAUNCH",
          "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
          "autoscaling:EC2_INSTANCE_TERMINATE",
          "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
```

```

        "autoscaling:TEST_NOTIFICATION"
    ]
}
]
}
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - snsTopicForAutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref logicalName
      Version: !GetAtt logicalName.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref snsTopicForAutoScalingGroup
        NotificationTypes:
          - autoscaling:EC2_INSTANCE_LAUNCH
          - autoscaling:EC2_INSTANCE_LAUNCH_ERROR
          - autoscaling:EC2_INSTANCE_TERMINATE
          - autoscaling:EC2_INSTANCE_TERMINATE_ERROR
          - autoscaling:TEST_NOTIFICATION

```

Criar um grupo do Auto Scaling que usa um **CreationPolicy** e um **UpdatePolicy**

O exemplo a seguir mostra como adicionar os atributos **CreationPolicy** e **UpdatePolicy** a um recurso [AWS::AutoScaling::AutoScalingGroup](#).

A política de criação de exemplos impede que o grupo do Auto Scaling atinja o status **CREATE_COMPLETE** até que CloudFormation receba o número **Count** de sinais de êxito quando o grupo estiver pronto. Para sinalizar que o grupo do Auto Scaling está pronto, um script auxiliar **cfn-signal** adicionado aos dados do usuário do modelo de inicialização (não mostrado) é executado nas instâncias. Se as instâncias não enviarem um sinal dentro do **Timeout** especificado,

o CloudFormation assumirá que as instâncias não foram criadas, a criação do recurso falhará e o CloudFormation reverterá a pilha.

A política de atualização de exemplo instrui o CloudFormation a executar uma atualização contínua usando a propriedade `AutoScalingRollingUpdate`. A atualização contínua faz alterações no grupo de Auto Scaling em pequenos lotes (neste exemplo, instância por instância) com base no `MaxBatchSize` e em um tempo de pausa entre lotes de atualizações com base no `PauseTime`. O atributo `MinInstancesInService` especifica o número mínimo de instâncias que devem estar em serviço no grupo do Auto Scaling e o CloudFormation atualiza instâncias antigas.

O atributo `WaitOnResourceSignals` é definido como `true`. O CloudFormation deve receber um sinal de cada instância nova dentro do `PauseTime` especificado para continuar a atualização. Embora a atualização da pilha esteja em andamento, os seguintes processos de Auto Scaling do EC2 são suspensos: `HealthCheck`, `ReplaceUnhealthy`, `AZRebalance`, `AlarmNotification` e `ScheduledActions`. Observação: não suspenda os tipos de processo `Launch`, `Terminate` ou `AddToLoadBalancer` (se o grupo de Auto Scaling estiver sendo usado com o Elastic Load Balancing), pois isso pode impedir que a atualização contínua funcione corretamente.

A propriedade `VPCZoneIdentifier` do grupo do Auto Scaling especifica uma lista de sub-redes existentes em três diferentes zonas de disponibilidade. É necessário especificar os IDs de sub-rede aplicáveis da conta antes de criar a pilha. A propriedade `LaunchTemplate` faz referência ao nome lógico de um recurso [AWS::EC2::LaunchTemplate](#) declarado em outro lugar no mesmo modelo.

Para obter mais informações sobre os atributos `CreationPolicy` e `UpdatePolicy`, consulte a [Referência de atributos de recursos](#).

JSON

```
{
  "Resources": {
    "myASG": {
      "CreationPolicy": {
        "ResourceSignal": {
          "Count": "3",
          "Timeout": "PT15M"
        }
      },
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "3",
          "MaxBatchSize": "1",
```



```

        "PauseTime": "PT12M5S",
        "WaitOnResourceSignals": "true",
        "SuspendProcesses": [
            "HealthCheck",
            "ReplaceUnhealthy",
            "AZRebalance",
            "AlarmNotification",
            "ScheduledActions",
            "InstanceRefresh"
        ]
    },
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Properties": {
        "VPCZoneIdentifier": [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
        "LaunchTemplate": {
            "LaunchTemplateId": {
                "Ref": "logicalName"
            },
            "Version": {
                "Fn::GetAtt": [
                    "logicalName",
                    "LatestVersionNumber"
                ]
            }
        },
        "MaxSize": "5",
        "MinSize": "3"
    }
}
}
}
}

```

YAML

```

---
Resources:
  myASG:
    CreationPolicy:
      ResourceSignal:
        Count: '3'
        Timeout: PT15M
    UpdatePolicy:

```

```
AutoScalingRollingUpdate:
  MinInstancesInService: '3'
  MaxBatchSize: '1'
  PauseTime: PT12M5S
  WaitOnResourceSignals: true
  SuspendProcesses:
    - HealthCheck
    - ReplaceUnhealthy
    - AZRebalance
    - AlarmNotification
    - ScheduledActions
    - InstanceRefresh
Type: AWS::AutoScaling::AutoScalingGroup
Properties:
  VPCZoneIdentifier:
    - subnetIdAz1
    - subnetIdAz2
    - subnetIdAz3
  LaunchTemplate:
    LaunchTemplateId: !Ref logicalName
    Version: !GetAtt logicalName.LatestVersionNumber
  MaxSize: '5'
  MinSize: '3'
```

Criar uma política de escalabilidade em etapas

Este exemplo mostra um recurso [AWS::AutoScaling::ScalingPolicy](#) que aumenta a escala horizontalmente do grupo do Auto Scaling usando uma política de escalabilidade em etapas. A propriedade `AdjustmentType` especifica `ChangeInCapacity`, o que significa que `ScalingAdjustment` representa o número de instâncias para adicionar (se `ScalingAdjustment` for positivo) ou excluir (se for negativo). Neste exemplo, `ScalingAdjustment` é 1; por isso, a política incrementa o número de instâncias do EC2 no grupo em 1 quando o limite do alarme é atingido.

O recurso `CPUAlarmHigh` de [AWS::CloudWatch::Alarm](#) especifica a política de escalabilidade `ASGScalingPolicyHigh` como a ação a ser executada quando o alarme está em um estado `ALARM` (`AlarmActions`). A propriedade `Dimensions` faz referência ao nome lógico de um recurso [AWS::AutoScaling::AutoScalingGroup](#) declarado em outro lugar no mesmo modelo.

JSON

```

{
  "Resources":{
    "ASGScalingPolicyHigh":{
      "Type":"AWS::AutoScaling::ScalingPolicy",
      "Properties":{
        "AutoScalingGroupName":{ "Ref":"logicalName" },
        "PolicyType":"StepScaling",
        "AdjustmentType":"ChangeInCapacity",
        "StepAdjustments":[
          {
            "MetricIntervalLowerBound":0,
            "ScalingAdjustment":1
          }
        ]
      }
    },
    "CPUAlarmHigh":{
      "Type":"AWS::CloudWatch::Alarm",
      "Properties":{
        "EvaluationPeriods":"2",
        "Statistic":"Average",
        "Threshold":"90",
        "AlarmDescription":"Scale out if CPU > 90% for 2 minutes",
        "Period":"60",
        "AlarmActions":[ { "Ref":"ASGScalingPolicyHigh" } ],
        "Namespace":"AWS/EC2",
        "Dimensions":[
          {
            "Name":"AutoScalingGroupName",
            "Value":{ "Ref":"logicalName" }
          }
        ],
        "ComparisonOperator":"GreaterThanThreshold",
        "MetricName":"CPUUtilization"
      }
    }
  }
}

```

YAML

```

---
Resources:

```

```
ASGScalingPolicyHigh:
  Type: AWS::AutoScaling::ScalingPolicy
  Properties:
    AutoScalingGroupName: !Ref logicalName
    PolicyType: StepScaling
    AdjustmentType: ChangeInCapacity
    StepAdjustments:
      - MetricIntervalLowerBound: 0
        ScalingAdjustment: 1
CPUAlarmHigh:
  Type: AWS::CloudWatch::Alarm
  Properties:
    EvaluationPeriods: 2
    Statistic: Average
    Threshold: 90
    AlarmDescription: 'Scale out if CPU > 90% for 2 minutes'
    Period: 60
    AlarmActions:
      - !Ref ASGScalingPolicyHigh
    Namespace: AWS/EC2
    Dimensions:
      - Name: AutoScalingGroupName
        Value:
          !Ref logicalName
    ComparisonOperator: GreaterThanThreshold
    MetricName: CPUUtilization
```

Consulte também

Para obter mais exemplos de modelos para políticas de escalabilidade, consulte a seção [Exemplos](#) no recurso `AWS::AutoScaling::ScalingPolicy`.

Exemplos de grupos de instâncias mistas

Criar um grupo do Auto Scaling usando seleção de tipo de instância baseada em atributos

Este exemplo mostra um recurso [AWS::AutoScaling::AutoScalingGroup](#) que contém informações para executar um grupo de instâncias mistas utilizando a seleção de tipos de instâncias baseadas em atributos. Você especifica os valores mínimo e máximo para a propriedade `VCpuCount` e o valor mínimo para a propriedade `MemoryMiB`. Todos os tipos de instância usados pelo grupo do Auto Scaling devem corresponder aos atributos de instância exigidos.

A propriedade `VPCZoneIdentifier` do grupo do Auto Scaling especifica uma lista de sub-redes existentes em três diferentes zonas de disponibilidade. É necessário especificar os IDs de sub-rede aplicáveis da conta antes de criar a pilha. A propriedade `LaunchTemplate` faz referência ao nome lógico de um recurso [AWS::EC2::LaunchTemplate](#) declarado em outro lugar no mesmo modelo.

JSON

```
{
  "Resources":{
    "myASG":{
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "VPCZoneIdentifier":[
          "subnetIdAz1",
          "subnetIdAz2",
          "subnetIdAz3"
        ],
        "MixedInstancesPolicy":{
          "LaunchTemplate":{
            "LaunchTemplateSpecification":{
              "LaunchTemplateId":{
                "Ref":"logicalName"
              },
              "Version":{
                "Fn::GetAtt":[
                  "logicalName",
                  "LatestVersionNumber"
                ]
              }
            },
            "Overrides":[
              {
                "InstanceRequirements":{
                  "VCpuCount":{
                    "Min":2,
                    "Max":4
                  },
                  "MemoryMiB":{
                    "Min":2048
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

```

    }
  },
  "MaxSize": "5",
  "MinSize": "1"
}
}
}
}
}

```

YAML

```

---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
        - subnetIdAz2
        - subnetIdAz3
      MixedInstancesPolicy:
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref logicalName
            Version: !GetAtt logicalName.LatestVersionNumber
          Overrides:
            - InstanceRequirements:
                VCpuCount:
                  Min: 2
                  Max: 4
                MemoryMiB:
                  Min: 2048
      MaxSize: '5'
      MinSize: '1'

```

Exemplos de configuração de ativação

Criar uma configuração de execução

Este exemplo mostra um recurso [AWS::AutoScaling::LaunchConfiguration](#) para um grupo do AutoScaling em que você especifica valores para as propriedades ImageId, InstanceType e SecurityGroups. A propriedade SecurityGroups especifica o nome lógico de um recurso

[AWS::EC2::SecurityGroup](#) especificado em outro local no modelo e um grupo de segurança existente do EC2 chamado `myExistingEC2SecurityGroup`.

JSON

```
"mySimpleConfig" : {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
  "Properties" : {
    "ImageId" : "ami-02354e95b3example",
    "InstanceType" : "t3.micro",
    "SecurityGroups" : [ { "Ref" : "logicalName" }, "myExistingEC2SecurityGroup" ]
  }
}
```

YAML

```
mySimpleConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: ami-02354e95b3example
    InstanceType: t3.micro
    SecurityGroups:
      - !Ref logicalName
      - myExistingEC2SecurityGroup
```

Criar um grupo do Auto Scaling que usa uma configuração de execução

Este exemplo mostra um recurso [AWS::AutoScaling::AutoScalingGroup](#) com uma única instância. A propriedade `VPCZoneIdentifier` do grupo do Auto Scaling especifica uma lista de sub-redes existentes em três diferentes zonas de disponibilidade. É necessário especificar os IDs de sub-rede aplicáveis da conta antes de criar a pilha. A propriedade `LaunchConfigurationName` faz referência a um recurso [AWS::AutoScaling::LaunchConfiguration](#) com o nome lógico `mySimpleConfig` definido em seu modelo.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
```

```
    "LaunchConfigurationName" : { "Ref" : "mySimpleConfig" },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchConfigurationName: !Ref mySimpleConfig
    MaxSize: '1'
    MinSize: '1'
```

Configurar recursos de ajuste de escala automático de aplicações com o CloudFormation

Esta seção fornece exemplos de modelos do CloudFormation para políticas de escalabilidade e ações programadas do aplicação Auto Scaling para diferentes recursos da AWS.

Important

Quando um snippet do ajuste de escala automático da aplicação é incluído no modelo, talvez seja necessário declarar uma dependência do recurso escalável específico criado por meio do modelo usando o atributo `DependsOn`. Isso substitui o paralelismo padrão e direciona o CloudFormation para operar em recursos em uma ordem especificada. Caso contrário, a configuração de escalabilidade pode ser aplicada antes que o recurso tenha sido configurado completamente.

Para obter mais informações, consulte o atributo [DependsOn](#).

Categorias de snippets

- [Criar uma política de escalabilidade para uma frota do AppStream](#)
- [Criar uma política de escalabilidade para um cluster de banco de dados Aurora](#)

- [Criar uma política de escalabilidade para uma tabela do DynamoDB](#)
- [Criar uma política de escalabilidade para um serviço do Amazon ECS \(métricas: média de CPU e memória\)](#)
- [Criar uma política de escalabilidade para um serviço do Amazon ECS \(métrica: contagem média de solicitações por destino\)](#)
- [Criar uma ação programada com uma expressão cron para uma função do Lambda](#)
- [Criar uma ação programada com uma expressão at para uma frota spot](#)

Criar uma política de escalabilidade para uma frota do AppStream

Este snippet mostra como criar uma política e aplicá-la a um recurso [AWS::AppStream::Fleet](#) usando o recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#). O recurso [AWS::ApplicationAutoScaling::ScalableTarget](#) declara um destino escalável ao qual essa política é aplicada. O Application Auto Scaling pode dimensionar o número de instâncias da frota no mínimo de 1 instância e no máximo 20. A política mantém a utilização média da capacidade da frota em 75%, com períodos de esfriamento de aumento e redução de escala horizontalmente de 300 segundos (5 minutos).

Ela usa as funções intrínsecas Fn::Join e Ref para construir a propriedade ResourceId com o nome lógico do recurso AWS::AppStream::Fleet especificado no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 20,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet" },
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
```

```
[
    "fleet",
    {
        "Ref" : "logicalName"
    }
]
]
}
},
"ScalingPolicyAppStreamFleet" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu75" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
            "Fn::Join" : [
                "/",
                [
                    "fleet",
                    {
                        "Ref" : "logicalName"
                    }
                ]
            ]
        },
    },
    "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : 75,
        "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "AppStreamAverageCapacityUtilization"
        },
        "ScaleInCooldown" : 300,
        "ScaleOutCooldown" : 300
    }
}
}
```

YAML

```

---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 20
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet'
      ServiceNamespace: appstream
      ScalableDimension: appstream:fleet:DesiredCapacity
      ResourceId: !Join
        - /
        - - fleet
          - !Ref logicalName
  ScalingPolicyAppStreamFleet:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu75
      PolicyType: TargetTrackingScaling
      ServiceNamespace: appstream
      ScalableDimension: appstream:fleet:DesiredCapacity
      ResourceId: !Join
        - /
        - - fleet
          - !Ref logicalName
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 75
        PredefinedMetricSpecification:
          PredefinedMetricType: AppStreamAverageCapacityUtilization
        ScaleInCooldown: 300
        ScaleOutCooldown: 300

```

Criar uma política de escalabilidade para um cluster de banco de dados Aurora

Neste snippet, você registra um recurso [AWS::RDS::DBCluster](#). O recurso [AWS::ApplicationAutoScaling::ScalableTarget](#) indica que o cluster de banco de dados deve ser escalado dinamicamente para ter de uma a oito réplicas do Aurora. Também é possível

aplicar uma política de escalabilidade de rastreamento de destino ao cluster usando o recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#).

Nessa configuração, a métrica predefinida `RDSReaderAverageCPUUtilization` é usada para ajustar um cluster de banco de dados Aurora com base em uma utilização média da CPU de 40 por cento em todas as réplicas do Aurora nesse cluster de banco de dados Aurora. A configuração fornece um desaquecimento de redução de 10 minutos e em um desaquecimento de expansão de 5 minutos.

Esse exemplo usa a função intrínseca `Fn::Sub` para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::RDS::DBCluster` especificado no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 8,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam:${AWS::AccountId}:role/
aws-service-role/rds.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_RDSCluster" },
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" }
      }
    },
    "ScalingPolicyDBCluster" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu40" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 40,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "RDSReaderAverageCPUUtilization"
          }
        }
      }
    }
  }
}
```

```

        "ScaleInCooldown" : 600,
        "ScaleOutCooldown" : 300
    }
}
}
}
}

```

YAML

```

---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 8
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster'
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount
      ResourceId: !Sub cluster:${logicalName}
  ScalingPolicyDBCluster:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu40
      PolicyType: TargetTrackingScaling
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount
      ResourceId: !Sub cluster:${logicalName}
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 40
        PredefinedMetricSpecification:
          PredefinedMetricType: RDSReaderAverageCPUUtilization
        ScaleInCooldown: 600
        ScaleOutCooldown: 300

```

Criar uma política de escalabilidade para uma tabela do DynamoDB

Esse trecho mostra como criar uma política com o tipo de política

`TargetTrackingScaling` e aplicá-la a um recurso [AWS::DynamoDB::Table](#)

usando o recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#). O recurso

[`AWS::ApplicationAutoScaling::ScalableTarget`](#) declara um destino escalável ao qual essa política é aplicada, com um mínimo de cinco unidades de capacidade de gravação e um máximo de 15. A política de escalabilidade escala o throughput da capacidade de gravação da tabela para manter a utilização de destino em 50% com base na métrica `DynamoDBWriteCapacityUtilization` predefinida.

Ela usa as funções intrínsecas `Fn::Join` e `Ref` para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::DynamoDB::Table` especificado no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

Note

Para obter mais informações sobre como criar um modelo do CloudFormation para recursos do DynamoDB, consulte a postagem [Como usar o CloudFormation para configurar o ajuste de escala automático para tabelas e índices do Amazon DynamoDB](#) no blog sobre banco de dados da AWS.

JSON

```
{
  "Resources" : {
    "WriteCapacityScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 15,
        "MinCapacity" : 5,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
        "ServiceNamespace" : "dynamodb",
        "ScalableDimension" : "dynamodb:table:WriteCapacityUnits",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "table",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    }
  }
}
```

```
]
}
}
},
"WriteScalingPolicy" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
        "PolicyName" : "WriteScalingPolicy",
        "PolicyType" : "TargetTrackingScaling",
        "ScalingTargetId" : { "Ref" : "WriteCapacityScalableTarget" },
        "TargetTrackingScalingPolicyConfiguration" : {
            "TargetValue" : 50.0,
            "ScaleInCooldown" : 60,
            "ScaleOutCooldown" : 60,
            "PredefinedMetricSpecification" : {
                "PredefinedMetricType" : "DynamoDBWriteCapacityUtilization"
            }
        }
    }
}
}
```

YAML

```

---
Resources:
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ServiceNamespace: dynamodb
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ResourceId: !Join
        - /
        - - table
          - !Ref LogicalName
    WriteScalingPolicy:

```

```

Type: AWS::ApplicationAutoScaling::ScalingPolicy
Properties:
  PolicyName: WriteScalingPolicy
  PolicyType: TargetTrackingScaling
  ScalingTargetId: !Ref WriteCapacityScalableTarget
  TargetTrackingScalingPolicyConfiguration:
    TargetValue: 50.0
    ScaleInCooldown: 60
    ScaleOutCooldown: 60
    PredefinedMetricSpecification:
      PredefinedMetricType: DynamoDBWriteCapacityUtilization

```

Criar uma política de escalabilidade para um serviço do Amazon ECS (métricas: média de CPU e memória)

Este snippet mostra como criar uma política e aplicá-la a um recurso [AWS::ECS::Service](#) usando o recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#). O recurso [AWS::ApplicationAutoScaling::ScalableTarget](#) declara um destino escalável ao qual essa política é aplicada. O Application Auto Scaling pode escalar o número de tarefas em um mínimo de 1 tarefa e um máximo de 6.

Ele cria duas políticas de escalabilidade com o tipo de política `TargetTrackingScaling`. As políticas são usadas para escalar o serviço do ECS com base no uso médio de CPU e memória do serviço. Ele usa as funções intrínsecas `Fn::Join` e `Ref` para criar a propriedade `ResourceId` com os nomes lógicos dos recursos [AWS::ECS::Cluster](#) (`myContainerCluster`) e [AWS::ECS::Service](#) (`myService`) especificados no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

JSON

```

{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : "6",
        "MinCapacity" : "1",
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam:${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",
        "ScalableDimension" : "ecs:service:DesiredCount",

```



```

    "ResourceId" : {
      "Fn::Join" : [
        "/",
        [
          "service",
          {
            "Ref" : "myContainerCluster"
          },
          {
            "Fn::GetAtt" : [
              "myService",
              "Name"
            ]
          }
        ]
      ]
    }
  },
  "ServiceScalingPolicyCPU" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
      "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu70" },
      "PolicyType" : "TargetTrackingScaling",
      "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
      "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : 70.0,
        "ScaleInCooldown" : 180,
        "ScaleOutCooldown" : 60,
        "PredefinedMetricSpecification" : {
          "PredefinedMetricType" : "ECSServiceAverageCPUUtilization"
        }
      }
    }
  },
  "ServiceScalingPolicyMem" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
      "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-mem90" },
      "PolicyType" : "TargetTrackingScaling",
      "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
      "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : 90.0,
        "ScaleInCooldown" : 180,

```

```

        "ScaleOutCooldown" : 60,
        "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "ECSServiceAverageMemoryUtilization"
        }
    }
}
}
}
}
}
}
}

```

YAML

```

---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 6
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyCPU:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu70
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 70.0
        ScaleInCooldown: 180
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: ECSServiceAverageCPUUtilization
  ServiceScalingPolicyMem:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy

```

Properties:

```

PolicyName: !Sub ${AWS::StackName}-target-tracking-mem90
PolicyType: TargetTrackingScaling
ScalingTargetId: !Ref ECSScalableTarget
TargetTrackingScalingPolicyConfiguration:
  TargetValue: 90.0
  ScaleInCooldown: 180
  ScaleOutCooldown: 60
  PredefinedMetricSpecification:
    PredefinedMetricType: ECSServiceAverageMemoryUtilization

```

Criar uma política de escalabilidade para um serviço do Amazon ECS (métrica: contagem média de solicitações por destino)

O exemplo a seguir aplica uma política de escalabilidade de rastreamento de destino com a métrica predefinida `ALBRequestCountPerTarget` a um serviço do ECS. A política é usada para adicionar capacidade ao serviço do ECS quando a contagem de solicitações por destino (por minuto) excede o valor de destino. Como o valor de `DisableScaleIn` é definido como `true`, a política de rastreamento de destino não removerá a capacidade do destino escalável.

Ele usa as funções intrínsecas `Fn::Join` e `Fn::GetAtt` para criar a propriedade `ResourceLabel` com os nomes lógicos dos recursos [AWS::ElasticLoadBalancingV2::LoadBalancer](#) (`myLoadBalancer`) e [AWS::ElasticLoadBalancingV2::TargetGroup](#) (`myTargetGroup`) especificados no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

As propriedades `MaxCapacity` e `MinCapacity` do destino escalável e a propriedade `TargetValue` dos valores de parâmetro de referência da política de escalabilidade que você transfere para o modelo ao criar ou atualizar uma pilha.

JSON

```

{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : { "Ref" : "MaxCount" },
        "MinCapacity" : { "Ref" : "MinCount" },
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService" },

```

```

    "ServiceNamespace" : "ecs",
    "ScalableDimension" : "ecs:service:DesiredCount",
    "ResourceId" : {
      "Fn::Join" : [
        "/",
        [
          "service",
          {
            "Ref" : "myContainerCluster"
          },
          {
            "Fn::GetAtt" : [
              "myService",
              "Name"
            ]
          }
        ]
      ]
    }
  },
  "ServiceScalingPolicyALB" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
      "PolicyName" : "alb-requests-per-target-per-minute",
      "PolicyType" : "TargetTrackingScaling",
      "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
      "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : { "Ref" : "ALBPolicyTargetValue" },
        "ScaleInCooldown" : 180,
        "ScaleOutCooldown" : 30,
        "DisableScaleIn" : true,
        "PredefinedMetricSpecification" : {
          "PredefinedMetricType" : "ALBRequestCountPerTarget",
          "ResourceLabel" : {
            "Fn::Join" : [
              "/",
              [
                {
                  "Fn::GetAtt" : [
                    "myLoadBalancer",
                    "LoadBalancerFullName"
                  ]
                }
              ]
            ]
          }
        }
      }
    }
  },

```

```
{
  "Fn::GetAtt" : [
    "myTargetGroup",
    "TargetGroupFullName"
  ]
}
```

YAML

```

---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: !Ref MaxCount
      MinCapacity: !Ref MinCount
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyALB:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: alb-requests-per-target-per-minute
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: !Ref ALBPolicyTargetValue

```

```

ScaleInCooldown: 180
ScaleOutCooldown: 30
DisableScaleIn: true
PredefinedMetricSpecification:
  PredefinedMetricType: ALBRequestCountPerTarget
  ResourceLabel: !Join
    - '/'
    - !GetAtt myLoadBalancer.LoadBalancerFullName
    - !GetAtt myTargetGroup.TargetGroupFullName

```

Criar uma ação programada com uma expressão cron para uma função do Lambda

Esse trecho registra a simultaneidade provisionada para um alias de função ([AWS::Lambda::Alias](#)) denominado BLUE usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ele também cria uma ação programada com uma programação recorrente usando uma expressão cron. O fuso horário para a programação recorrente é UTC.

Ele usa as funções intrínsecas Fn::Join e Ref na propriedade RoleARN para especificar o ARN do perfil vinculado ao serviço. Ele usa a função intrínseca Fn::Sub para criar a propriedade ResourceId com o nome lógico do recurso [AWS::Lambda::Function](#) ou [AWS::Serverless::Function](#) especificado no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

Note

Não é possível alocar simultaneidade provisionada em um alias que aponte para a versão não publicada (\$LATEST).

Para obter mais informações sobre como criar um modelo do CloudFormation para recursos do Lambda, consulte a publicação do blog [Scheduling AWS Lambda Provisioned Concurrency for recurring peak usage](#), no blog sobre computação da AWS.

JSON

```

{
  "ScalableTarget" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties" : {
      "MaxCapacity" : 250,
      "MinCapacity" : 0,

```

```

    "RoleARN" : {
      "Fn::Join" : [
        ":",
        [
          "arn:aws:iam:",
          {
            "Ref" : "AWS::AccountId"
          },
          "role/aws-service-role/lambda.application-autoscaling.amazonaws.com/"
        ]
      ],
      "ServiceNamespace" : "lambda",
      "ScalableDimension" : "lambda:function:ProvisionedConcurrency",
      "ResourceId" : { "Fn::Sub" : "function:${logicalName}:BLUE" },
      "ScheduledActions" : [
        {
          "ScalableTargetAction" : {
            "MinCapacity" : "250"
          },
          "ScheduledActionName" : "my-scale-out-scheduled-action",
          "Schedule" : "cron(0 18 * * ? *)",
          "EndTime" : "2022-12-31T12:00:00.000Z"
        }
      ]
    }
  }
}

```

YAML

```

ScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: 250
    MinCapacity: 0
    RoleARN: !Join
      - ':'
      - - 'arn:aws:iam:'
        - !Ref 'AWS::AccountId'
        - role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
        - AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency

```

```

ServiceNamespace: lambda
ScalableDimension: lambda:function:ProvisionedConcurrency
ResourceId: !Sub function:${logicalName}:BLUE
ScheduledActions:
  - ScalableTargetAction:
      MinCapacity: 250
      ScheduledActionName: my-scale-out-scheduled-action
      Schedule: 'cron(0 18 * * ? *)'
      EndTime: '2022-12-31T12:00:00.000Z'

```

Criar uma ação programada com uma expressão **at** para uma frota spot

Este snippet mostra como criar duas ações programadas que ocorrem somente uma vez para um recurso [AWS::EC2::SpotFleet](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). O fuso horário para cada ação programada para ocorrer uma única vez é UTC.

Ela usa as funções intrínsecas Fn::Join e Ref para construir a propriedade ResourceId com o nome lógico do recurso AWS::EC2::SpotFleet especificado no mesmo modelo. Para obter mais informações, consulte [Referência de funções intrínsecas](#).

Note

A solicitação de frota spot deve ter o tipo de solicitação `maintain`. O dimensionamento automático não é compatível com solicitações únicas nem blocos spot.

JSON

```

{
  "Resources" : {
    "SpotFleetScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 0,
        "MinCapacity" : 0,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam:${AWS::AccountId}:role/aws-service-role/ec2.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest" },
        "ServiceNamespace" : "ec2",
        "ScalableDimension" : "ec2:spot-fleet-request:TargetCapacity",

```



```

    "ResourceId" : {
      "Fn::Join" : [
        "/",
        [
          "spot-fleet-request",
          {
            "Ref" : "logicalName"
          }
        ]
      ]
    },
    "ScheduledActions" : [
      {
        "ScalableTargetAction" : {
          "MaxCapacity" : "10",
          "MinCapacity" : "10"
        },
        "ScheduledActionName" : "my-scale-out-scheduled-action",
        "Schedule" : "at(2022-05-20T13:00:00)"
      },
      {
        "ScalableTargetAction" : {
          "MaxCapacity" : "0",
          "MinCapacity" : "0"
        },
        "ScheduledActionName" : "my-scale-in-scheduled-action",
        "Schedule" : "at(2022-05-20T21:00:00)"
      }
    ]
  }
}

```

YAML

```

---
Resources:
  SpotFleetScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 0
      MinCapacity: 0

```

```

RoleARN:
  Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ec2.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest'
ServiceNamespace: ec2
ScalableDimension: 'ec2:spot-fleet-request:TargetCapacity'
ResourceId: !Join
  - /
  - - spot-fleet-request
    - !Ref LogicalName
ScheduledActions:
  - ScalableTargetAction:
      MaxCapacity: 10
      MinCapacity: 10
      ScheduledActionName: my-scale-out-scheduled-action
      Schedule: 'at(2022-05-20T13:00:00)'
  - ScalableTargetAction:
      MaxCapacity: 0
      MinCapacity: 0
      ScheduledActionName: my-scale-in-scheduled-action
      Schedule: 'at(2022-05-20T21:00:00)'

```

Snippets de modelo do Console de faturamento da AWS

Esse exemplo cria um plano de preços com uma regra de preço com marcação global de 10%. Esse plano de preços está anexado ao grupo de faturamento. O grupo de faturamento também tem dois itens de linha personalizados que aplicam uma cobrança de USD 10 e uma cobrança de 10% sobre o custo total do grupo de faturamento.

JSON

```

{
  "Parameters": {
    "LinkedAccountIds": {
      "Type": "ListNumber"
    },
    "PrimaryAccountId": {
      "Type": "Number"
    }
  },
  "Resources": {
    "TestPricingRule": {
      "Type": "AWS::BillingConductor::PricingRule",
      "Properties": {

```

```

        "Name": "TestPricingRule",
        "Description": "Test pricing rule created through Cloudformation. Mark
everything by 10%.",
        "Type": "MARKUP",
        "Scope": "GLOBAL",
        "ModifierPercentage": 10
    }
},
"TestPricingPlan": {
    "Type": "AWS::BillingConductor::PricingPlan",
    "Properties": {
        "Name": "TestPricingPlan",
        "Description": "Test pricing plan created through Cloudformation.",
        "PricingRuleArns": [
            {"Fn::GetAtt": ["TestPricingRule", "Arn"]}
        ]
    }
},
"TestBillingGroup": {
    "Type": "AWS::BillingConductor::BillingGroup",
    "Properties": {
        "Name": "TestBillingGroup",
        "Description": "Test billing group created through Cloudformation with 1
linked account. The linked account is also the primary account.",
        "PrimaryAccountId": {
            "Ref": "PrimaryAccountId"
        },
        "AccountGrouping": {
            "LinkedAccountIds": null
        },
        "ComputationPreference": {
            "PricingPlanArn": {
                "Fn::GetAtt": ["TestPricingPlan", "Arn"]
            }
        }
    }
},
"TestFlatCustomLineItem": {
    "Type": "AWS::BillingConductor::CustomLineItem",
    "Properties": {
        "Name": "TestFlatCustomLineItem",
        "Description": "Test flat custom line item created through Cloudformation
for a $10 charge.",
        "BillingGroupArn": {

```

```

    "Fn::GetAtt": ["TestBillingGroup", "Arn"]
  },
  "CustomLineItemChargeDetails": {
    "Flat": {
      "ChargeValue": 10
    },
    "Type": "FEE"
  }
},
"TestPercentageCustomLineItem": {
  "Type": "AWS::BillingConductor::CustomLineItem",
  "Properties": {
    "Name": "TestPercentageCustomLineItem",
    "Description": "Test percentage custom line item created through Cloudformation for a %10 additional charge on the overall total bill of the billing group.",
    "BillingGroupArn": {
      "Fn::GetAtt": ["TestBillingGroup", "Arn"]
    },
    "CustomLineItemChargeDetails": {
      "Percentage": {
        "PercentageValue": 10,
        "ChildAssociatedResources": [
          {"Fn::GetAtt": ["TestBillingGroup", "Arn"]}
        ]
      },
      "Type": "FEE"
    }
  }
}
}
```

YAML

```
Parameters:
  LinkedAccountIds:
    Type: ListNumber
  PrimaryAccountId:
    Type: Number
Resources:
  TestPricingRule:
```

```

Type: AWS::BillingConductor::PricingRule
Properties:
  Name: 'TestPricingRule'
  Description: 'Test pricing rule created through Cloudformation. Mark everything
by 10%.'
  Type: 'MARKUP'
  Scope: 'GLOBAL'
  ModifierPercentage: 10
TestPricingPlan:
  Type: AWS::BillingConductor::PricingPlan
  Properties:
    Name: 'TestPricingPlan'
    Description: 'Test pricing plan created through Cloudformation.'
    PricingRuleArns:
      - !GetAtt TestPricingRule.Arn
TestBillingGroup:
  Type: AWS::BillingConductor::BillingGroup
  Properties:
    Name: 'TestBillingGroup'
    Description: 'Test billing group created through Cloudformation with 1 linked
account. The linked account is also the primary account.'
    PrimaryAccountId: !Ref PrimaryAccountId
    AccountGrouping:
      LinkedAccountIds: !Ref LinkedAccountIds
    ComputationPreference:
      PricingPlanArn: !GetAtt TestPricingPlan.Arn
TestFlatCustomLineItem:
  Type: AWS::BillingConductor::CustomLineItem
  Properties:
    Name: 'TestFlatCustomLineItem'
    Description: 'Test flat custom line item created through Cloudformation for a $10
charge.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn
    CustomLineItemChargeDetails:
      Flat:
        ChargeValue: 10
      Type: 'FEE'
TestPercentageCustomLineItem:
  Type: AWS::BillingConductor::CustomLineItem
  Properties:
    Name: 'TestPercentageCustomLineItem'
    Description: 'Test percentage custom line item created through Cloudformation for
a %10 additional charge on the overall total bill of the billing group.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn

```

```
CustomLineItemChargeDetails:
  Percentage:
    PercentageValue: 10
  ChildAssociatedResources:
    - !GetAtt TestBillingGroup.Arn
  Type: 'FEE'
```

Trechos de modelo do CloudFormation

Tópicos

- [Pilhas aninhadas](#)
- [Condição de espera](#)

Pilhas aninhadas

Aninhar uma pilha em um modelo

Este modelo de exemplo contém um recurso de pilha aninhada chamado `myStack`. Quando cria uma pilha com base no modelo, o CloudFormation cria o `myStack`, cujo modelo é especificado na propriedade `TemplateURL`. O valor de saída `StackRef` retorna o ID de pilha de `myStack`, e o valor `OutputFromNestedStack` retorna o valor de saída `BucketName` do recurso `myStack`. O formato `Outputs.nestedstackoutputname` é reservado para especificar valores de saída de pilhas aninhadas e pode ser usado em qualquer lugar dentro do modelo que a contém.

Para obter mais informações, consulte [AWS::CloudFormation::Stack](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template",
        "TimeoutInMinutes" : "60"
      }
    }
  },
}
```

```

    "Outputs": {
      "StackRef": {"Value": { "Ref" : "myStack"}},
      "OutputFromNestedStack" : {
        "Value" : { "Fn::GetAtt" : [ "myStack", "Outputs.BucketName" ] }
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template
      TimeoutInMinutes: '60'
Outputs:
  StackRef:
    Value: !Ref myStack
  OutputFromNestedStack:
    Value: !GetAtt myStack.Outputs.BucketName

```

Aninhar uma pilha com parâmetros de entrada em um modelo

Este modelo de exemplo contém um recurso de pilha que especifica parâmetros de entrada. Quando o CloudFormation cria uma pilha com base nesse modelo, ele usa os pares de valores declarados dentro da propriedade `Parameters` como os parâmetros de entrada para o modelo usado na criação da pilha `myStackWithParams`. Neste exemplo, os parâmetros `InstanceType` e `KeyName` são especificados.

Para obter mais informações, consulte [AWS::CloudFormation::Stack](#).

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStackWithParams" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {

```

```

    "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-
east-1/EC2ChooseAMI.template",
    "Parameters" : {
        "InstanceType" : "t2.micro",
        "KeyName" : "mykey"
    }
}
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStackWithParams:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
EC2ChooseAMI.template
      Parameters:
        InstanceType: t2.micro
        KeyName: mykey

```

Condição de espera

Usar uma condição de espera com uma instância do Amazon EC2

Important

Para recursos do Amazon EC2 e do Auto Scaling, recomendamos usar um atributo `CreationPolicy`, em vez de condições de espera. Adicione um atributo `CreationPolicy` a esses recursos e use o script auxiliar `cfn-signal` para sinalizar quando um processo de criação de instância foi concluído com êxito.

Caso não possa usar uma política de criação, você visualiza o modelo de exemplo a seguir, que declara uma instância Amazon EC2 com uma condição de espera. A condição de espera `myWaitCondition` usa `myWaitConditionHandle` para sinalização, usa o atributo `DependsOn` para especificar que a condição de espera será acionada depois do recurso de instância do Amazon EC2 ter sido criado e usa a propriedade `Timeout` para especificar uma duração de 4.500 segundos

para a condição de espera. Além disso, o URL pré-assinado que sinaliza a condição de espera é passado para a instância do Amazon EC2 com a propriedade `UserData` do recurso `Ec2Instance`, o que permite que uma aplicação ou um script em execução nessa instância do Amazon EC2 recupere o URL pré-assinado e o empregue a fim de sinalizar um êxito ou uma falha para a condição de espera. Você precisa usar `cfn-signal` ou criar a aplicação ou o script que sinaliza a condição de espera. O valor de saída `ApplicationData` contém os dados repassados pelo sinal da condição de espera.

Para obter mais informações, consulte [Criar condições de espera em um modelo do CloudFormation](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0123456789abcdef0"
      },
      "us-west-1" : {
        "AMI" : "ami-0987654321fedcba0"
      },
      "eu-west-1" : {
        "AMI" : "ami-0abcdef123456789a"
      },
      "ap-northeast-1" : {
        "AMI" : "ami-0fedcba987654321b"
      },
      "ap-southeast-1" : {
        "AMI" : "ami-0c1d2e3f4a5b6c7d8"
      }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "UserData" : { "Fn::Base64" : {"Ref" : "myWaitHandle"} },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ] }
      }
    },
    "myWaitHandle" : {
```

```

        "Type" : "AWS::CloudFormation::WaitConditionHandle",
        "Properties" : {
        }
    },
    "myWaitCondition" : {
        "Type" : "AWS::CloudFormation::WaitCondition",
        "DependsOn" : "Ec2Instance",
        "Properties" : {
            "Handle" : { "Ref" : "myWaitHandle" },
            "Timeout" : "4500"
        }
    }
},
"Outputs" : {
    "ApplicationData" : {
        "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ] },
        "Description" : "The data passed back as part of signalling the
WaitCondition."
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0123456789abcdef0
    us-west-1:
      AMI: ami-0987654321fedcba0
    eu-west-1:
      AMI: ami-0abcdef123456789a
    ap-northeast-1:
      AMI: ami-0fedcba987654321b
    ap-southeast-1:
      AMI: ami-0c1d2e3f4a5b6c7d8
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle

```

```

    ImageId:
      Fn::FindInMap:
        - RegionMap
        - Ref: AWS::Region
        - AMI
    myWaitHandle:
      Type: AWS::CloudFormation::WaitConditionHandle
      Properties: {}
    myWaitCondition:
      Type: AWS::CloudFormation::WaitCondition
      DependsOn: Ec2Instance
      Properties:
        Handle: !Ref myWaitHandle
        Timeout: '4500'
  Outputs:
    ApplicationData:
      Value: !GetAtt myWaitCondition.Data
      Description: The data passed back as part of signalling the WaitCondition.

```

Usar o script auxiliar `cfn-signal` para sinalizar uma condição de espera

Este exemplo mostra uma linha de comando `cfn-signal` que sinaliza êxito para uma condição de espera. Você precisa definir a linha de comando na propriedade `UserData` da instância do EC2.

JSON

```

"UserData": {
  "Fn::Base64": {
    "Fn::Join": [
      "",
      [
        "#!/bin/bash -xe\n",
        "/opt/aws/bin/cfn-signal --exit-code 0 '",
        {
          "Ref": "myWaitHandle"
        },
        "'\n"
      ]
    ]
  }
}

```

YAML

```
UserData:
  'Fn::Base64':
    'Fn::Join':
      - ''
      - - |
          #!/bin/bash -xe
          - /opt/aws/bin/cfn-signal --exit-code 0 '
          - Ref: myWaitHandle
          - |
            ;
```

Usar Curl para sinalizar uma condição de espera

Este exemplo mostra uma linha de comando Curl que sinaliza êxito para uma condição de espera.

```
curl -T /tmp/a "https://cloudformation-waitcondition-test.s3.amazonaws.com/
arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo
%3D"
```

Em que o arquivo/tmp/a contém a seguinte estrutura JSON:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

Este exemplo mostra uma linha de comando Curl que envia o mesmo sinal de êxito, exceto por enviar o JSON como um parâmetro na linha de comando.

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" :
"Configuration Complete","UniqueId" : "ID1234","Data" : "Application
has completed configuration."}' "https://cloudformation-waitcondition-
test.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?"
```

```
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo%3D"
```

Trechos de modelo do Amazon CloudFront

Use esses trechos do modelo de exemplo com o recurso de distribuição do Amazon CloudFront no CloudFormation. Para obter mais informações, consulte [Amazon CloudFront resource type reference](#).

Tópicos

- [Recurso de distribuição do Amazon CloudFront com uma origem do Amazon S3](#)
- [Recurso de distribuição do Amazon CloudFront com origem personalizada](#)
- [Distribuição do Amazon CloudFront com suporte a múltiplas origens.](#)
- [Distribuição do Amazon CloudFront com uma função do Lambda como origem](#)
- [Consulte também](#)

Recurso de distribuição do Amazon CloudFront com uma origem do Amazon S3

O modelo de exemplo, apresentado a seguir, mostra uma [distribuição](#) do Amazon CloudFront usando um [S3Origin](#) e uma identidade do acesso de origem (OAI) herdada. Para obter informações sobre como usar o controle do acesso à origem (OAC), consulte [Restringir o acesso ao conteúdo de uma origem do Amazon Simple Storage Service](#) no Guia do desenvolvedor do Amazon CloudFront.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "Id" : "myS3Origin",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          }
        ]
      }
    }
  },
}
```

```
"Enabled" : "true",
"Comment" : "Some comment",
"DefaultRootObject" : "index.html",
"Logging" : {
    "IncludeCookies" : "false",
    "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
    "Prefix" : "myprefix"
},
"Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
"DefaultCacheBehavior" : {
    "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
    "TargetOriginId" : "myS3Origin",
    "ForwardedValues" : {
        "QueryString" : "false",
        "Cookies" : { "Forward" : "none" }
    },
    "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
    "ViewerProtocolPolicy" : "allow-all"
},
"PriceClass" : "PriceClass_200",
"Restrictions" : {
    "GeoRestriction" : {
        "RestrictionType" : "whitelist",
        "Locations" : [ "AQ", "CV" ]
    }
},
"ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
}
}
```

YAML

```
AWS::Template::FormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
```

```
- DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
  Id: myS3Origin
  S3OriginConfig:
    OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
Enabled: 'true'
Comment: Some comment
DefaultRootObject: index.html
Logging:
  IncludeCookies: 'false'
  Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
  Prefix: myprefix
Aliases:
- mysite.example.com
- yoursite.example.com
DefaultCacheBehavior:
  AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myS3Origin
  ForwardedValues:
    QueryString: 'false'
    Cookies:
      Forward: none
  TrustedSigners:
  - 1234567890EX
  - 1234567891EX
  ViewerProtocolPolicy: allow-all
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
    - AQ
    - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

Recurso de distribuição do Amazon CloudFront com origem personalizada

O seguinte modelo de exemplo mostra uma [distribuição](#) do Amazon CloudFront usando um [CustomOrigin](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "www.example.com",
            "Id" : "myCustomOrigin",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          } ],
          "Enabled" : "true",
          "Comment" : "Somecomment",
          "DefaultRootObject" : "index.html",
          "Logging" : {
            "IncludeCookies" : "true",
            "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
            "Prefix" : "myprefix"
          },
          "Aliases" : [
            "mysite.example.com",
            "*.yoursite.example.com"
          ],
          "DefaultCacheBehavior" : {
            "TargetOriginId" : "myCustomOrigin",
            "SmoothStreaming" : "false",
            "ForwardedValues" : {
              "QueryString" : "false",
              "Cookies" : { "Forward" : "all" }
            }
          },
          "TrustedSigners" : [
```



```
        "1234567890EX",
        "1234567891EX"
    ],
    "ViewerProtocolPolicy" : "allow-all"
},
"CustomErrorResponses" : [ {
    "ErrorCode" : "404",
    "ResponsePagePath" : "/error-pages/404.html",
    "ResponseCode" : "200",
    "ErrorCachingMinTTL" : "30"
} ],
"PriceClass" : "PriceClass_200",
"Restrictions" : {
    "GeoRestriction" : {
        "RestrictionType" : "whitelist",
        "Locations" : [ "AQ", "CV" ]
    }
},
"ViewerCertificate": { "CloudFrontDefaultCertificate" : "true" }
}
}
}
```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: www.example.com
            Id: myCustomOrigin
            CustomOriginConfig:
              HTTPPort: '80'
              HTTPSPort: '443'
              OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Somecomment
        DefaultRootObject: index.html

```

```

Logging:
  IncludeCookies: 'true'
  Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
  Prefix: myprefix
Aliases:
- mysite.example.com
- "*.yoursite.example.com"
DefaultCacheBehavior:
  TargetOriginId: myCustomOrigin
  SmoothStreaming: 'false'
  ForwardedValues:
    QueryString: 'false'
    Cookies:
      Forward: all
  TrustedSigners:
    - 1234567890EX
    - 1234567891EX
  ViewerProtocolPolicy: allow-all
CustomErrorResponses:
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
      - AQ
      - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'

```

Distribuição do Amazon CloudFront com suporte a múltiplas origens.

O seguinte exemplo mostra como declarar uma [distribuição](#) do CloudFront com suporte para múltiplas origens. Em [DistributionConfig](#), uma lista de origens é fornecida e um [DefaultCacheBehavior](#) é definido.

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",

```

```

"Resources" : {
  "myDistribution" : {
    "Type" : "AWS::CloudFront::Distribution",
    "Properties" : {
      "DistributionConfig" : {
        "Origins" : [ {
          "Id" : "myS3Origin",
          "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
          "S3OriginConfig" : {
            "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
          }
        },
        {
          "Id" : "myCustomOrigin",
          "DomainName" : "www.example.com",
          "CustomOriginConfig" : {
            "HTTPPort" : "80",
            "HTTPSPort" : "443",
            "OriginProtocolPolicy" : "http-only"
          }
        }
      ],
      "Enabled" : "true",
      "Comment" : "Some comment",
      "DefaultRootObject" : "index.html",
      "Logging" : {
        "IncludeCookies" : "true",
        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
        "Prefix" : "myprefix"
      },
      "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
      "DefaultCacheBehavior" : {
        "TargetOriginId" : "myS3Origin",
        "ForwardedValues" : {
          "QueryString" : "false",
          "Cookies" : { "Forward" : "all" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "100",
        "SmoothStreaming" : "true"
      },
      "CacheBehaviors" : [ {

```

```
"PATCH", "POST", "PUT" ],
    "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
    "TargetOriginId" : "myS30origin",
    "ForwardedValues" : {
        "QueryString" : "true",
        "Cookies" : { "Forward" : "none" }
    },
    "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
    "ViewerProtocolPolicy" : "allow-all",
    "MinTTL" : "50",
    "PathPattern" : "images1/*.jpg"
},
{
    "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
    "PATCH", "POST", "PUT" ],
    "TargetOriginId" : "myCustomOrigin",
    "ForwardedValues" : {
        "QueryString" : "true",
        "Cookies" : { "Forward" : "none" }
    },
    "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
    "ViewerProtocolPolicy" : "allow-all",
    "MinTTL" : "50",
    "PathPattern" : "images2/*.jpg"
}
],
"CustomErrorResponses" : [ {
    "ErrorCode" : "404",
    "ResponsePagePath" : "/error-pages/404.html",
    "ResponseCode" : "200",
    "ErrorCachingMinTTL" : "30"
} ],
"PriceClass" : "PriceClass_All",
"ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
}
}
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

Resources:**myDistribution:****Type:** AWS::CloudFront::Distribution**Properties:****DistributionConfig:****Origins:**

- Id: myS3Origin

DomainName: amzn-s3-demo-bucket.s3.amazonaws.com**S3OriginConfig:****OriginAccessIdentity:** origin-access-identity/cloudfront/E127EXAMPLE51Z

- Id: myCustomOrigin

DomainName: www.example.com**CustomOriginConfig:****HTTPPort:** '80'**HTTPSPort:** '443'**OriginProtocolPolicy:** http-only**Enabled:** 'true'**Comment:** Some comment**DefaultRootObject:** index.html**Logging:****IncludeCookies:** 'true'**Bucket:** amzn-s3-demo-logging-bucket.s3.amazonaws.com**Prefix:** myprefix**Aliases:**

- mysite.example.com
- yoursite.example.com

DefaultCacheBehavior:**TargetOriginId:** myS3Origin**ForwardedValues:****QueryString:** 'false'**Cookies:****Forward:** all**TrustedSigners:**

- 1234567890EX
- 1234567891EX

ViewerProtocolPolicy: allow-all**MinTTL:** '100'**SmoothStreaming:** 'true'**CacheBehaviors:**

- AllowedMethods:

- DELETE
- GET
- HEAD
- OPTIONS

```
- PATCH
- POST
- PUT
TargetOriginId: myS3Origin
ForwardedValues:
  QueryString: 'true'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '50'
PathPattern: images1/*.jpg
- AllowedMethods:
- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT
TargetOriginId: myCustomOrigin
ForwardedValues:
  QueryString: 'true'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '50'
PathPattern: images2/*.jpg
CustomErrorResponses:
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_All
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

Distribuição do Amazon CloudFront com uma função do Lambda como origem

O exemplo a seguir cria uma distribuição do CloudFront que apresenta um URL específico da função do Lambda (fornecida como parâmetro), possibilitando o acesso somente por HTTPS, o armazenamento em cache, a compactação e a entrega global. Ele configura o URL do Lambda como uma origem HTTPS personalizada e aplica uma política de armazenamento em cache da AWS padrão. A distribuição é otimizada para performance com suporte a HTTP/2 e IPv6 e gera o nome de domínio do CloudFront, permitindo que os usuários acessem a função do Lambda por meio de um endpoint seguro baseado em CDN. Para obter mais informações, consulte [Using Amazon CloudFront with AWS Lambda as origin to accelerate your web applications](#) no Blog da AWS.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "LambdaEndpoint": {
      "Type": "String",
      "Description": "The Lambda function URL endpoint without the 'https://'"
    }
  },
  "Resources": {
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "PriceClass": "PriceClass_All",
          "HttpVersion": "http2",
          "IPV6Enabled": true,
          "Origins": [
            {
              "DomainName": {
                "Ref": "LambdaEndpoint"
              },
              "Id": "LambdaOrigin",
              "CustomOriginConfig": {
                "HTTPSPort": 443,
                "OriginProtocolPolicy": "https-only"
              }
            }
          ]
        },
        "Enabled": "true",
```

```

        "DefaultCacheBehavior": {
            "TargetOriginId": "LambdaOrigin",
            "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
            "ViewerProtocolPolicy": "redirect-to-https",
            "SmoothStreaming": "false",
            "Compress": "true"
        }
    },
    "Outputs": {
        "CloudFrontDomain": {
            "Description": "CloudFront default domain name configured",
            "Value": {
                "Fn::Sub": "https://${MyDistribution.DomainName}/"
            }
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  LambdaEndpoint:
    Type: String
    Description: The Lambda function URL endpoint without the 'https://'
Resources:
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        PriceClass: PriceClass_All
        HttpVersion: http2
        IPV6Enabled: true
        Origins:
          - DomainName: !Ref LambdaEndpoint
            Id: LambdaOrigin
            CustomOriginConfig:
              HTTPSPort: 443
              OriginProtocolPolicy: https-only
        Enabled: 'true'

```



```
DefaultCacheBehavior:
  TargetOriginId: LambdaOrigin
  CachePolicyId: '658327ea-f89d-4fab-a63d-7e88639e58f6'
  ViewerProtocolPolicy: redirect-to-https
  SmoothStreaming: 'false'
  Compress: 'true'
```

Outputs:**CloudFrontDomain:**

```
Description: CloudFront default domain name configured
Value: !Sub https://${MyDistribution.DomainName}/
```

Consulte também

Para obter um exemplo de adição de um alias personalizado a um registro do Route 53 para criar um nome amigável para uma distribuição do CloudFront, consulte [Conjunto de registros de recursos de alias para uma distribuição do CloudFront](#).

Trechos de modelo do Amazon CloudWatch

Use trechos do modelo de exemplo para ajudar a descrever os recursos do Amazon CloudWatch nos modelos do CloudFormation. Para obter mais informações, consulte [Referência de tipos de recursos do Amazon CloudWatch](#).

Tópicos

- [Alarme de faturamento](#)
- [Alarme de utilização da CPU](#)
- [Recuperar uma instância Amazon Elastic Compute Cloud](#)
- [Criar um painel básico](#)
- [Criar um painel com widgets exibidos lado a lado](#)

Alarme de faturamento

No exemplo a seguir, o Amazon CloudWatch envia uma notificação por e-mail quando as despesas de sua conta da AWS excedem o limite de alarme. Para receber notificações de uso, basta permitir os alertas de pagamento. Consulte mais informações em [Criar um alarme de faturamento para monitorar suas cobranças estimadas da AWS](#) no Guia do usuário do Amazon CloudWatch.

JSON

```
"SpendingAlarm": {
  "Type": "AWS::CloudWatch::Alarm",
  "Properties": {
    "AlarmDescription": { "Fn::Join": [ "", [
      "Alarm if AWS spending is over $",
      { "Ref": "AlarmThreshold" }
    ] ] },
    "Namespace": "AWS/Billing",
    "MetricName": "EstimatedCharges",
    "Dimensions": [{
      "Name": "Currency",
      "Value": "USD"
    } ],
    "Statistic": "Maximum",
    "Period": "21600",
    "EvaluationPeriods": "1",
    "Threshold": { "Ref": "AlarmThreshold" },
    "ComparisonOperator": "GreaterThanOrEqualToThreshold",
    "AlarmActions": [{
      "Ref": "BillingAlarmNotification"
    } ],
    "InsufficientDataActions": [{
      "Ref": "BillingAlarmNotification"
    } ]
  }
}
```

YAML

```
SpendingAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription:
      'Fn::Join':
        - ''
        - - Alarm if AWS spending is over $
          - !Ref: AlarmThreshold
    Namespace: AWS/Billing
    MetricName: EstimatedCharges
    Dimensions:
      - Name: Currency
```

```

    Value: USD
    Statistic: Maximum
    Period: '21600'
    EvaluationPeriods: '1'
    Threshold:
      !Ref: "AlarmThreshold"
    ComparisonOperator: GreaterThanThreshold
    AlarmActions:
      - !Ref: "BillingAlarmNotification"
    InsufficientDataActions:
      - !Ref: "BillingAlarmNotification"

```

Alarme de utilização da CPU

O trecho de exemplo a seguir cria um alarme que envia uma notificação quando a utilização média da CPU de uma instância Amazon EC2 é excedida em 90% por mais de 60 segundos em três períodos de avaliação.

JSON

```

"CPUALarm" : {
  "Type" : "AWS::CloudWatch::Alarm",
  "Properties" : {
    "AlarmDescription" : "CPU alarm for my instance",
    "AlarmActions" : [ { "Ref" : "logical name of an AWS::SNS::Topic resource" } ],
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "60",
    "EvaluationPeriods" : "3",
    "Threshold" : "90",
    "ComparisonOperator" : "GreaterThanThreshold",
    "Dimensions" : [ {
      "Name" : "InstanceId",
      "Value" : { "Ref" : "logical name of an AWS::EC2::Instance resource" }
    } ]
  }
}

```

YAML

```

CPUALarm:
  Type: AWS::CloudWatch::Alarm

```

```

Properties:
  AlarmDescription: CPU alarm for my instance
  AlarmActions:
    - !Ref: "logical name of an AWS::SNS::Topic resource"
  MetricName: CPUUtilization
  Namespace: AWS/EC2
  Statistic: Average
  Period: '60'
  EvaluationPeriods: '3'
  Threshold: '90'
  ComparisonOperator: GreaterThanThreshold
  Dimensions:
    - Name: InstanceId
      Value: !Ref: "logical name of an AWS::EC2::Instance resource"

```

Recuperar uma instância Amazon Elastic Compute Cloud

O alarme do CloudWatch a seguir recupera uma instância do EC2 quando ocorrem falhas de verificação de status por 15 minutos consecutivos. Consulte mais informações sobre ações de alarme em [Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2](#) no Guia do usuário do Amazon CloudWatch.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "RecoveryInstance" : {
      "Description" : "The EC2 instance ID to associate this alarm with.",
      "Type" : "AWS::EC2::Instance::Id"
    }
  },
  "Resources": {
    "RecoveryTestAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Trigger a recovery when instance status check fails for 15 consecutive minutes.",
        "Namespace": "AWS/EC2" ,
        "MetricName": "StatusCheckFailed_System",
        "Statistic": "Minimum",
        "Period": "60",
        "EvaluationPeriods": "15",

```

```

        "ComparisonOperator": "GreaterThanThreshold",
        "Threshold": "0",
        "AlarmActions": [ {"Fn::Join" : [ "", [ "arn:aws:automate:", { "Ref" :
"AWS::Region" }, ":ec2:recover" ] ] } ],
        "Dimensions": [ { "Name": "InstanceId", "Value": { "Ref": "RecoveryInstance" } } ]
    }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  RecoveryInstance:
    Description: The EC2 instance ID to associate this alarm with.
    Type: AWS::EC2::Instance::Id
Resources:
  RecoveryTestAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: Trigger a recovery when instance status check fails for 15
        consecutive minutes.
      Namespace: AWS/EC2
      MetricName: StatusCheckFailed_System
      Statistic: Minimum
      Period: '60'
      EvaluationPeriods: '15'
      ComparisonOperator: GreaterThanThreshold
      Threshold: '0'
      AlarmActions: [ !Sub "arn:aws:automate:${AWS::Region}:ec2:recover" ]
      Dimensions:
        - Name: InstanceId
          Value: !Ref: RecoveryInstance

```

Criar um painel básico

O exemplo a seguir cria um painel simples do CloudWatch com um widget de métrica que exibe a utilização da CPU e um widget de texto que exibe uma mensagem.

JSON

```
{
```

```

    "BasicDashboard": {
      "Type": "AWS::CloudWatch::Dashboard",
      "Properties": {
        "DashboardName": "Dashboard1",
        "DashboardBody": "{\n  \"widgets\": [\n    {\n      \"type\": \"metric\",\n      \"x\": 0,\n      \"y\": 0,\n      \"width\": 12,\n      \"height\": 6,\n      \"properties\": {\n        \"metrics\": [\n          [\n            \"AWS/EC2\",\n            \"CPUUtilization\",\n            \"InstanceId\",\n            \"i-012345\"\n          ],\n          {\n            \"period\": 300,\n            \"stat\": \"Average\",\n            \"region\": \"us-east-1\",\n            \"title\": \"EC2 Instance CPU\"\n          }\n        ],\n        \"type\": \"text\",\n        \"x\": 0,\n        \"y\": 7,\n        \"width\": 3,\n        \"height\": 3,\n        \"properties\": {\n          \"markdown\": \"Hello world\"\n        }\n      }\n    }\n  ]\n}"
      }
    }
  }
}

```

YAML

```

BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type": "metric", "x": 0, "y": 0, "width": 12, "height": 6, "properties": {"metrics": [{"AWS/
EC2", "CPUUtilization", "InstanceId", "i-012345"}], "period": 300, "stat": "Average", "region": "us-
east-1", "title": "EC2 Instance CPU"}},
{"type": "text", "x": 0, "y": 7, "width": 3, "height": 3, "properties": {"markdown": "Hello
world"}}]}'

```

Criar um painel com widgets exibidos lado a lado

O exemplo a seguir cria um painel com dois widgets de métrica exibidos lado a lado.

JSON

```

{
  "DashboardSideBySide": {
    "Type": "AWS::CloudWatch::Dashboard",
    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\n  \"widgets\": [\n    {\n      \"type\": \"metric\",\n      \"x\": 0,\n      \"y\": 0,\n      \"width\": 12,\n      \"height\": 6,\n      \"properties\": {\n        \"metrics\": [\n          [\n            \"AWS/EC2\",\n            \"CPUUtilization\",\n            \"InstanceId\",\n            \"i-012345\"\n          ],\n          {\n            \"period\": 300,\n            \"stat\": \"Average\",\n            \"region\": \"us-east-1\",\n            \"title\": \"EC2 Instance CPU\"\n          }\n        ],\n        \"type\": \"metric\",\n        \"x\": 12,\n        \"y\": 0,\n        \"width\": 12,\n        \"height\": 6,\n        \"properties\": {\n          \"metrics\": [\n            [\n              \"AWS/S3\",\n              \"BucketSizeBytes\",\n              \"BucketName\",\n              \"arn:aws:s3:::mybucket\"\n            ],\n            {\n              \"period\": 300,\n              \"stat\": \"Average\",\n              \"region\": \"us-east-1\",\n              \"title\": \"S3 Bucket Size\"\n            }\n          ]\n        }\n      }\n    }\n  ]\n}"
    }
  }
}

```

```
\\"BucketName\\",\\"amzn-s3-demo-bucket\\"]],\"period\\":86400,\"stat\\":\\"Maximum\\",\\"region\\":\\"us-east-1\\",\\"title\\":\\"amzn-s3-demo-bucket bytes\\"]]]]"
    }
  }
}
```

YAML

```
DashboardSideBySide:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]],"period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"metric","x":12,"y":0,"width":12,"height":6,"properties":
{"metrics":[["AWS/S3","BucketSizeBytes","BucketName","amzn-s3-demo-
bucket"]],"period":86400,"stat":"Maximum","region":"us-east-1","title":"amzn-s3-demo-
bucket bytes"}]]}'
```

Trechos de modelo do Amazon CloudWatch Logs

O Amazon CloudWatch Logs pode monitorar seu sistema, aplicativos e arquivos de log personalizados em instâncias do Amazon EC2 ou outras origens. Você pode usar o CloudFormation para provisionar e gerenciar grupos de logs e filtros de métricas. Para obter mais informações sobre o CloudWatch Logs, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

Tópicos

- [Enviar logs para o CloudWatch Logs de uma instância do Linux](#)
- [Enviar logs para o CloudWatch Logs de uma instância do Windows](#)
- [Consulte também](#)

Enviar logs para o CloudWatch Logs de uma instância do Linux

O modelo a seguir demonstra como configurar um servidor Web no Amazon Linux 2023 com a integração com o CloudWatch Logs. O modelo executa as seguintes tarefas:

- Instala o Apache e o PHP.
- Configura o agente do CloudWatch para encaminhar logs de acesso do Apache para o CloudWatch Logs.
- Configura um perfil do IAM para permitir que o agente do CloudWatch envie dados de log ao CloudWatch Logs.
- Cria alarmes e notificações personalizados para monitorar erros 404 ou alto uso de largura de banda.

Os eventos de logs do servidor Web fornecem dados de métricas para os alarmes do CloudWatch. Os dois filtros de métrica descrevem como as informações de log são transformadas em métricas do CloudWatch. A métrica 404 conta o número de ocorrências 404. A métrica de tamanho controla o tamanho de uma solicitação. Os dois alarmes do CloudWatch enviarão notificações se houver mais que duas 404s em 2 minutos ou se o tamanho médio da solicitação for superior a 3500 KB por 10 minutos.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on Amazon Linux 2023 instance.",
  "Parameters": {
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  }
}
```



```

    "OperatorEmail": {
      "Description": "Email address to notify when CloudWatch alarms are
triggered (404 errors or high bandwidth usage)",
      "Type": "String"
    }
  },
  "Resources": {
    "LogRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  "ec2.amazonaws.com"
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyName": "LogRolePolicy",
            "PolicyDocument": {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Effect": "Allow",
                  "Action": [
                    "logs:PutLogEvents",
                    "logs:DescribeLogStreams",
                    "logs:DescribeLogGroups",
                    "logs:CreateLogGroup",
                    "logs:CreateLogStream"
                  ],
                  "Resource": "*"
                }
              ]
            }
          }
        ]
      }
    }
  }
}

```

```

    ]
  }
}
]
}
},
"LogRoleInstanceProfile": {
  "Type": "AWS::IAM::InstanceProfile",
  "Properties": {
    "Path": "/",
    "Roles": [{"Ref": "LogRole"}]
  }
},
"WebServerSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Enable HTTP access via port 80 and SSH access via
port 22",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "CidrIp": {"Ref": "SSHLocation"}
      }
    ]
  }
},
"WebServerHost": {
  "Type": "AWS::EC2::Instance",
  "Metadata": {
    "Comment": "Install a simple PHP application on Amazon Linux 2023",
    "AWS::CloudFormation::Init": {
      "config": {
        "packages": {
          "dnf": {
            "httpd": [],
            "php": [],

```

```

        "php-fpm": []
    },
    "files": {
        "/etc/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.json": {
            "content": {
                "logs": {
                    "logs_collected": {
                        "files": {
                            "collect_list": [{
                                "file_path": "/var/log/httpd/
access_log",
                                "log_group_name": {"Ref":
"WebServerLogGroup"},
                                "log_stream_name": "{instance_id}/
apache.log",
                                "timestamp_format": "%d/%b/%Y:%H:
%M:%S %z"
                            }]
                        }
                    }
                },
                "mode": "000644",
                "owner": "root",
                "group": "root"
            },
            "/var/www/html/index.php": {
                "content": "<?php\\necho '<h1>AWS CloudFormation sample
PHP application on Amazon Linux 2023</h1>';\\n?>\\n",
                "mode": "000644",
                "owner": "apache",
                "group": "apache"
            },
            "/etc/cfn/cfn-hup.conf": {
                "content": {
                    "Fn::Join": [
                        "",
                        [
                            "[main]\\n",
                            "stack=",
                            {"Ref": "AWS::StackId"},
                            "\\n",

```

```

        "region=",
        {"Ref": "AWS::Region"},
        "\n"
    ]
]
},
"mode": "000400",
"owner": "root",
"group": "root"
},
"/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
    "content": {
        "Fn::Join": [
            "",
            [
                "[cfn-auto-reloader-hook]\n",
                "triggers=post.update\n",
                "\n",
                "path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
                "action=/opt/aws/bin/cfn-init -s ",
                {"Ref": "AWS::StackId"},
                " -r WebServerHost ",
                "--region ",
                {"Ref": "AWS::Region"},
                "\n",
                "runas=root\n"
            ]
        ]
    }
},
"services": {
    "systemd": {
        "httpd": {
            "enabled": "true",
            "ensureRunning": "true"
        },
        "php-fpm": {
            "enabled": "true",
            "ensureRunning": "true"
        }
    }
}
}

```

```

    }
  },
  "CreationPolicy": {
    "ResourceSignal": {
      "Timeout": "PT5M"
    }
  },
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
    "KeyName": {"Ref": "KeyName"},
    "InstanceType": "t3.micro",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {"Fn::Base64": {"Fn::Join": [ "", [
      "#!/bin/bash\n",
      "dnf update -y aws-cfn-bootstrap\n",
      "dnf install -y amazon-cloudwatch-agent\n",
      "/opt/aws/bin/cfn-init -v --stack ", {"Ref": "AWS::StackName"}, "
--resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n",
      "\n",
      "# Verify Apache log directory exists and create if needed\n",
      "mkdir -p /var/log/httpd\n",
      "\n",
      "# Start CloudWatch agent\n",
      "/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -
a fetch-config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json
-s\n",
      "\n",
      "# Signal success\n",
      "/opt/aws/bin/cfn-signal -e $? --stack ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n"
]]]}
  }
},
  "WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "Properties": {
      "RetentionInDays": 7
    }
  },
  "404MetricFilter": {

```

```

        "Type": "AWS::Logs::MetricFilter",
        "Properties": {
            "LogGroupName": {"Ref": "WebServerLogGroup"},
            "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code = 404, size, ...]",
            "MetricTransformations": [
                {
                    "MetricValue": "1",
                    "MetricNamespace": "test/404s",
                    "MetricName": "test404Count"
                }
            ]
        }
    },
    "BytesTransferredMetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {
            "LogGroupName": {"Ref": "WebServerLogGroup"},
            "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code, size, ...]",
            "MetricTransformations": [
                {
                    "MetricValue": "$size",
                    "MetricNamespace": "test/BytesTransferred",
                    "MetricName": "testBytesTransferred"
                }
            ]
        }
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },

```

```

    "BandwidthAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "The average volume of traffic is greater 3500 KB
over 10 minutes",
        "MetricName": "testBytesTransferred",
        "Namespace": "test/BytesTransferred",
        "Statistic": "Average",
        "Period": "300",
        "EvaluationPeriods": "2",
        "Threshold": "3500",
        "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
        "ComparisonOperator": "GreaterThanThreshold"
      }
    },
    "AlarmNotificationTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Description": "The instance ID of the web server",
      "Value": {"Ref": "WebServerHost"}
    },
    "WebsiteURL": {
      "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
      "Description": "URL for the web server"
    },
    "PublicIP": {
      "Description": "Public IP address of the web server",
      "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
    },
    "CloudWatchLogGroupName": {
      "Description": "The name of the CloudWatch log group",
      "Value": {"Ref": "WebServerLogGroup"}
    }
  }
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Sample template that sets up and configures CloudWatch Logs on Amazon
Linux 2023 instance.
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify when CloudWatch alarms are triggered (404
errors or high bandwidth usage)
    Type: String
Resources:
  LogRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Path: /
    Policies:
      - PolicyName: LogRolePolicy
        PolicyDocument:
          Version: 2012-10-17
          Statement:
            - Effect: Allow
              Action:
                - 'logs:PutLogEvents'
```



```

        - 'logs:DescribeLogStreams'
        - 'logs:DescribeLogGroups'
        - 'logs:CreateLogGroup'
        - 'logs:CreateLogStream'
    Resource: '*'
LogRoleInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref LogRole
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
WebServerHost:
  Type: AWS::EC2::Instance
  Metadata:
    Comment: Install a simple PHP application on Amazon Linux 2023
    'AWS::CloudFormation::Init':
      config:
        packages:
          dnf:
            httpd: []
            php: []
            php-fpm: []
        files:
          /etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json:
            content: !Sub |
              {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [
                        {

```

```

        "file_path": "/var/log/httpd/access_log",
        "log_group_name": "${WebServerLogGroup}",
        "log_stream_name": "${instance_id}/apache.log",
        "timestamp_format": "%d/%b/%Y:%H:%M:%S %z"
    }
  ]
}
}
}
mode: '000644'
owner: root
group: root
/var/www/html/index.php:
content: |
    <?php echo '<h1>AWS CloudFormation sample PHP application on Amazon
Linux 2023</h1>';
    ?>
mode: '000644'
owner: apache
group: apache
/etc/cfn/cfn-hup.conf:
content: !Sub |
    [main]
    stack=${AWS::StackId}
    region=${AWS::Region}
mode: '000400'
owner: root
group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
content: !Sub |
    [cfn-auto-reloader-hook]
    triggers=post.update
    path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
    action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerHost --
region ${AWS::Region}
    runas=root
services:
  systemd:
    httpd:
      enabled: 'true'
      ensureRunning: 'true'
    php-fpm:
      enabled: 'true'

```

```

        ensureRunning: 'true'
CreationPolicy:
  ResourceSignal:
    Timeout: PT5M
Properties:
  ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
  KeyName: !Ref KeyName
  InstanceType: t3.micro
  SecurityGroupIds:
    - !Ref WebServerSecurityGroup
  IamInstanceProfile: !Ref LogRoleInstanceProfile
  UserData: !Base64
    Fn::Sub: |
      #!/bin/bash
      dnf update -y aws-cfn-bootstrap
      dnf install -y amazon-cloudwatch-agent
      /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServerHost
--region ${AWS::Region}

      # Verify Apache log directory exists and create if needed
      mkdir -p /var/log/httpd

      # Start CloudWatch agent
      /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json -s

      # Signal success
      /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerHost --region ${AWS::Region}
      echo "Done"
WebServerLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    RetentionInDays: 7
404MetricFilter:
  Type: AWS::Logs::MetricFilter
  Properties:
    LogGroupName: !Ref WebServerLogGroup
    FilterPattern: >-
      [ip, identity, user_id, timestamp, request, status_code = 404, size, ...]
    MetricTransformations:

```

```
- MetricValue: '1'
  MetricNamespace: test/404s
  MetricName: test404Count
BytesTransferredMetricFilter:
  Type: AWS::Logs::MetricFilter
  Properties:
    LogGroupName: !Ref WebServerLogGroup
    FilterPattern: '[ip, identity, user_id, timestamp, request, status_code,
size, ...]'
    MetricTransformations:
      - MetricValue: $size
        MetricNamespace: test/BytesTransferred
        MetricName: testBytesTransferred
404Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
BandwidthAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The average volume of traffic is greater 3500 KB over 10
minutes
    MetricName: testBytesTransferred
    Namespace: test/BytesTransferred
    Statistic: Average
    Period: '300'
    EvaluationPeriods: '2'
    Threshold: '3500'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
AlarmNotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
```

```
- Endpoint: !Ref OperatorEmail
  Protocol: email
```

Outputs:

```
InstanceId:
  Description: The instance ID of the web server
  Value: !Ref WebServerHost
WebsiteURL:
  Value: !Sub 'http://${WebServerHost.PublicDnsName}'
  Description: URL for the web server
PublicIP:
  Description: Public IP address of the web server
  Value: !GetAtt WebServerHost.PublicIp
CloudWatchLogGroupName:
  Description: The name of the CloudWatch log group
  Value: !Ref WebServerLogGroup
```

Enviar logs para o CloudWatch Logs de uma instância do Windows

O modelo a seguir configura o CloudWatch Logs para uma instância do Windows 2012R2.

O agente do CloudWatch Logs no Windows (agente SSM em AMIs do Windows 2012R2 e Windows 2016) só envia os logs depois que ele é iniciado, de maneira que os logs gerados antes da inicialização não são enviados. Para resolver isso, o modelo ajuda a garantir que o agente inicie antes que qualquer log seja gravado ao:

- Configurar o agente como o primeiro item config no configSets do cfn-init.
- Usar o waitAfterCompletion para inserir uma pausa depois do comando que inicia o agente.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on
Windows 2012R2 instance.",
  "Parameters": {
    "KeyPair": {
      "Description": "Name of an existing EC2 KeyPair to enable RDP access to the
instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "RDPLocation": {
```

```
"Description": "The IP address range that can be used to RDP to the EC2 instances",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,2})|256)|255)|254)|253)|252)|251)|250)|249)|248)|247)|246)|245)|244)|243)|242)|241)|240)|239)|238)|237)|236)|235)|234)|233)|232)|231)|230)|229)|228)|227)|226)|225)|224)|223)|222)|221)|220)|219)|218)|217)|216)|215)|214)|213)|212)|211)|210)|209)|208)|207)|206)|205)|204)|203)|202)|201)|200)|199)|198)|197)|196)|195)|194)|193)|192)|191)|190)|189)|188)|187)|186)|185)|184)|183)|182)|181)|180)|179)|178)|177)|176)|175)|174)|173)|172)|171)|170)|169)|168)|167)|166)|165)|164)|163)|162)|161)|160)|159)|158)|157)|156)|155)|154)|153)|152)|151)|150)|149)|148)|147)|146)|145)|144)|143)|142)|141)|140)|139)|138)|137)|136)|135)|134)|133)|132)|131)|130)|129)|128)|127)|126)|125)|124)|123)|122)|121)|120)|119)|118)|117)|116)|115)|114)|113)|112)|111)|110)|109)|108)|107)|106)|105)|104)|103)|102)|101)|100)|99)|98)|97)|96)|95)|94)|93)|92)|91)|90)|89)|88)|87)|86)|85)|84)|83)|82)|81)|80)|79)|78)|77)|76)|75)|74)|73)|72)|71)|70)|69)|68)|67)|66)|65)|64)|63)|62)|61)|60)|59)|58)|57)|56)|55)|54)|53)|52)|51)|50)|49)|48)|47)|46)|45)|44)|43)|42)|41)|40)|39)|38)|37)|36)|35)|34)|33)|32)|31)|30)|29)|28)|27)|26)|25)|24)|23)|22)|21)|20)|19)|18)|17)|16)|15)|14)|13)|12)|11)|10)|9)|8)|7)|6)|5)|4)|3)|2)|1)",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "OperatorEmail": {
    "Description": "Email address to notify when CloudWatch alarms are triggered (404 errors)",
    "Type": "String"
  }
},
"Resources": {
  "WebServerSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable HTTP access via port 80 and RDP access via port 3389",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": "80",
          "ToPort": "80",
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": "3389",
          "ToPort": "3389",
          "CidrIp": {"Ref": "RDPLocation"}
        }
      ]
    }
  },
  "LogRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
```

```

        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "ec2.amazonaws.com"
                    ]
                },
                "Action": [
                    "sts:AssumeRole"
                ]
            }
        ],
        "ManagedPolicyArns": [
            "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
        ],
        "Path": "/",
        "Policies": [
            {
                "PolicyName": "LogRolePolicy",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "logs:Create*",
                                "logs:PutLogEvents",
                                "s3:GetObject"
                            ],
                            "Resource": [
                                "arn:aws:logs:*:*:*",
                                "arn:aws:s3:::*"
                            ]
                        }
                    ]
                }
            }
        ]
    },
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",

```

```

        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        },
        "WebServerHost": {
            "Type": "AWS::EC2::Instance",
            "CreationPolicy": {
                "ResourceSignal": {
                    "Timeout": "PT15M"
                }
            },
            "Metadata": {
                "AWS::CloudFormation::Init": {
                    "configSets": {
                        "config": [
                            "00-ConfigureCWLogs",
                            "01-InstallWebServer",
                            "02-ConfigureApplication",
                            "03-Finalize"
                        ]
                    },
                    "00-ConfigureCWLogs": {
                        "files": {
                            "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\\
\\AWS.EC2.Windows.CloudWatch.json": {
                                "content": {
                                    "EngineConfiguration": {
                                        "Components": [
                                            {
                                                "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                                                "Id": "ApplicationEventLog",
                                                "Parameters": {
                                                    "Levels": "7",
                                                    "LogName": "Application"
                                                }
                                            },
                                            {
                                                "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                                                "Id": "SystemEventLog",
                                                "Parameters": {
                                                    "Levels": "7",

```



```

        "LogName": "System"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "SecurityEventLog",
        "Parameters": {
            "Levels": "7",
            "LogName": "Security"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\Program
Files\\Amazon\\Ec2ConfigService\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-
ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\cfn\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd
HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",

```

```

        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\inetpub\\
\\logs\\LogFiles\\W3SVC1",

            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-dd
HH:mm:ss"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
ApplicationEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",

```

```

        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
SystemEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
SecurityEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
EC2ConfigLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
CfnInitLog",

```

```

        "Region": {"Ref": "AWS::Region"},
        "SecretKey": ""
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
IISLogs",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
],
"Flows": {
    "Flows": [

"ApplicationEventLog,CloudWatchApplicationEventLog",

"SystemEventLog,CloudWatchSystemEventLog",

"SecurityEventLog,CloudWatchSecurityEventLog",

"EC2ConfigLog,CloudWatchEC2ConfigLog",
"CfnInitLog,CloudWatchCfnInitLog",
"IISLogs,CloudWatchIISLogs",
"MemoryPerformanceCounter,CloudWatch"

    ]
},
"PollInterval": "00:00:05"

```

```

        },
        "IsEnabled": true
    }
}
},
"commands": {
    "0-enableSSM": {
        "command": "powershell.exe -Command \"Set-Service -Name
AmazonSSMAgent -StartupType Automatic\" ",
        "waitAfterCompletion": "0"
    },
    "1-restartSSM": {
        "command": "powershell.exe -Command \"Restart-Service
AmazonSSMAgent \"",
        "waitAfterCompletion": "30"
    }
},
"01-InstallWebServer": {
    "commands": {
        "01_install_webserver": {
            "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server -IncludeAllSubFeature\"",
            "waitAfterCompletion": "0"
        }
    }
},
"02-ConfigureApplication": {
    "files": {
        "c:\\Inetpub\\wwwroot\\index.htm": {
            "content": "<html> <head> <title>Test Application
Page</title> </head> <body> <h1>Congratulations!! Your IIS server is configured.</h1>
</body> </html>"
        }
    }
},
"03-Finalize": {
    "commands": {
        "00_signal_success": {
            "command": {
                "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
            },
            "waitAfterCompletion": "0"
        }
    }
}

```

```

        }
    }
}
},
"Properties": {
    "KeyName": {
        "Ref": "KeyPair"
    },
    "ImageId": "{{resolve:ssm:/aws/service/ami-windows-latest/
Windows_Server-2012-R2_RTM-English-64Bit-Base}}",
    "InstanceType": "t2.xlarge",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {
        "Fn::Base64": {
            "Fn::Join": [
                "",
                [
                    "<script>\n",
                    "wmic product where \"description='Amazon SSM Agent' \"
uninstall\n",
                    "wmic product where \"description='aws-cfn-bootstrap'
\" uninstall \n",
                    "start /wait c:\\\\Windows\\\\system32\\\\msiexec /passive /
qn /i https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi\n",
                    "powershell.exe -Command \"iwr https://
s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:\\\\AmazonSSMAgentSetup.exe\\\"\\n",
                    "start /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\n",
                    "cfn-init.exe -v -c config -s ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "
\n",
                    "</script>\n"
                ]
            ]
        }
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",

```

```

        "Properties": {
            "RetentionInDays": 7
        }
    },
    "404MetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {
            "LogGroupName": {"Ref": "LogGroup"},
            "FilterPattern": "[timestamps, serverip, method, uri, query, port,
dash, clientip, useragent, status_code = 404, ...]",
            "MetricTransformations": [
                {
                    "MetricValue": "1",
                    "MetricNamespace": "test/404s",
                    "MetricName": "test404Count"
                }
            ]
        }
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    }
},
"Outputs": {
    "InstanceId": {

```

```

        "Description": "The instance ID of the web server",
        "Value": {"Ref": "WebServerHost"}
    },
    "WebsiteURL": {
        "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
        "Description": "URL for the web server"
    },
    "PublicIP": {
        "Description": "Public IP address of the web server",
        "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
    },
    "CloudWatchLogGroupName": {
        "Description": "The name of the CloudWatch log group",
        "Value": {"Ref": "LogGroup"}
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch Logs on Windows 2012R2
  instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify when CloudWatch alarms are triggered (404
    errors)
    Type: String
Resources:
  WebServerSecurityGroup:

```


Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389

SecurityGroupIngress:

- IpProtocol: tcp
FromPort: '80'
ToPort: '80'
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
FromPort: '3389'
ToPort: '3389'
CidrIp: !Ref RDPLocation

LogRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow
Principal:
Service:
 - ec2.amazonaws.com

Action:

- 'sts:AssumeRole'

ManagedPolicyArns:

- 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'

Path: /

Policies:

- PolicyName: LogRolePolicy

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow
Action:
 - 'logs:Create*'
 - 'logs:PutLogEvents'
 - 's3:GetObject'
Resource:
 - 'arn:aws:logs:*:*:*'
 - 'arn:aws:s3:::*'

LogRoleInstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Path: /

```

    Roles:
      - !Ref LogRole
WebServerHost:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
  Metadata:
    'AWS::CloudFormation::Init':
      configSets:
        config:
          - 00-ConfigureCWLogs
          - 01-InstallWebServer
          - 02-ConfigureApplication
          - 03-Finalize
      00-ConfigureCWLogs:
        files:
          'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
            content: !Sub |
              {
                "EngineConfiguration": {
                  "Components": [
                    {
                      "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                      "Id": "ApplicationEventLog",
                      "Parameters": {
                        "Levels": "7",
                        "LogName": "Application"
                      }
                    },
                    {
                      "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                      "Id": "SystemEventLog",
                      "Parameters": {
                        "Levels": "7",
                        "LogName": "System"
                      }
                    }
                  ]
                }
              }

```

```

        "Id": "SecurityEventLog",
        "Parameters": {
            "Levels": "7",
            "LogName": "Security"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\\\Program Files\\\\Amazon\\
\\Ec2ConfigService\\\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\\\cfn\\\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\\\inetpub\\\\logs\\\\LogFiles\\
\\W3SVC1",

```

```

        "TimeZoneKind": "UTC",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/ApplicationEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SystemEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {

```

```

        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SecurityEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/EC2ConfigLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/CfnInitLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/IISLogs",
            "Region": "${AWS::Region}",

```

```

        "SecretKey": ""
    }
},
{
    "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
    "Id": "CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "NameSpace": "Windows/Default",
        "Region": "${AWS::Region}",
        "SecretKey": ""
    }
}
],
"Flows": {
    "Flows": [
        "ApplicationEventLog,CloudWatchApplicationEventLog",
        "SystemEventLog,CloudWatchSystemEventLog",
        "SecurityEventLog,CloudWatchSecurityEventLog",
        "EC2ConfigLog,CloudWatchEC2ConfigLog",
        "CfnInitLog,CloudWatchCfnInitLog",
        "IISLogs,CloudWatchIISLogs",
        "MemoryPerformanceCounter,CloudWatch"
    ]
},
"PollInterval": "00:00:05"
},
"IsEnabled": true
}
commands:
  0-enableSSM:
    command: >-
      powershell.exe -Command "Set-Service -Name AmazonSSMAgent
        -StartupType Automatic"
    waitAfterCompletion: '0'
  1-restartSSM:
    command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
    waitAfterCompletion: '30'
01-InstallWebServer:
  commands:
    01_install_webserver:
      command: >-
        powershell.exe -Command "Install-WindowsFeature Web-Server

```

```

        -IncludeAllSubFeature"
        waitAfterCompletion: '0'
    02-ConfigureApplication:
        files:
            'c:\Inetpub\wwwroot\index.htm':
                content: >-
                    <html> <head> <title>Test Application Page</title> </head>
                    <body> <h1>Congratulations !! Your IIS server is
                    configured.</h1> </body> </html>
    03-Finalize:
        commands:
            00_signal_success:
                command: !Sub >-
                    cfn-signal.exe -e 0 --resource WebServerHost --stack
                    ${AWS::StackName} --region ${AWS::Region}
                waitAfterCompletion: '0'
Properties:
    KeyName: !Ref KeyPair
    ImageId: "{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2012-
R2_RTM-English-64Bit-Base}}"
    InstanceType: t2.xlarge
    SecurityGroupIds:
        - !Ref WebServerSecurityGroup
    IamInstanceProfile: !Ref LogRoleInstanceProfile
    UserData: !Base64
        'Fn::Sub': >
            <script>

            wmic product where "description='Amazon SSM Agent' " uninstall

            wmic product where "description='aws-cfn-bootstrap' " uninstall

            start /wait c:\Windows\system32\msiexec /passive /qn /i
            https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi

            powershell.exe -Command "iwr
            https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
            -UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"

            start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet

            cfn-init.exe -v -c config -s ${AWS::StackName} --resource

```

```
WebServerHost --region ${AWS::Region}

</script>
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties:
    RetentionInDays: 7
404MetricFilter:
  Type: AWS::Logs::MetricFilter
  Properties:
    LogGroupName: !Ref LogGroup
    FilterPattern: >-
      [timestamps, serverip, method, uri, query, port, dash, clientip,
       useragent, status_code = 404, ...]
    MetricTransformations:
      - MetricValue: '1'
        MetricNamespace: test/404s
        MetricName: test404Count
404Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
AlarmNotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for the web server
```



```
PublicIP:
  Description: Public IP address of the web server
  Value: !GetAtt
    - WebServerHost
    - PublicIp
CloudWatchLogGroupName:
  Description: The name of the CloudWatch log group
  Value: !Ref LogGroup
```

Consulte também

Para obter mais informações sobre recursos CloudWatch Logs, consulte [AWS::Logs::LogGroup](#) e [AWS::Logs::MetricFilter](#).

Trechos de modelo do Amazon DynamoDB

Tópicos

- [Aplicativo Auto Scaling com uma tabela do Amazon DynamoDB](#)
- [Consulte também](#)

Aplicativo Auto Scaling com uma tabela do Amazon DynamoDB

Este exemplo configura o Application Auto Scaling para um recurso `AWS::DynamoDB::Table`. O modelo define uma política de escalabilidade `TargetTrackingScaling` que aumenta o throughput de `WriteCapacityUnits` para a tabela.

JSON

```
{
  "Resources": {
    "DDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [
          {
            "AttributeName": "ArtistId",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Concert",
            "AttributeType": "S"
          }
        ]
      }
    }
  }
}
```

```

        },
        {
            "AttributeName": "TicketSales",
            "AttributeType": "S"
        }
    ],
    "KeySchema": [
        {
            "AttributeName": "ArtistId",
            "KeyType": "HASH"
        },
        {
            "AttributeName": "Concert",
            "KeyType": "RANGE"
        }
    ],
    "GlobalSecondaryIndexes": [
        {
            "IndexName": "GSI",
            "KeySchema": [
                {
                    "AttributeName": "TicketSales",
                    "KeyType": "HASH"
                }
            ],
            "Projection": {
                "ProjectionType": "KEYS_ONLY"
            },
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            }
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
    }
},
"WriteCapacityScalableTarget": {
    "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties": {
        "MaxCapacity": 15,

```

```

        "MinCapacity": 5,
        "ResourceId": {
            "Fn::Join": [
                "/",
                [
                    "table",
                    {
                        "Ref": "DDBTable"
                    }
                ]
            ],
            "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "ServiceNamespace": "dynamodb"
        }
    },
    "WriteScalingPolicy": {
        "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties": {
            "PolicyName": "WriteAutoScalingPolicy",
            "PolicyType": "TargetTrackingScaling",
            "ScalingTargetId": {
                "Ref": "WriteCapacityScalableTarget"
            },
            "TargetTrackingScalingPolicyConfiguration": {
                "TargetValue": 50,
                "ScaleInCooldown": 60,
                "ScaleOutCooldown": 60,
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "DynamoDBWriteCapacityUtilization"
                }
            }
        }
    }
}

```

YAML

Resources:

```
DDBTable:
  Type: AWS::DynamoDB::Table
  Properties:
    AttributeDefinitions:
      - AttributeName: "ArtistId"
        AttributeType: "S"
      - AttributeName: "Concert"
        AttributeType: "S"
      - AttributeName: "TicketSales"
        AttributeType: "S"
    KeySchema:
      - AttributeName: "ArtistId"
        KeyType: "HASH"
      - AttributeName: "Concert"
        KeyType: "RANGE"
    GlobalSecondaryIndexes:
      - IndexName: "GSI"
        KeySchema:
          - AttributeName: "TicketSales"
            KeyType: "HASH"
        Projection:
          ProjectionType: "KEYS_ONLY"
        ProvisionedThroughput:
          ReadCapacityUnits: 5
          WriteCapacityUnits: 5
    ProvisionedThroughput:
      ReadCapacityUnits: 5
      WriteCapacityUnits: 5
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      ResourceId: !Join
        - /
        - - table
          - !Ref DDBTable
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ServiceNamespace: dynamodb
  WriteScalingPolicy:
```

```
Type: AWS::ApplicationAutoScaling::ScalingPolicy
Properties:
  PolicyName: WriteAutoScalingPolicy
  PolicyType: TargetTrackingScaling
  ScalingTargetId: !Ref WriteCapacityScalableTarget
  TargetTrackingScalingPolicyConfiguration:
    TargetValue: 50.0
    ScaleInCooldown: 60
    ScaleOutCooldown: 60
    PredefinedMetricSpecification:
      PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Consulte também

Consulte mais informações na publicação do blog [How to use CloudFormation to configure auto scaling for DynamoDB tables and indexes](#) no blog de banco de dados da AWS.

Para obter mais informações sobre recursos DynamoDB, consulte [AWS::DynamoDB::Table](#).

Snippets de modelo do CloudFormation do Amazon EC2

O Amazon EC2 fornece capacidade de computação escalável na Nuvem AWS. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento. Esses servidores virtuais, conhecidos como instâncias, podem executar uma variedade de sistemas operacionais e aplicações e podem ser personalizados para atender às suas necessidades específicas. O Amazon EC2 permite que você aumente ou reduza a escala verticalmente para lidar com as mudanças nos requisitos ou spikes em uso.

É possível definir e provisionar instâncias do Amazon EC2 como parte de sua infraestrutura usando modelos do CloudFormation. Os modelos facilitam o gerenciamento e a automação da implantação dos recursos do Amazon EC2 de forma consistente e repetível.

Os snippets de modelos de exemplo a seguir descrevem os recursos ou componentes do CloudFormation para o Amazon EC2. Esses snippets foram projetados para serem integrados a um modelo e não devem ser executados de forma independente.

Categorias de snippets

- [Configurar instâncias do Amazon EC2 com o CloudFormation](#)
- [Criar modelos de execução com o CloudFormation](#)
- [Gerenciar grupos de segurança com o CloudFormation](#)

- [Alocar e associar endereços IP elásticos com o CloudFormation](#)
- [Configurar os recursos da Amazon VPC com o CloudFormation](#)

Configurar instâncias do Amazon EC2 com o CloudFormation

Os snippets a seguir demonstram como configurar instâncias do Amazon EC2 usando o CloudFormation.

Categorias de snippets

- [Configurações gerais do Amazon EC2](#)
- [Especificar os mapeamentos de dispositivos de blocos para uma instância](#)

Configurações gerais do Amazon EC2

Os snippets a seguir demonstram as configurações gerais para as instâncias do Amazon EC2 usando o CloudFormation.

Trechos de exemplo

- [Criar uma instância do Amazon EC2 em uma zona de disponibilidade especificada](#)
- [Configurar uma instância do Amazon EC2 marcada com um volume do EBS e dados do usuário](#)
- [Definir o nome da tabela do DynamoDB nos dados de usuário para a execução da instância do Amazon EC2](#)
- [Criar um volume do Amazon EBS com uma DeletionPolicy](#)

Criar uma instância do Amazon EC2 em uma zona de disponibilidade especificada

O snippet a seguir cria uma instância do Amazon EC2 na zona de disponibilidade especificada usando um recurso [AWS::EC2::Instance](#). O código da zona de disponibilidade é o seu código de região seguido por um identificador de letra. Você pode iniciar uma instância em uma zona de disponibilidade única.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "AvailabilityZone": "aa-example-1a",
```

```
    "ImageId": "ami-1234567890abcdef0"  
  }  
}
```

YAML

```
Ec2Instance:  
  Type: AWS::EC2::Instance  
  Properties:  
    AvailabilityZone: aa-example-1a  
    ImageId: ami-1234567890abcdef0
```

Configurar uma instância do Amazon EC2 marcada com um volume do EBS e dados do usuário

O snippet a seguir cria uma instância do Amazon EC2 com uma tag, um volume do EBS e dados do usuário. Ele usa um recurso [AWS::EC2::Instance](#). No mesmo modelo, você deve definir um recurso [AWS::EC2::SecurityGroup](#), [AWS::SNS::Topic](#) e [AWS::EC2::Volume](#). Você deve definir o KeyName na seção `Parameters` do modelo.

As tags ajudam a categorizar os recursos da AWS com base em suas preferências, como finalidade, proprietário ou ambiente. Os dados do usuário permitem o provisionamento de scripts ou dados personalizados para uma instância durante a execução. Esses dados facilitam a automação de tarefas, a configuração de software, a instalação de pacotes e outras ações em uma instância durante a inicialização.

Para obter mais informações sobre marcação de recursos com tags, consulte [Marcar recursos do Amazon EC2 com tags](#) no Guia do usuário do Amazon EC2.

Para obter informações sobre dados de usuário, consulte [Usar metadados da instância para gerenciar sua instância do EC2](#) no Guia do usuário do Amazon EC2.

JSON

```
"Ec2Instance": {  
  "Type": "AWS::EC2::Instance",  
  "Properties": {  
    "KeyName": { "Ref": "KeyName" },  
    "SecurityGroups": [ { "Ref": "Ec2SecurityGroup" } ],  
    "UserData": {  
      "Fn::Base64": {  
        "Fn::Join": [ ":", [  
          "PORT=80",
```

```

        "TOPIC=",
        { "Ref": "MySNSTopic" }
      ]
    ]
  },
  "InstanceType": "aa.size",
  "AvailabilityZone": "aa-example-1a",
  "ImageId": "ami-1234567890abcdef0",
  "Volumes": [
    {
      "VolumeId": { "Ref": "MyVolumeResource" },
      "Device": "/dev/sdk"
    }
  ],
  "Tags": [ { "Key": "Name", "Value": "MyTag" } ]
}

```

YAML

```

Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    KeyName: !Ref KeyName
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    UserData:
      Fn::Base64:
        Fn::Join:
          - ":"
          - - "PORT=80"
            - "TOPIC="
            - !Ref MySNSTopic
    InstanceType: aa.size
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
    Volumes:
      - VolumeId: !Ref MyVolumeResource
        Device: "/dev/sdk"
    Tags:
      - Key: Name
        Value: MyTag

```


Definir o nome da tabela do DynamoDB nos dados de usuário para a execução da instância do Amazon EC2

O snippet a seguir cria uma instância do Amazon EC2 e define um nome de tabela do DynamoDB nos dados de usuário para transmitir para a instância na execução. Ele usa um recurso [AWS::EC2::Instance](#). Você pode definir parâmetros ou valores dinâmicos nos dados de usuário para transmitir uma instância do EC2 na execução.

Para obter mais informações sobre dados de usuário, consulte [Usar metadados da instância para gerenciar sua instância do EC2](#) no Guia do usuário do Amazon EC2.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [
          "",
          [
            "TableName=",
            {
              "Ref": "DynamoDBTableName"
            }
          ]
        ]
      },
      "AvailabilityZone": "aa-example-1a",
      "ImageId": "ami-1234567890abcdef0"
    }
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    UserData:
      Fn::Base64:
        Fn::Join:
          - ''
```

```
- - 'TableName='  
  - Ref: DynamoDBTableName  
AvailabilityZone: aa-example-1a  
ImageId: ami-1234567890abcdef0
```

Criar um volume do Amazon EBS com uma **DeletionPolicy**

Os snippets a seguir criam um volume do Amazon EBS usando um recurso [AWS::EC2::Volume](#) do Amazon EC2. Você pode usar as propriedades `Size` ou `SnapshotID` para definir o volume, mas não ambas. Um atributo `DeletionPolicy` é definido para criar um snapshot do volume quando a pilha é excluída.

Para obter mais informações sobre o atributo `DeletionPolicy`, consulte [Atributo DeletionPolicy](#).

Para obter mais informações sobre criar volumes do Amazon EBS, consulte [Criar um volume do Amazon EBS](#).

JSON

Esse snippet cria um volume do Amazon EBS com um tamanho especificado. O tamanho está definido como 10, mas você pode ajustá-lo conforme necessário. O recurso [AWS::EC2::Volume](#) permite que você especifique o tamanho ou o ID do snapshot, mas não ambos.

```
"MyEBSVolume": {  
  "Type": "AWS::EC2::Volume",  
  "Properties": {  
    "Size": "10",  
    "AvailabilityZone": {  
      "Ref": "AvailabilityZone"  
    }  
  },  
  "DeletionPolicy": "Snapshot"  
}
```

Esse snippet cria um volume do Amazon EBS usando um ID de snapshot fornecido. O recurso [AWS::EC2::Volume](#) permite que você especifique o tamanho ou o ID do snapshot, mas não ambos.

```
"MyEBSVolume": {  
  "Type": "AWS::EC2::Volume",  
  "Properties": {  
    "SnapshotId" : "snap-1234567890abcdef0",  
    "AvailabilityZone": {
```

```
        "Ref": "AvailabilityZone"
      }
    },
    "DeletionPolicy": "Snapshot"
  }
}
```

YAML

Esse snippet cria um volume do Amazon EBS com um tamanho especificado. O tamanho está definido como 10, mas você pode ajustá-lo conforme necessário. O recurso [AWS::EC2::Volume](#) permite que você especifique o tamanho ou o ID do snapshot, mas não ambos.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
    DeletionPolicy: Snapshot
```

Esse snippet cria um volume do Amazon EBS usando um ID de snapshot fornecido. O recurso [AWS::EC2::Volume](#) permite que você especifique o tamanho ou o ID do snapshot, mas não ambos.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    SnapshotId: snap-1234567890abcdef0
    AvailabilityZone:
      Ref: AvailabilityZone
    DeletionPolicy: Snapshot
```

Especificar os mapeamentos de dispositivos de blocos para uma instância

Um mapeamento de dispositivos de blocos define os dispositivos de blocos, que inclui volumes de armazenamento de instâncias e volumes do EBS, para anexar a uma instância. Você pode especificar um mapeamento de dispositivos de blocos ao criar uma AMI para que o mapeamento seja usado por todas as instâncias executadas da AMI. Como alternativa, você pode especificar um mapeamento de dispositivos de blocos ao executar uma instância para que o mapeamento substitua o especificado na AMI do qual a instância foi executada.

Você pode usar os snippets de modelo a seguir para especificar os mapeamentos de dispositivos de blocos para seu EBS ou volumes de armazenamento de instâncias usando a propriedade `BlockDeviceMappings` de um recurso [AWS::EC2::Instance](#).

Para obter mais informações sobre mapeamento de dispositivos de blocos, consulte [Mapeamento de dispositivos de blocos para volumes em instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Cenários

- [Especificar os mapeamentos de dispositivos de blocos para dois volumes do EBS](#)
- [Especificar os mapeamentos de dispositivos de blocos para um volume de armazenamento de instância](#)

Especificar os mapeamentos de dispositivos de blocos para dois volumes do EBS

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "KeyName": { "Ref": "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/sda1",
        "Ebs": { "VolumeSize": "50" }
      },
      {
        "DeviceName": "/dev/sdm",
        "Ebs": { "VolumeSize": "100" }
      }
    ]
  }
}
```

YAML

```

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 50
      -
        DeviceName: /dev/sdm
        Ebs:
          VolumeSize: 100

```

Especificar os mapeamentos de dispositivos de blocos para um volume de armazenamento de instância

JSON

```

"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdc",
        "VirtualName" : "ephemeral0"
      }
    ]
  }
}

```

YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        VirtualName: ephemeral0
```

Criar modelos de execução com o CloudFormation

Esta seção fornece um exemplo de como criar um modelo de execução do Amazon EC2 usando o CloudFormation. Os modelos de execução permitem que você crie modelos para configurar e provisionar instâncias do Amazon EC2 na AWS. Com os modelos de execução, é possível armazenar parâmetros de inicialização de modo que você não precise especificá-los toda vez que iniciar uma instância. Para obter mais exemplos, consulte a seção [Exemplos](#) no recurso `AWS::EC2::LaunchTemplate`.

Para obter mais informações sobre como criar um modelo de execução, consulte [Armazenar os parâmetros de execução de instâncias nos modelos de execução do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Para obter mais informações sobre como criar modelos de execução para usar com grupos do Auto Scaling, consulte [Modelos de execução do Auto Scaling](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Categorias de snippets

- [Crie um modelo de execução que especifique grupos de segurança, tags, dados do usuário e um perfil do IAM](#)

Crie um modelo de execução que especifique grupos de segurança, tags, dados do usuário e um perfil do IAM

Este snippet mostra um recurso [AWS::EC2::LaunchTemplate](#) que contém as informações de configuração para iniciar uma instância. Você especifica os valores para as propriedades ImageId, InstanceType, SecurityGroups, UserData e TagSpecifications. A propriedade SecurityGroups especifica um grupo de segurança do EC2 existente, bem como um novo grupo de segurança. A função Ref recebe o ID do myNewEC2SecurityGroup do recurso [AWS::EC2::SecurityGroup](#) que é declarado em outro lugar no modelo da pilha.

O modelo de execução contém uma seção para dados de usuário personalizados. É possível transmitir tarefas de configuração e scripts que são executados quando uma instância é iniciada nessa seção. Neste exemplo, os dados do usuário instalam o AWS Systems Manager Agent e iniciam o agente.

O modelo de execução também inclui um perfil do IAM que permite que aplicações em execução em instâncias executem ações em seu nome. Este exemplo mostra um recurso [AWS::IAM::Role](#) para o modelo de inicialização, que usa a propriedade IamInstanceProfile para especificar o perfil do IAM. A função Ref recebe o nome do myInstanceProfile do recurso [AWS::IAM::InstanceProfile](#). Para configurar as permissões do perfil do IAM, especifique um valor para a propriedade ManagedPolicyArns.

JSON

```
{
  "Resources":{
    "myLaunchTemplate":{
      "Type":"AWS::EC2::LaunchTemplate",
      "Properties":{
        "LaunchTemplateName":{ "Fn::Sub": "${AWS::StackName}-launch-template" },
        "LaunchTemplateData":{
          "ImageId":"ami-02354e95b3example",
          "InstanceType":"t3.micro",
          "IamInstanceProfile":{
            "Name":{
              "Ref":"myInstanceProfile"
            }
          },
        },
        "SecurityGroupIds":[
          {
            "Ref":"myNewEC2SecurityGroup"
          }
        ]
      }
    }
  }
}
```

```

    },
    "sg-083cd3bfb8example"
  ],
  "UserData":{
    "Fn::Base64":{
      "Fn::Join": [
        "", [
          "#!/bin/bash\n",
          "cd /tmp\n",
          "yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm\n",
          "systemctl enable amazon-ssm-agent\n",
          "systemctl start amazon-ssm-agent\n"
        ]
      ]
    }
  },
  "TagSpecifications":[
    {
      "ResourceType":"instance",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    },
    {
      "ResourceType":"volume",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    }
  ]
},
"myInstanceRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "RoleName":"InstanceRole",

```



```

    "AssumeRolePolicyDocument":{
      "Version": "2012-10-17",
      "Statement":[
        {
          "Effect":"Allow",
          "Principal":{"
            "Service":[
              "ec2.amazonaws.com"
            ]
          },
          "Action":[
            "sts:AssumeRole"
          ]
        }
      ]
    },
    "ManagedPolicyArns":[
      "arn:aws:iam::aws:policy/myCustomerManagedPolicy"
    ]
  }
},
"myInstanceProfile":{
  "Type":"AWS::IAM::InstanceProfile",
  "Properties":{
    "Path":"/",
    "Roles":[
      {
        "Ref":"myInstanceRole"
      }
    ]
  }
}
}
}
}
}

```

YAML

```

---
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template

```

```

LaunchTemplateData:
  ImageId: ami-02354e95b3example
  InstanceType: t3.micro
  IamInstanceProfile:
    Name: !Ref myInstanceProfile
  SecurityGroupIds:
    - !Ref myNewEC2SecurityGroup
    - sg-083cd3bfb8example
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash
      cd /tmp
      yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
      systemctl enable amazon-ssm-agent
      systemctl start amazon-ssm-agent
  TagSpecifications:
    - ResourceType: instance
      Tags:
        - Key: environment
          Value: development
    - ResourceType: volume
      Tags:
        - Key: environment
          Value: development
myInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: InstanceRole
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: 'Allow'
          Principal:
            Service:
              - 'ec2.amazonaws.com'
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/myCustomerManagedPolicy'
myInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: '/'

```

```
Roles:  
- !Ref myInstanceRole
```

Gerenciar grupos de segurança com o CloudFormation

Os snippets a seguir demonstram como usar o CloudFormation para gerenciar grupos de segurança e instâncias do Amazon EC2 para controlar o acesso aos seus recursos da AWS.

Categorias de snippets

- [Associar uma instância do Amazon EC2 com um grupo de segurança](#)
- [Criar grupo de segurança com regras de entrada](#)
- [Criar um Elastic Load Balancer com uma regra de entrada de grupos de segurança](#)

Associar uma instância do Amazon EC2 com um grupo de segurança

Os snippets de exemplo a seguir demonstram como associar uma instância do Amazon EC2 a um grupo de segurança padrão da Amazon VPC usando o CloudFormation.

Trechos de exemplo

- [Associar uma instância do Amazon EC2 a um grupo de segurança padrão da VPC](#)
- [Criar uma instância do Amazon EC2 com um grupo de segurança e um volume anexado](#)

Associar uma instância do Amazon EC2 a um grupo de segurança padrão da VPC

O snippet a seguir cria uma Amazon VPC, uma sub-rede dentro da VPC e uma instância do Amazon EC2. A VPC é criada usando um recurso [AWS::EC2::VPC](#). O intervalo de endereços IP da VPC é definido no modelo maior e é referenciado pelo parâmetro MyVPCCIDRRange.

Uma sub-rede é criada dentro da VPC usando um recurso [AWS::EC2::Subnet](#). A sub-rede está associada à VPC, que é referenciada como MyVPC.

Uma instância do EC2 é executada dentro da VPC e na sub-rede usando um recurso [AWS::EC2::Instance](#). Esse recurso especifica a imagem de máquina da Amazon (AMI) a ser usada para executar a instância, a sub-rede em que a instância será executada e o grupo de segurança a ser associado à instância. O ImageId usa um parâmetro do Systems Manager para recuperar dinamicamente a AMI mais recente do Amazon Linux 2.

O ID do grupo de segurança é obtido usando a função `Fn::GetAtt`, que recupera o grupo de segurança padrão do recurso `MyVPC`.

A instância é colocada no recurso `MySubnet` definido no snippet.

Quando você cria uma VPC usando o CloudFormation, a AWS cria automaticamente recursos padrão dentro da VPC, incluindo um grupo de segurança padrão. No entanto, ao definir uma VPC em um modelo do CloudFormation, você talvez não tenha acesso aos IDs desses recursos padrão ao criar o modelo. Para acessar e usar os recursos padrão especificados no modelo, você pode usar funções intrínsecas, como `Fn::GetAtt`. Essa função permite que você trabalhe com os recursos padrão que são criados automaticamente pelo CloudFormation.

JSON

```
"MyVPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "EnableDnsSupport": false,
    "EnableDnsHostnames": false,
    "InstanceTenancy": "default"
  }
},
"MySubnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "VpcId": {
      "Ref": "MyVPC"
    }
  }
},
"MyInstance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "SecurityGroupIds": [
      {
```

```

        "Fn::GetAtt": [
            "MyVPC",
            "DefaultSecurityGroup"
        ]
    },
    "SubnetId": {
        "Ref": "MySubnet"
    }
}
}

```

YAML

```

MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet

```

Criar uma instância do Amazon EC2 com um grupo de segurança e um volume anexado

O snippet a seguir cria uma instância do Amazon EC2 usando um recurso [AWS::EC2::Instance](#), que é executado de uma AMI designada. A instância está associada a um grupo de segurança que permite a entrada de tráfego SSH na porta 22 de um endereço IP especificado, usando um recurso [AWS::EC2::SecurityGroup](#). Ele cria um volume do Amazon EBS de 100 GB usando um recurso [AWS::EC2::Volume](#). O volume é criado na mesma zona de disponibilidade da instância, conforme especificado pela função `GetAtt`, e é montado na instância no dispositivo `/dev/sdh`.

Para obter mais informações sobre criar volumes do Amazon EBS, consulte [Criar um volume do Amazon EBS](#).

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "SecurityGroups": [
      {
        "Ref": "InstanceSecurityGroup"
      }
    ],
    "ImageId": "ami-1234567890abcdef0"
  }
},
"InstanceSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Enable SSH access via port 22",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
      }
    ]
  }
},
"NewVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "Size": "100",
```

```

        "AvailabilityZone": {
            "Fn::GetAtt": [
                "Ec2Instance",
                "AvailabilityZone"
            ]
        }
    },
    "MountPoint": {
        "Type": "AWS::EC2::VolumeAttachment",
        "Properties": {
            "InstanceId": {
                "Ref": "Ec2Instance"
            },
            "VolumeId": {
                "Ref": "NewVolume"
            },
            "Device": "/dev/sdh"
        }
    }
}

```

YAML

```

Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    SecurityGroups:
      - !Ref InstanceSecurityGroup
    ImageId: ami-1234567890abcdef0
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24
NewVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt [Ec2Instance, AvailabilityZone]

```

```
MountPoint:
  Type: AWS::EC2::VolumeAttachment
  Properties:
    InstanceId: !Ref Ec2Instance
    VolumeId: !Ref NewVolume
    Device: /dev/sdh
```

Criar grupo de segurança com regras de entrada

Os snippets de exemplo a seguir demonstram como configurar grupos de segurança com regras de entrada específicas usando o CloudFormation.

Snippets

- [Criar um grupo de segurança com regras de entrada para acesso SSH e HTTP](#)
- [Criar um grupo de segurança com regras de entrada para acesso HTTP e SSH de intervalos CIDR especificados](#)
- [Criar grupos de segurança de referência cruzada com regras de entrada](#)

Criar um grupo de segurança com regras de entrada para acesso SSH e HTTP

Esse snippet descreve duas regras de entrada do grupo de segurança usando um recurso [AWS::EC2::SecurityGroup](#). A primeira regra de entrada permite o acesso SSH (porta 22) de um grupo de segurança existente chamado MyAdminSecurityGroup, que pertence à conta da AWS com o número de conta 1111-2222-3333. A segunda regra de entrada permite o acesso HTTP (porta 80) de outro grupo de segurança chamado MySecurityGroupCreatedInCFN, que é criado no mesmo modelo. A função Ref é usada para fazer referência ao nome lógico do grupo de segurança criado no mesmo modelo.

Na primeira regra de entrada, você deve adicionar um valor para as propriedades SourceSecurityGroupOwnerId e SourceSecurityGroupName. Na segunda regra de entrada, MySecurityGroupCreatedInCFNTemplate faz referência a outro grupo de segurança criado no mesmo modelo. Verifique se o nome lógico MySecurityGroupCreatedInCFNTemplate corresponde ao nome lógico real do recurso do grupo de segurança que você especifica no modelo maior.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança do Amazon EC2 para suas instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

JSON

```
"SecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified source security group",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "SourceSecurityGroupName": "MyAdminSecurityGroup",
        "SourceSecurityGroupOwnerId": "1111-2222-3333"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "SourceSecurityGroupName": {
          "Ref": "MySecurityGroupCreatedInCFNTemplate"
        }
      }
    ]
  }
}
```

YAML

```
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified source security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        SourceSecurityGroupName: MyAdminSecurityGroup
        SourceSecurityGroupOwnerId: '1111-2222-3333'
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        SourceSecurityGroupName:
          Ref: MySecurityGroupCreatedInCFNTemplate
```

Criar um grupo de segurança com regras de entrada para acesso HTTP e SSH de intervalos CIDR especificados

O snippet a seguir cria um grupo de segurança para uma instância do Amazon EC2 com duas regras de entrada. As regras de entrada permitem o tráfego TCP de entrada nas portas especificadas dos intervalos CIDR designados. Um recurso [AWS::EC2::SecurityGroup](#) é usado para especificar as regras. Você deve especificar um protocolo para cada regra. Para TCP, você deve especificar uma porta ou um intervalo de portas. Se você não especificar um grupo de segurança de origem ou um intervalo CIDR, a pilha será executada com êxito, mas a regra não será aplicada ao grupo de segurança.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança do Amazon EC2 para suas instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

JSON

```
"ServerSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified CIDR ranges",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "CidrIp": "192.0.2.0/24"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
      }
    ]
  }
}
```

YAML

```
ServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
```

```
GroupDescription: Allow connections from specified CIDR ranges
SecurityGroupIngress:
  - IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 192.0.2.0/24
  - IpProtocol: tcp
    FromPort: 22
    ToPort: 22
    CidrIp: 192.0.2.0/24
```

Criar grupos de segurança de referência cruzada com regras de entrada

O snippet a seguir usa o recurso [AWS::EC2::SecurityGroup](#) para criar dois grupos de segurança do Amazon EC2: SGroup1 e SGroup2. As regras de entrada que permitem a comunicação entre os dois grupos de segurança são criadas usando o recurso [AWS::EC2::SecurityGroupIngress](#). SGroup1Ingress estabelece uma regra de entrada para SGroup1 que permite o tráfego TCP de entrada na porta 80 do grupo de segurança de origem, SGroup2. SGroup2Ingress estabelece uma regra de entrada para SGroup2 que permite o tráfego TCP de entrada na porta 80 do grupo de segurança de origem, SGroup1.

JSON

```
"SGroup1": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "EC2 instance access"
  }
},
"SGroup2": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "EC2 instance access"
  }
},
"SGroup1Ingress": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties": {
    "GroupName": {
      "Ref": "SGroup1"
    },
    "IpProtocol": "tcp",
```

```

        "ToPort": "80",
        "FromPort": "80",
        "SourceSecurityGroupName": {
            "Ref": "SGroup2"
        }
    },
    "SGroup2Ingress": {
        "Type": "AWS::EC2::SecurityGroupIngress",
        "Properties": {
            "GroupName": {
                "Ref": "SGroup2"
            },
            "IpProtocol": "tcp",
            "ToPort": "80",
            "FromPort": "80",
            "SourceSecurityGroupName": {
                "Ref": "SGroup1"
            }
        }
    }
}

```

YAML

```

SGroup1:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup2:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup1Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupName: !Ref SGroup1
    IpProtocol: tcp
    ToPort: 80
    FromPort: 80
    SourceSecurityGroupName: !Ref SGroup2
SGroup2Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:

```

```
GroupName: !Ref SGroup2
IpProtocol: tcp
ToPort: 80
FromPort: 80
SourceSecurityGroupName: !Ref SGroup1
```

Criar um Elastic Load Balancer com uma regra de entrada de grupos de segurança

O modelo a seguir cria um recurso [AWS::ElasticLoadBalancing::LoadBalancer](#) na zona de disponibilidade especificada. O recurso [AWS::ElasticLoadBalancing::LoadBalancer](#) está configurado para receber tráfego HTTP na porta 80 e direcionar solicitações para instâncias também na porta 80. O Elastic Load Balancer é responsável pelo balanceamento de carga do tráfego HTTP de entrada entre as instâncias.

Além disso, esse modelo gera um recurso [AWS::EC2::SecurityGroup](#) associado ao balanceador de carga. Esse grupo de segurança é criado com uma única regra de entrada, descrita como `ELB ingress group`, que permite o tráfego TCP de entrada na porta 80. A origem dessa regra de entrada é definida usando a função `Fn::GetAtt` para recuperar atributos do recurso do balanceador de carga. `SourceSecurityGroupOwnerId` usa `Fn::GetAtt` para obter o `OwnerAlias` do grupo de segurança de origem do balanceador de carga. `SourceSecurityGroupName` usa `Fn::GetAtt` para obter o `GroupName` do grupo de segurança de origem do ELB.

Essa configuração garante uma comunicação segura entre o ELB e as instâncias.

Para mais informações sobre balanceamento de carga consulte o [Manual do usuário do Elastic Load Balancing](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyELB": {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {
        "AvailabilityZones": [
          "aa-example-1a"
        ],
        "Listeners": [
          {
```

```
        "LoadBalancerPort": "80",
        "InstancePort": "80",
        "Protocol": "HTTP"
    }
]
},
"MyELBIngressGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "ELB ingress group",
        "SecurityGroupIngress": [
            {
                "IpProtocol": "tcp",
                "FromPort": 80,
                "ToPort": 80,
                "SourceSecurityGroupId": {
                    "Fn::GetAtt": [
                        "MyELB",
                        "SourceSecurityGroup.OwnerAlias"
                    ]
                },
                "SourceSecurityGroupName": {
                    "Fn::GetAtt": [
                        "MyELB",
                        "SourceSecurityGroup.GroupName"
                    ]
                }
            }
        ]
    }
}
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyELB:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
      AvailabilityZones:
```

```

- aa-example-1a
Listeners:
- LoadBalancerPort: '80'
  InstancePort: '80'
  Protocol: HTTP
MyELBIngressGroup:
Type: AWS::EC2::SecurityGroup
Properties:
  GroupDescription: ELB ingress group
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: '80'
      ToPort: '80'
      SourceSecurityGroupOwnerId:
        Fn::GetAtt:
          - MyELB
          - SourceSecurityGroup.OwnerAlias
  SourceSecurityGroupName:
    Fn::GetAtt:
      - MyELB
      - SourceSecurityGroup.GroupName

```

Alocar e associar endereços IP elásticos com o CloudFormation

Os snippets de modelos a seguir são exemplos relacionados a endereços IP elásticos (EIPs) no Amazon EC2. Esses exemplos abrangem alocação, associação e gerenciamento de EIPs para suas instâncias.

Trechos de exemplo

- [Alocar um endereço IP elástico e associá-lo a uma instância do Amazon EC2](#)
- [Associar um endereço IP elástico a uma instância do Amazon EC2 especificando o endereço IP](#)
- [Associar um endereço IP elástico a uma instância do Amazon EC2 especificando o ID de alocação do endereço IP](#)

Alocar um endereço IP elástico e associá-lo a uma instância do Amazon EC2

O snippet a seguir aloca um endereço IP elástico (EIP) do Amazon EC2 e o associa a uma instância do Amazon EC2 usando um recurso [AWS::EC2::EIP](#). Você pode alocar um endereço EIP de um grupo de endereços de propriedade da AWS ou de um grupo de endereços criados em um intervalo de endereços IPv4 públicos que você trouxe para a AWS para uso com seus recursos da AWS,

usando a opção [traga seus próprios endereços IP \(BYOIP\)](#). Neste exemplo, o EIP é alocado de um grupo de endereços de propriedade da AWS.

Para obter mais informações sobre endereços IP elásticos, consulte [Endereços IP elásticos](#) no Guia do usuário do Amazon EC2.

JSON

```
"ElasticIP": {
  "Type": "AWS::EC2::EIP",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    }
  }
}
```

YAML

```
ElasticIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref EC2Instance
```

Associar um endereço IP elástico a uma instância do Amazon EC2 especificando o endereço IP

O snippet a seguir associa um endereço IP elástico existente do Amazon EC2 a uma instância do EC2 usando um recurso [AWS::EC2::EIPAssociation](#). Você deve primeiro alocar um endereço IP elástico para usar em sua conta. Um endereço IP elástico pode ser associado a uma instância única.

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "EIP": "192.0.2.0"
  }
}
```


YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    EIP: 192.0.2.0
```

Associar um endereço IP elástico a uma instância do Amazon EC2 especificando o ID de alocação do endereço IP

O snippet a seguir associa um endereço IP elástico existente a uma instância do Amazon EC2 especificando o ID de alocação usando um recurso [AWS::EC2::EIPAssociation](#). Um ID de alocação é atribuído a um endereço IP elástico na alocação deste.

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "AllocationId": "eipalloc-1234567890abcdef0"
  }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    AllocationId: eipalloc-1234567890abcdef0
```

Configurar os recursos da Amazon VPC com o CloudFormation

Esta seção fornece exemplos de como configurar os recursos da Amazon VPC usando o CloudFormation. As VPCs permitem que você crie uma rede virtual na AWS, e esses snippets mostram como configurar aspectos das VPCs para atender aos seus requisitos de rede.

Trechos de exemplo

- [Habilitar o acesso à Internet IPv6 somente de saída em uma VPC](#)
- [Trechos do modelo da interface de rede elástica \(ENI\)](#)

Habilitar o acesso à Internet IPv6 somente de saída em uma VPC

Um gateway da Internet somente de saída permite que instâncias dentro de uma VPC acessem a Internet e impeçam que recursos na Internet se comuniquem com as instâncias. O snippet a seguir permite o acesso à Internet IPv6 somente de saída de dentro de uma VPC. Ele cria uma VPC com um intervalo de endereços IPv4 de 10.0.0/16 usando um recurso [AWS::EC2::VPC](#). Uma tabela de rotas é associada a esse recurso da VPC usando um recurso [AWS::EC2::RouteTable](#). A tabela de rotas gerencia rotas para instâncias dentro da VPC. Um [AWS::EC2::EgressOnlyInternetGateway](#) é usado para criar um gateway da Internet somente de saída para permitir a comunicação IPv6 para tráfego de saída de instâncias dentro da VPC, evitando o tráfego de entrada. Um recurso [AWS::EC2::Route](#) é especificado para criar uma rota IPv6 na tabela de rotas que direciona todo o tráfego IPv6 de saída (: : /0) para o gateway da Internet somente de saída.

Para obter mais informações sobre gateways da Internet somente de saída, consulte [Habilitar o tráfego IPv6 de saída usando gateways da Internet somente de saída](#) no Guia do usuário da Amazon VPC.

JSON

```
"DefaultIpv6Route": {
  "Type": "AWS::EC2::Route",
  "Properties": {
    "DestinationIpv6CidrBlock": ":::/0",
    "EgressOnlyInternetGatewayId": {
      "Ref": "EgressOnlyInternetGateway"
    },
    "RouteTableId": {
      "Ref": "RouteTable"
    }
  }
},
"EgressOnlyInternetGateway": {
  "Type": "AWS::EC2::EgressOnlyInternetGateway",
  "Properties": {
    "VpcId": {
      "Ref": "VPC"
    }
  }
}
```

```

    }
  }
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"VPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": "10.0.0.0/16"
  }
}

```

YAML

```

DefaultIpv6Route:
  Type: AWS::EC2::Route
  Properties:
    DestinationIpv6CidrBlock: "::/0"
    EgressOnlyInternetGatewayId:
      Ref: "EgressOnlyInternetGateway"
    RouteTableId:
      Ref: "RouteTable"
EgressOnlyInternetGateway:
  Type: AWS::EC2::EgressOnlyInternetGateway
  Properties:
    VpcId:
      Ref: "VPC"
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: "VPC"
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: "10.0.0.0/16"

```

Trechos do modelo da interface de rede elástica (ENI)

Criar uma instância do Amazon EC2 com interfaces de rede elástica (ENIs) anexadas

O trecho de exemplo a seguir cria uma instância do Amazon EC2 usando um recurso [AWS::EC2::Instance](#) na sub-rede e na Amazon VPC especificada. Ele anexa duas interfaces de rede (ENIs) à instância, associa endereços IP elásticos às instâncias por meio dos ENIs anexados e configura o grupo de segurança para acesso SSH e HTTP. Os dados de usuário são fornecidos à instância como parte da configuração de execução quando a instância é criada. Os dados de usuário incluem um script codificado no formato base64 para garantir que sejam transmitidos para a instância. Quando a instância é executada, o script é executado automaticamente como parte do processo de bootstrapping. Ele instala `ec2-net-utils`, configura as interfaces de rede e inicia o serviço HTTP.

Para determinar a imagem de máquina da Amazon (AMI) apropriada com base na região selecionada, o snippet usa uma função `Fn::FindInMap` que pesquisa valores em um mapeamento `RegionMap`. Esse mapeamento deve ser definido no modelo maior. As duas interfaces de rede são criadas usando recursos [AWS::EC2::NetworkInterface](#). Os endereços IP elásticos são especificados usando os recursos [AWS::EC2::EIP](#) alocados ao domínio `vpc`. Esses endereços IP elásticos são associados às interfaces de rede usando os recursos [AWS::EC2::EIPAssociation](#).

A seção `Outputs` define os valores ou recursos que você deseja acessar depois que a pilha é criada. Nesse snippet, a saída definida é `InstancePublicIp`, que representa o endereço IP público da instância do EC2 criada pela pilha. É possível recuperar essa saída na guia `Saída` no console do CloudFormation ou usando o comando [describe-stacks](#).

Para obter mais informações sobre interfaces de rede elástica, consulte [Interfaces de rede elástica](#).

JSON

```
"Resources": {
  "ControlPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
      "Domain": "vpc"
    }
  },
  "AssociateControlPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
      "AllocationId": {
```

```

        "Fn::GetAtt": [
            "ControlPortAddress",
            "AllocationId"
        ]
    },
    "NetworkInterfaceId": {
        "Ref": "controlXface"
    }
}
},
"WebPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
        "Domain": "vpc"
    }
},
"AssociateWebPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
        "AllocationId": {
            "Fn::GetAtt": [
                "WebPortAddress",
                "AllocationId"
            ]
        },
        "NetworkInterfaceId": {
            "Ref": "webXface"
        }
    }
},
"SSHSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "VpcId": {
            "Ref": "VpcId"
        },
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
            {
                "CidrIp": "0.0.0.0/0",
                "FromPort": 22,
                "IpProtocol": "tcp",
                "ToPort": 22
            }
        ]
    }
}

```

```

    ]
  },
  "WebSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "VpcId": {
        "Ref": "VpcId"
      },
      "GroupDescription": "Enable HTTP access via user-defined port",
      "SecurityGroupIngress": [
        {
          "CidrIp": "0.0.0.0/0",
          "FromPort": 80,
          "IpProtocol": "tcp",
          "ToPort": 80
        }
      ]
    }
  },
  "controlXface": {
    "Type": "AWS::EC2::NetworkInterface",
    "Properties": {
      "SubnetId": {
        "Ref": "SubnetId"
      },
      "Description": "Interface for controlling traffic such as SSH",
      "GroupSet": [
        {
          "Fn::GetAtt": [
            "SSHSecurityGroup",
            "GroupId"
          ]
        }
      ],
      "SourceDestCheck": true,
      "Tags": [
        {
          "Key": "Network",
          "Value": "Control"
        }
      ]
    }
  },
},

```

```
"webXface": {
  "Type": "AWS::EC2::NetworkInterface",
  "Properties": {
    "SubnetId": {
      "Ref": "SubnetId"
    },
    "Description": "Interface for web traffic",
    "GroupSet": [
      {
        "Fn::GetAtt": [
          "WebSecurityGroup",
          "GroupId"
        ]
      }
    ],
    "SourceDestCheck": true,
    "Tags": [
      {
        "Key": "Network",
        "Value": "Web"
      }
    ]
  }
},
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": {
      "Fn::FindInMap": [
        "RegionMap",
        {
          "Ref": "AWS::Region"
        }
      ],
      "AMI"
    },
    "KeyName": {
      "Ref": "KeyName"
    },
    "NetworkInterfaces": [
      {
        "NetworkInterfaceId": {
          "Ref": "controlXface"
        }
      }
    ]
  }
}
```

```

        "DeviceIndex": "0"
      },
      {
        "NetworkInterfaceId": {
          "Ref": "webXface"
        },
        "DeviceIndex": "1"
      }
    ],
    "Tags": [
      {
        "Key": "Role",
        "Value": "Test Instance"
      }
    ],
    "UserData": {
      "Fn::Base64": {
        "Fn::Sub": "#!/bin/bash -xe\nyum install ec2-net-utils -y\nec2ifup
eth1\nservice httpd start\n"
      }
    }
  }
},
"Outputs": {
  "InstancePublicIp": {
    "Description": "Public IP Address of the EC2 Instance",
    "Value": {
      "Fn::GetAtt": [
        "Ec2Instance",
        "PublicIp"
      ]
    }
  }
}
}

```

YAML

```

Resources:
  ControlPortAddress:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc

```



```
AssociateControlPort:
  Type: AWS::EC2::EIPAssociation
  Properties:
    AllocationId:
      Fn::GetAtt:
        - ControlPortAddress
        - AllocationId
    NetworkInterfaceId:
      Ref: controlXface
WebPortAddress:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc
AssociateWebPort:
  Type: AWS::EC2::EIPAssociation
  Properties:
    AllocationId:
      Fn::GetAtt:
        - WebPortAddress
        - AllocationId
    NetworkInterfaceId:
      Ref: webXface
SSHSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 22
        IpProtocol: tcp
        ToPort: 22
WebSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable HTTP access via user-defined port
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 80
        IpProtocol: tcp
        ToPort: 80
```

```
controlXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
    Description: Interface for controlling traffic such as SSH
    GroupSet:
      - Fn::GetAtt:
          - SSHSecurityGroup
          - GroupId
    SourceDestCheck: true
    Tags:
      - Key: Network
        Value: Control
webXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
    Description: Interface for web traffic
    GroupSet:
      - Fn::GetAtt:
          - WebSecurityGroup
          - GroupId
    SourceDestCheck: true
    Tags:
      - Key: Network
        Value: Web
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Fn::FindInMap:
        - RegionMap
        - Ref: AWS::Region
        - AMI
    KeyName:
      Ref: KeyName
    NetworkInterfaces:
      - NetworkInterfaceId:
          Ref: controlXface
        DeviceIndex: "0"
      - NetworkInterfaceId:
          Ref: webXface
```

```
    DeviceIndex: "1"
  Tags:
    - Key: Role
      Value: Test Instance
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash -xe
      yum install ec2-net-utils -y
      ec2ifup eth1
      service httpd start
Outputs:
  InstancePublicIp:
    Description: Public IP Address of the EC2 Instance
    Value:
      Fn::GetAtt:
        - Ec2Instance
        - PublicIp
```

Modelos de exemplo do Amazon Elastic Container Service

O Amazon Elastic Container Service (Amazon ECS) é um serviço de gerenciamento de contêineres que facilita a execução, a interrupção e o gerenciamento de contêineres do Docker em um cluster de instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

Criar um cluster com a AMI otimizada para Amazon ECS para AL2023

Defina um cluster que usa um provedor de capacidade que executa instâncias AL2023 no Amazon EC2.

Important

Para os IDs da AMI mais recente, consulte [AMI otimizada pelo Amazon ECS](#), no Amazon Elastic Container Service Developer Guide.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.",
```

```

"Parameters": {
  "InstanceType": {
    "Type": "String",
    "Description": "EC2 instance type",
    "Default": "t2.medium",
    "AllowedValues": [
      "t1.micro",
      "t2.2xlarge",
      "t2.large",
      "t2.medium",
      "t2.micro",
      "t2.nano",
      "t2.small",
      "t2.xlarge",
      "t3.2xlarge",
      "t3.large",
      "t3.medium",
      "t3.micro",
      "t3.nano",
      "t3.small",
      "t3.xlarge"
    ]
  },
  "DesiredCapacity": {
    "Type": "Number",
    "Default": "0",
    "Description": "Number of EC2 instances to launch in your ECS cluster."
  },
  "MaxSize": {
    "Type": "Number",
    "Default": "100",
    "Description": "Maximum number of EC2 instances that can be launched in your
ECS cluster."
  },
  "ECSAMI": {
    "Description": "The Amazon Machine Image ID used for the cluster",
    "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
    "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/
image_id"
  },
  "VpcId": {
    "Type": "AWS::EC2::VPC::Id",
    "Description": "VPC ID where the ECS cluster is launched",
    "Default": "vpc-1234567890abcdef0"
  }
}

```

```
    },
    "SubnetIds": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "List of subnet IDs where the EC2 instances will be launched",
      "Default": "subnet-021345abcdef67890"
    }
  },
  "Resources": {
    "ECSCluster": {
      "Type": "AWS::ECS::Cluster",
      "Properties": {
        "ClusterSettings": [
          {
            "Name": "containerInsights",
            "Value": "enabled"
          }
        ]
      }
    },
    "ECSAutoScalingGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "DependsOn": [
        "ECSCluster",
        "EC2Role"
      ],
      "Properties": {
        "VPCZoneIdentifier": {
          "Ref": "SubnetIds"
        },
        "LaunchTemplate": {
          "LaunchTemplateId": {
            "Ref": "ContainerInstances"
          },
          "Version": {
            "Fn::GetAtt": [
              "ContainerInstances",
              "LatestVersionNumber"
            ]
          }
        },
        "MinSize": 0,
        "MaxSize": {
          "Ref": "MaxSize"
        }
      }
    },
  },
}
```

```

        "DesiredCapacity": {
            "Ref": "DesiredCapacity"
        },
        "NewInstancesProtectedFromScaleIn": true
    },
    "UpdatePolicy": {
        "AutoScalingReplacingUpdate": {
            "WillReplace": "true"
        }
    }
},
"ContainerInstances": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
        "LaunchTemplateName": "asg-launch-template",
        "LaunchTemplateData": {
            "ImageId": {
                "Ref": "ECSAMI"
            },
            "InstanceType": {
                "Ref": "InstanceType"
            },
            "IamInstanceProfile": {
                "Name": {
                    "Ref": "EC2InstanceProfile"
                }
            },
            "SecurityGroupIds": [
                {
                    "Ref": "ContainerHostSecurityGroup"
                }
            ],
            "UserData": {
                "Fn::Base64": {
                    "Fn::Sub": "#!/bin/bash -xe\n echo ECS_CLUSTER=${ECSCluster}
>> /etc/ecs/ecs.config\n yum install -y aws-cfn-bootstrap\n /opt/aws/bin/cfn-init -
v --stack ${AWS::StackId} --resource ContainerInstances --configsets full_install --
region ${AWS::Region} &\n"
                }
            },
            "MetadataOptions": {
                "HttpEndpoint": "enabled",
                "HttpTokens": "required"
            }
        }
    }
}

```

```

    }
  }
},
"EC2InstanceProfile": {
  "Type": "AWS::IAM::InstanceProfile",
  "Properties": {
    "Path": "/",
    "Roles": [
      {
        "Ref": "EC2Role"
      }
    ]
  }
},
"CapacityProvider": {
  "Type": "AWS::ECS::CapacityProvider",
  "Properties": {
    "AutoScalingGroupProvider": {
      "AutoScalingGroupArn": {
        "Ref": "ECSAutoScalingGroup"
      },
    },
    "ManagedScaling": {
      "InstanceWarmupPeriod": 60,
      "MinimumScalingStepSize": 1,
      "MaximumScalingStepSize": 100,
      "Status": "ENABLED",
      "TargetCapacity": 100
    },
    "ManagedTerminationProtection": "ENABLED"
  }
},
"CapacityProviderAssociation": {
  "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
  "Properties": {
    "CapacityProviders": [
      {
        "Ref": "CapacityProvider"
      }
    ],
    "Cluster": {
      "Ref": "ECSCluster"
    },
    "DefaultCapacityProviderStrategy": [

```

```

        {
            "Base": 0,
            "CapacityProvider": {
                "Ref": "CapacityProvider"
            },
            "Weight": 1
        }
    ]
},
"ContainerHostSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the EC2 hosts that run containers",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"EC2Role": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "ec2.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        },
        "Path": "/",
        "ManagedPolicyArns": [
            "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role",
            "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
        ]
    }
}

```



```

    },
    "ECSTaskExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  "ecs-tasks.amazonaws.com"
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ],
              "Condition": {
                "ArnLike": {
                  "aws:SourceArn": {
                    "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
                  }
                },
                "StringEquals": {
                  "aws:SourceAccount": {
                    "Fn::Sub": "${AWS::AccountId}"
                  }
                }
              }
            }
          ]
        },
        "Path": "/",
        "ManagedPolicyArns": [
          "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
        ]
      }
    },
    "Outputs": {
      "ClusterName": {
        "Description": "The ECS cluster into which to launch resources",
        "Value": "ECSCluster"
      }
    }
  }
}

```

```

    },
    "ECSTaskExecutionRole": {
      "Description": "The role used to start up a task",
      "Value": "ECSTaskExecutionRole"
    },
    "CapacityProvider": {
      "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task",
      "Value": "CapacityProvider"
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet.
  An ECS capacity provider automatically launches more EC2 instances as required
  on the fly when you request ECS to launch services or standalone tasks.
Parameters:
  InstanceType:
    Type: String
    Description: EC2 instance type
    Default: "t2.medium"
    AllowedValues:
      - t1.micro
      - t2.2xlarge
      - t2.large
      - t2.medium
      - t2.micro
      - t2.nano
      - t2.small
      - t2.xlarge
      - t3.2xlarge
      - t3.large
      - t3.medium
      - t3.micro
      - t3.nano
      - t3.small
      - t3.xlarge
  DesiredCapacity:
    Type: Number

```

```
    Default: "0"
    Description: Number of EC2 instances to launch in your ECS cluster.
  MaxSize:
    Type: Number
    Default: "100"
    Description: Maximum number of EC2 instances that can be launched in your ECS
cluster.
  ECSAMI:
    Description: The Amazon Machine Image ID used for the cluster
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VPC ID where the ECS cluster is launched
    Default: vpc-1234567890abcdef0
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
    Description: List of subnet IDs where the EC2 instances will be launched
    Default: "subnet-021345abcdef67890"
Resources:
# This is authorizes ECS to manage resources on your
# account on your behalf. This role is likely already created on your account
# ECSRole:
#   Type: AWS::IAM::ServiceLinkedRole
#   Properties:
#     AWSServiceName: 'ecs.amazonaws.com'

# ECS Resources
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterSettings:
      - Name: containerInsights
        Value: enabled

# Autoscaling group. This launches the actual EC2 instances that will register
# themselves as members of the cluster, and run the docker containers.
ECSAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    # This is to ensure that the ASG gets deleted first before these
    # resources, when it comes to stack teardown.
    - ECSCluster
    - EC2Role
```

```

Properties:
  VPCZoneIdentifier:
    Ref: SubnetIds
  LaunchTemplate:
    LaunchTemplateId: !Ref ContainerInstances
    Version: !GetAtt ContainerInstances.LatestVersionNumber
  MinSize: 0
  MaxSize:
    Ref: MaxSize
  DesiredCapacity:
    Ref: DesiredCapacity
  NewInstancesProtectedFromScaleIn: true
UpdatePolicy:
  AutoScalingReplacingUpdate:
    WillReplace: "true"
# The config for each instance that is added to the cluster
ContainerInstances:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: "asg-launch-template"
    LaunchTemplateData:
      ImageId:
        Ref: ECSAMI
      InstanceType:
        Ref: InstanceType
      IamInstanceProfile:
        Name: !Ref EC2InstanceProfile
      SecurityGroupIds:
        - !Ref ContainerHostSecurityGroup
      # This injected configuration file is how the EC2 instance
      # knows which ECS cluster on your AWS account it should be joining
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
ContainerInstances --configsets full_install --region ${AWS::Region} &
          # Disable IMDSv1, and require IMDSv2
      MetadataOptions:
        HttpEndpoint: enabled
        HttpTokens: required
    EC2InstanceProfile:
      Type: AWS::IAM::InstanceProfile

```

```

Properties:
  Path: /
  Roles:
    - !Ref EC2Role
# Create an ECS capacity provider to attach the ASG to the ECS cluster
# so that it autoscales as we launch more containers
CapacityProvider:
  Type: AWS::ECS::CapacityProvider
  Properties:
    AutoScalingGroupProvider:
      AutoScalingGroupArn: !Ref ECSAutoScalingGroup
    ManagedScaling:
      InstanceWarmupPeriod: 60
      MinimumScalingStepSize: 1
      MaximumScalingStepSize: 100
      Status: ENABLED
      # Percentage of cluster reservation to try to maintain
      TargetCapacity: 100
    ManagedTerminationProtection: ENABLED
# Create a cluster capacity provider association so that the cluster
# will use the capacity provider
CapacityProviderAssociation:
  Type: AWS::ECS::ClusterCapacityProviderAssociations
  Properties:
    CapacityProviders:
      - !Ref CapacityProvider
    Cluster: !Ref ECSCluster
    DefaultCapacityProviderStrategy:
      - Base: 0
        CapacityProvider: !Ref CapacityProvider
        Weight: 1
# A security group for the EC2 hosts that will run the containers.
# This can be used to limit incoming traffic to or outgoing traffic
# from the container's host EC2 instance.
ContainerHostSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Access to the EC2 hosts that run containers
    VpcId:
      Ref: VpcId
# Role for the EC2 hosts. This allows the ECS agent on the EC2 hosts
# to communicate with the ECS control plane, as well as download the docker
# images from ECR to run on your host.
EC2Role:

```

```

Type: AWS::IAM::Role
Properties:
  AssumeRolePolicyDocument:
    Statement:
      - Effect: Allow
        Principal:
          Service:
            - ec2.amazonaws.com
        Action:
          - sts:AssumeRole
    Path: /
  ManagedPolicyArns:
    # See reference: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/
security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerServiceforEC2Role
    - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
    # This managed policy allows us to connect to the instance using SSM
    - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
  # This is a role which is used within Fargate to allow the Fargate agent
  # to download images, and upload logs.
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ecs-tasks.amazonaws.com
          Action:
            - sts:AssumeRole
      Condition:
        ArnLike:
          aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
        StringEquals:
          aws:SourceAccount: !Sub ${AWS::AccountId}
    Path: /
    # This role enables all features of ECS. See reference:
    # https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-iam-
awsmanpol.html#security-iam-awsmanpol-AmazonECSTaskExecutionRolePolicy
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
  Outputs:
    ClusterName:

```

Description: The ECS cluster into which to launch resources

Value: ECSCluster

ECSTaskExecutionRole:

Description: The role used to start up a task

Value: ECSTaskExecutionRole

CapacityProvider:

Description: The cluster capacity provider that the service should use to request capacity when it wants to start up a task

Value: CapacityProvider

Implantar um serviço.

O modelo a seguir define um serviço que usa o provedor de capacidade para solicitar capacidade do AL2023 para execução. Os contêineres serão lançados nas instâncias do AL2023 assim que se tornarem online:

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.",
  "Parameters": {
    "VpcId": {
      "Type": "String",
      "Description": "The VPC that the service is running inside of"
    },
    "PublicSubnetIds": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "List of public subnet ID's to put the load balancer in"
    },
    "PrivateSubnetIds": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "List of private subnet ID's that the AWS VPC tasks are in"
    },
    "ClusterName": {
      "Type": "String",
      "Description": "The name of the ECS cluster into which to launch capacity."
    },
    "ECSTaskExecutionRole": {
      "Type": "String",
```

```
    "Description": "The role used to start up an ECS task"
  },
  "CapacityProvider": {
    "Type": "String",
    "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task"
  },
  "ServiceName": {
    "Type": "String",
    "Default": "web",
    "Description": "A name for the service"
  },
  "ImageUrl": {
    "Type": "String",
    "Default": "public.ecr.aws/docker/library/nginx:latest",
    "Description": "The url of a docker image that contains the application
process that will handle the traffic for this service"
  },
  "ContainerCpu": {
    "Type": "Number",
    "Default": 256,
    "Description": "How much CPU to give the container. 1024 is 1 CPU"
  },
  "ContainerMemory": {
    "Type": "Number",
    "Default": 512,
    "Description": "How much memory in megabytes to give the container"
  },
  "ContainerPort": {
    "Type": "Number",
    "Default": 80,
    "Description": "What port that the application expects traffic on"
  },
  "DesiredCount": {
    "Type": "Number",
    "Default": 2,
    "Description": "How many copies of the service task to run"
  }
},
"Resources": {
  "TaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "Family": {
```



```
        "Ref": "ServiceName"
    },
    "Cpu": {
        "Ref": "ContainerCpu"
    },
    "Memory": {
        "Ref": "ContainerMemory"
    },
    "NetworkMode": "awsvpc",
    "RequiresCompatibilities": [
        "EC2"
    ],
    "ExecutionRoleArn": {
        "Ref": "ECSTaskExecutionRole"
    },
    "ContainerDefinitions": [
        {
            "Name": {
                "Ref": "ServiceName"
            },
            "Cpu": {
                "Ref": "ContainerCpu"
            },
            "Memory": {
                "Ref": "ContainerMemory"
            },
            "Image": {
                "Ref": "ImageUrl"
            },
            "PortMappings": [
                {
                    "ContainerPort": {
                        "Ref": "ContainerPort"
                    },
                    "HostPort": {
                        "Ref": "ContainerPort"
                    }
                }
            ],
            "LogConfiguration": {
                "LogDriver": "awslogs",
                "Options": {
                    "mode": "non-blocking",
                    "max-buffer-size": "25m",
```

```

        "awslogs-group": {
            "Ref": "LogGroup"
        },
        "awslogs-region": {
            "Ref": "AWS::Region"
        },
        "awslogs-stream-prefix": {
            "Ref": "ServiceName"
        }
    }
}

],
}

},
"Service": {
    "Type": "AWS::ECS::Service",
    "DependsOn": "PublicLoadBalancerListener",
    "Properties": {
        "ServiceName": {
            "Ref": "ServiceName"
        },
        "Cluster": {
            "Ref": "ClusterName"
        },
        "PlacementStrategies": [
            {
                "Field": "attribute:ecs.availability-zone",
                "Type": "spread"
            },
            {
                "Field": "cpu",
                "Type": "binpack"
            }
        ],
        "CapacityProviderStrategy": [
            {
                "Base": 0,
                "CapacityProvider": {
                    "Ref": "CapacityProvider"
                },
                "Weight": 1
            }
        ]
    }
},

```

```

    "NetworkConfiguration": {
      "AwsVpcConfiguration": {
        "SecurityGroups": [
          {
            "Ref": "ServiceSecurityGroup"
          }
        ],
        "Subnets": {
          "Ref": "PrivateSubnetIds"
        }
      }
    },
    "DeploymentConfiguration": {
      "MaximumPercent": 200,
      "MinimumHealthyPercent": 75
    },
    "DesiredCount": {
      "Ref": "DesiredCount"
    },
    "TaskDefinition": {
      "Ref": "TaskDefinition"
    },
    "LoadBalancers": [
      {
        "ContainerName": {
          "Ref": "ServiceName"
        },
        "ContainerPort": {
          "Ref": "ContainerPort"
        },
        "TargetGroupArn": {
          "Ref": "ServiceTargetGroup"
        }
      }
    ]
  },
  "ServiceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Security group for service",
      "VpcId": {
        "Ref": "VpcId"
      }
    }
  }
}

```

```

    }
  },
  "ServiceTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "HealthCheckIntervalSeconds": 6,
      "HealthCheckPath": "/",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 2,
      "TargetType": "ip",
      "Port": {
        "Ref": "ContainerPort"
      },
      "Protocol": "HTTP",
      "UnhealthyThresholdCount": 10,
      "VpcId": {
        "Ref": "VpcId"
      },
      "TargetGroupAttributes": [
        {
          "Key": "deregistration_delay.timeout_seconds",
          "Value": 0
        }
      ]
    }
  },
  "PublicLoadBalancerSG": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Access to the public facing load balancer",
      "VpcId": {
        "Ref": "VpcId"
      },
      "SecurityGroupIngress": [
        {
          "CidrIp": "0.0.0.0/0",
          "IpProtocol": -1
        }
      ]
    }
  },
  "PublicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",

```

```

    "Properties": {
      "Scheme": "internet-facing",
      "LoadBalancerAttributes": [
        {
          "Key": "idle_timeout.timeout_seconds",
          "Value": "30"
        }
      ],
      "Subnets": {
        "Ref": "PublicSubnetIds"
      },
      "SecurityGroups": [
        {
          "Ref": "PublicLoadBalancerSG"
        }
      ]
    },
    "PublicLoadBalancerListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "DefaultActions": [
          {
            "Type": "forward",
            "ForwardConfig": {
              "TargetGroups": [
                {
                  "TargetGroupArn": {
                    "Ref": "ServiceTargetGroup"
                  },
                  "Weight": 100
                }
              ]
            }
          }
        ],
        "LoadBalancerArn": {
          "Ref": "PublicLoadBalancer"
        },
        "Port": 80,
        "Protocol": "HTTP"
      }
    },
    "ServiceIngressfromLoadBalancer": {

```

```

        "Type": "AWS::EC2::SecurityGroupIngress",
        "Properties": {
            "Description": "Ingress from the public ALB",
            "GroupId": {
                "Ref": "ServiceSecurityGroup"
            },
            "IpProtocol": "-1",
            "SourceSecurityGroupId": {
                "Ref": "PublicLoadBalancerSG"
            }
        },
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup"
    }
}
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Description: >-
```

An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.

```
Parameters:
```

```
VpcId:
```

```
  Type: String
```

```
  Description: The VPC that the service is running inside of
```

```
PublicSubnetIds:
```

```
  Type: 'List<AWS::EC2::Subnet::Id>'
```

```
  Description: List of public subnet ID's to put the load balancer in
```

```
PrivateSubnetIds:
```

```
  Type: 'List<AWS::EC2::Subnet::Id>'
```

```
  Description: List of private subnet ID's that the AWS VPC tasks are in
```

```
ClusterName:
```

```
  Type: String
```

```
  Description: The name of the ECS cluster into which to launch capacity.
```

```
ECSTaskExecutionRole:
```

```
  Type: String
```

```
  Description: The role used to start up an ECS task
```

```
CapacityProvider:
```

```
Type: String
Description: >-
    The cluster capacity provider that the service should use to request
    capacity when it wants to start up a task
ServiceName:
    Type: String
    Default: web
    Description: A name for the service
ImageUrl:
    Type: String
    Default: 'public.ecr.aws/docker/library/nginx:latest'
    Description: >-
        The url of a docker image that contains the application process that will
        handle the traffic for this service
ContainerCpu:
    Type: Number
    Default: 256
    Description: How much CPU to give the container. 1024 is 1 CPU
ContainerMemory:
    Type: Number
    Default: 512
    Description: How much memory in megabytes to give the container
ContainerPort:
    Type: Number
    Default: 80
    Description: What port that the application expects traffic on
DesiredCount:
    Type: Number
    Default: 2
    Description: How many copies of the service task to run
Resources:
    TaskDefinition:
        Type: AWS::ECS::TaskDefinition
        Properties:
            Family: !Ref ServiceName
            Cpu: !Ref ContainerCpu
            Memory: !Ref ContainerMemory
            NetworkMode: awsvpc
            RequiresCompatibilities:
                - EC2
            ExecutionRoleArn: !Ref ECSTaskExecutionRole
            ContainerDefinitions:
                - Name: !Ref ServiceName
                  Cpu: !Ref ContainerCpu
```

```
    Memory: !Ref ContainerMemory
    Image: !Ref ImageUrl
    PortMappings:
      - ContainerPort: !Ref ContainerPort
        HostPort: !Ref ContainerPort
    LogConfiguration:
      LogDriver: awslogs
      Options:
        mode: non-blocking
        max-buffer-size: 25m
        awslogs-group: !Ref LogGroup
        awslogs-region: !Ref AWS::Region
        awslogs-stream-prefix: !Ref ServiceName
Service:
  Type: AWS::ECS::Service
  DependsOn: PublicLoadBalancerListener
  Properties:
    ServiceName: !Ref ServiceName
    Cluster: !Ref ClusterName
    PlacementStrategies:
      - Field: 'attribute:ecs.availability-zone'
        Type: spread
      - Field: cpu
        Type: binpack
    CapacityProviderStrategy:
      - Base: 0
        CapacityProvider: !Ref CapacityProvider
        Weight: 1
    NetworkConfiguration:
      AwsvpcConfiguration:
        SecurityGroups:
          - !Ref ServiceSecurityGroup
        Subnets: !Ref PrivateSubnetIds
    DeploymentConfiguration:
      MaximumPercent: 200
      MinimumHealthyPercent: 75
    DesiredCount: !Ref DesiredCount
    TaskDefinition: !Ref TaskDefinition
    LoadBalancers:
      - ContainerName: !Ref ServiceName
        ContainerPort: !Ref ContainerPort
        TargetGroupArn: !Ref ServiceTargetGroup
  ServiceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
```



```
Properties:
  GroupDescription: Security group for service
  VpcId: !Ref VpcId
ServiceTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 6
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 2
    TargetType: ip
    Port: !Ref ContainerPort
    Protocol: HTTP
    UnhealthyThresholdCount: 10
    VpcId: !Ref VpcId
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: 0
PublicLoadBalancerSG:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Access to the public facing load balancer
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        IpProtocol: -1
PublicLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    LoadBalancerAttributes:
      - Key: idle_timeout.timeout_seconds
        Value: '30'
    Subnets: !Ref PublicSubnetIds
    SecurityGroups:
      - !Ref PublicLoadBalancerSG
PublicLoadBalancerListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        ForwardConfig:
          TargetGroups:
```

```
- TargetGroupArn: !Ref ServiceTargetGroup
  Weight: 100
  LoadBalancerArn: !Ref PublicLoadBalancer
  Port: 80
  Protocol: HTTP
ServiceIngressfromLoadBalancer:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    Description: Ingress from the public ALB
    GroupId: !Ref ServiceSecurityGroup
    IpProtocol: -1
    SourceSecurityGroupId: !Ref PublicLoadBalancerSG
LogGroup:
  Type: AWS::Logs::LogGroup
```

Modelo de exemplo do Amazon Elastic File System

O Amazon Elastic File System (Amazon EFS) é um serviço de armazenamento de arquivos para instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Com o Amazon EFS, seus aplicativos têm storage quando precisam dela porque a capacidade de storage cresce e diminui automaticamente à medida que arquivos são adicionados e removidos.

O modelo de exemplo a seguir implanta instâncias do EC2 (em um grupo do Auto Scaling) que estão associadas a um sistema de arquivos do Amazon EFS. Para associar as instâncias ao sistema de arquivos, as instâncias executam o script auxiliar `cfn-init`, que faz download e instala o pacote `yum` do `nfs-utils`, cria um novo diretório e, em seguida, usa o nome DNS do sistema de arquivos para montar o sistema de arquivos no diretório. O nome DNS do sistema de arquivos é resolvido para um endereço IP de destino de montagem na Zona de disponibilidade da instância Amazon EC2. Para obter mais informações sobre a estrutura de nome DNS, consulte [Como montar sistemas de arquivos](#) no Guia do usuário do Amazon Elastic File System.

Para medir a atividade do Network File System, o modelo inclui métricas personalizadas do Amazon CloudWatch. O modelo também cria uma VPC, uma sub-rede e security groups. Para permitir que as instâncias se comuniquem com o sistema de arquivos, a VPC deve ter o DNS habilitado, e o destino de montagem e as instâncias EC2 devem estar na mesma Zona de disponibilidade (AZ) que é especificada pela sub-rede.

O security group do destino de montagem permite uma conexão de rede com a porta TCP 2049 que é necessária para que um cliente NFSv4 monte um sistema de arquivos. Para obter mais

informações sobre security groups para instâncias do EC2 e destinos de montagem, consulte [Segurança](#), no [Guia do usuário do Amazon Elastic File System](#).

Note

Se você fizer uma atualização no destino de montagem que faça com que ele seja substituído, as instâncias ou os aplicativos que usam o sistema de arquivos associado podem ser interrompidos. Isso pode fazer com que as gravações não confirmadas sejam perdidas. Para evitar interrupções, interrompa suas instâncias ao atualizar o destino de montagem definindo a capacidade desejada como zero. Isso permite que as instâncias desmontem o sistema de arquivos antes que o destino de montagem seja substituído. Após a conclusão da atualização da montagem, inicie suas instâncias em uma atualização posterior definindo a capacidade desejada.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. **WARNING** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.",
  "Parameters": {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [
        "t1.micro",
        "t2.nano",
        "t2.micro",
        "t2.small",
        "t2.medium",
        "t2.large",
        "m1.small",
        "m1.medium",
        "m1.large",
        "m1.xlarge",
        "m2.xlarge",
        "m2.2xlarge",
```

```
"m2.4xlarge",
"m3.medium",
"m3.large",
"m3.xlarge",
"m3.2xlarge",
"m4.large",
"m4.xlarge",
"m4.2xlarge",
"m4.4xlarge",
"m4.10xlarge",
"c1.medium",
"c1.xlarge",
"c3.large",
"c3.xlarge",
"c3.2xlarge",
"c3.4xlarge",
"c3.8xlarge",
"c4.large",
"c4.xlarge",
"c4.2xlarge",
"c4.4xlarge",
"c4.8xlarge",
"g2.2xlarge",
"g2.8xlarge",
"r3.large",
"r3.xlarge",
"r3.2xlarge",
"r3.4xlarge",
"r3.8xlarge",
"i2.xlarge",
"i2.2xlarge",
"i2.4xlarge",
"i2.8xlarge",
"d2.xlarge",
"d2.2xlarge",
"d2.4xlarge",
"d2.8xlarge",
"hi1.4xlarge",
"hs1.8xlarge",
"cr1.8xlarge",
"cc2.8xlarge",
"cg1.4xlarge"
],
"ConstraintDescription" : "must be a valid EC2 instance type."
```

```

    },
    "KeyName": {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Description": "Name of an existing EC2 key pair to enable SSH access to the EC2
instances"
    },
    "AsgMaxSize": {
      "Type": "Number",
      "Description": "Maximum size and initial desired capacity of Auto Scaling Group",
      "Default": "2"
    },
    "SSHLocation" : {
      "Description" : "The IP address range that can be used to connect to the EC2
instances by using SSH",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "VolumeName" : {
      "Description" : "The name to be used for the EFS volume",
      "Type": "String",
      "MinLength": "1",
      "Default": "myEFSvolume"
    },
    "MountPoint" : {
      "Description" : "The Linux mount point for the EFS volume",
      "Type": "String",
      "MinLength": "1",
      "Default": "myEFSvolume"
    }
  },
  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"      : { "Arch" : "HVM64" },
      "t2.nano"      : { "Arch" : "HVM64" },
      "t2.micro"      : { "Arch" : "HVM64" },
      "t2.small"     : { "Arch" : "HVM64" },
      "t2.medium"    : { "Arch" : "HVM64" },
      "t2.large"     : { "Arch" : "HVM64" },
      "m1.small"     : { "Arch" : "HVM64" },
      "m1.medium"    : { "Arch" : "HVM64" },

```

```
"m1.large"      : { "Arch" : "HVM64" },
"m1.xlarge"     : { "Arch" : "HVM64" },
"m2.xlarge"     : { "Arch" : "HVM64" },
"m2.2xlarge"    : { "Arch" : "HVM64" },
"m2.4xlarge"    : { "Arch" : "HVM64" },
"m3.medium"     : { "Arch" : "HVM64" },
"m3.large"      : { "Arch" : "HVM64" },
"m3.xlarge"     : { "Arch" : "HVM64" },
"m3.2xlarge"    : { "Arch" : "HVM64" },
"m4.large"      : { "Arch" : "HVM64" },
"m4.xlarge"     : { "Arch" : "HVM64" },
"m4.2xlarge"    : { "Arch" : "HVM64" },
"m4.4xlarge"    : { "Arch" : "HVM64" },
"m4.10xlarge"   : { "Arch" : "HVM64" },
"c1.medium"     : { "Arch" : "HVM64" },
"c1.xlarge"     : { "Arch" : "HVM64" },
"c3.large"      : { "Arch" : "HVM64" },
"c3.xlarge"     : { "Arch" : "HVM64" },
"c3.2xlarge"    : { "Arch" : "HVM64" },
"c3.4xlarge"    : { "Arch" : "HVM64" },
"c3.8xlarge"    : { "Arch" : "HVM64" },
"c4.large"      : { "Arch" : "HVM64" },
"c4.xlarge"     : { "Arch" : "HVM64" },
"c4.2xlarge"    : { "Arch" : "HVM64" },
"c4.4xlarge"    : { "Arch" : "HVM64" },
"c4.8xlarge"    : { "Arch" : "HVM64" },
"g2.2xlarge"    : { "Arch" : "HVMG2" },
"g2.8xlarge"    : { "Arch" : "HVMG2" },
"r3.large"      : { "Arch" : "HVM64" },
"r3.xlarge"     : { "Arch" : "HVM64" },
"r3.2xlarge"    : { "Arch" : "HVM64" },
"r3.4xlarge"    : { "Arch" : "HVM64" },
"r3.8xlarge"    : { "Arch" : "HVM64" },
"i2.xlarge"     : { "Arch" : "HVM64" },
"i2.2xlarge"    : { "Arch" : "HVM64" },
"i2.4xlarge"    : { "Arch" : "HVM64" },
"i2.8xlarge"    : { "Arch" : "HVM64" },
"d2.xlarge"     : { "Arch" : "HVM64" },
"d2.2xlarge"    : { "Arch" : "HVM64" },
"d2.4xlarge"    : { "Arch" : "HVM64" },
"d2.8xlarge"    : { "Arch" : "HVM64" },
"hi1.4xlarge"   : { "Arch" : "HVM64" },
"hs1.8xlarge"   : { "Arch" : "HVM64" },
"cr1.8xlarge"   : { "Arch" : "HVM64" },
```

```

    "cc2.8xlarge" : { "Arch" : "HVM64"  }
  },
  "AWSRegionArch2AMI" : {
    "us-east-1"      : { "HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
    "us-west-2"      : { "HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
    "us-west-1"      : { "HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
    "eu-west-1"      : { "HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
    "eu-west-2"      : { "HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
    "eu-west-3"      : { "HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
    "eu-central-1"   : { "HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},
    "ap-northeast-1" : { "HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
    "ap-northeast-2" : { "HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-northeast-3" : { "HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-southeast-1" : { "HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
    "ap-southeast-2" : { "HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
    "ap-south-1"     : { "HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},
    "us-east-2"      : { "HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
    "ca-central-1"   : { "HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
    "sa-east-1"      : { "HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-north-1"     : { "HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-northwest-1" : { "HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
  }
},
"Resources": {
  "CloudWatchPutMetricsRole" : {
    "Type" : "AWS::IAM::Role",
    "Properties" : {
      "AssumeRolePolicyDocument" : {
        "Statement" : [ {

```

```

        "Effect" : "Allow",
        "Principal" : {
            "Service" : [ "ec2.amazonaws.com" ]
        },
        "Action" : [ "sts:AssumeRole" ]
    } ]
},
"Path" : "/"
}
},
"CloudWatchPutMetricsRolePolicy" : {
    "Type" : "AWS::IAM::Policy",
    "Properties" : {
        "PolicyName" : "CloudWatch_PutMetricData",
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "CloudWatchPutMetricData",
                    "Effect": "Allow",
                    "Action": ["cloudwatch:PutMetricData"],
                    "Resource": ["*"]
                }
            ]
        },
        "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
    }
},
"CloudWatchPutMetricsInstanceProfile" : {
    "Type" : "AWS::IAM::InstanceProfile",
    "Properties" : {
        "Path" : "/",
        "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
    }
},
"VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
        "EnableDnsSupport" : "true",
        "EnableDnsHostnames" : "true",
        "CidrBlock": "10.0.0.0/16",
        "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
    }
},

```



```
"InternetGateway" : {
  "Type" : "AWS::EC2::InternetGateway",
  "Properties" : {
    "Tags" : [
      { "Key" : "Application", "Value" : { "Ref" : "AWS::StackName" } },
      { "Key" : "Network", "Value" : "Public" }
    ]
  }
},
"GatewayToInternet" : {
  "Type" : "AWS::EC2::VPCGatewayAttachment",
  "Properties" : {
    "VpcId" : { "Ref" : "VPC" },
    "InternetGatewayId" : { "Ref" : "InternetGateway" }
  }
},
"RouteTable":{
  "Type":"AWS::EC2::RouteTable",
  "Properties":{
    "VpcId": {"Ref":"VPC"}
  }
},
"SubnetRouteTableAssoc": {
  "Type" : "AWS::EC2::SubnetRouteTableAssociation",
  "Properties" : {
    "RouteTableId" : {"Ref":"RouteTable"},
    "SubnetId" : {"Ref":"Subnet"}
  }
},
"InternetGatewayRoute": {
  "Type":"AWS::EC2::Route",
  "Properties":{
    "DestinationCidrBlock":"0.0.0.0/0",
    "RouteTableId":{"Ref":"RouteTable"},
    "GatewayId":{"Ref":"InternetGateway"}
  }
},
"Subnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "VpcId": { "Ref": "VPC" },
    "CidrBlock": "10.0.0.0/24",
    "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
  }
}
```

```
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
          { "IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "CidrIp": { "Ref":
"SSHLocation" } },
          { "IpProtocol": "tcp", "FromPort": 80, "ToPort": 80, "CidrIp": "0.0.0.0/0" }
        ]
      }
    },
    "MountTargetSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "GroupDescription": "Security group for mount target",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 2049,
            "ToPort": 2049,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    },
    "FileSystem": {
      "Type": "AWS::EFS::FileSystem",
      "Properties": {
        "PerformanceMode": "generalPurpose",
        "FileSystemTags": [
          {
            "Key": "Name",
            "Value": { "Ref" : "VolumeName" }
          }
        ]
      }
    },
    "MountTarget": {
      "Type": "AWS::EFS::MountTarget",
      "Properties": {
        "FileSystemId": { "Ref": "FileSystem" },
```

```

    "SubnetId": { "Ref": "Subnet" },
    "SecurityGroups": [ { "Ref": "MountTargetSecurityGroup" } ]
  }
},
"LaunchConfiguration": {
  "Type": "AWS::AutoScaling::LaunchConfiguration",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "configSets" : {
        "MountConfig" : [ "setup", "mount" ]
      },
      "setup" : {
        "packages" : {
          "yum" : {
            "nfs-utils" : []
          }
        },
        "files" : {
          "/home/ec2-user/post_nfsstat" : {
            "content" : { "Fn::Join" : [ "", [
              "#!/bin/bash\n",
              "\n",
              "INPUT=\"$(cat)\"\n",
              "CW_JSON_OPEN='{ \"Namespace\": \"EFS\", \"MetricData\": [ '\n",
              "CW_JSON_CLOSE=' ] }'\n",
              "CW_JSON_METRIC=' '\n",
              "METRIC_COUNTER=0\n",
              "\n",
              "for COL in 1 2 3 4 5 6; do\n",
              "\n",
              "  COUNTER=0\n",
              "  METRIC_FIELD=$COL\n",
              "  DATA_FIELD=$(( $COL + ( $COL - 1 ) ) )\n",
              "\n",
              "  while read line; do\n",
              "    if [[ COUNTER -gt 0 ]]; then\n",
              "\n",
              "      LINE=`echo $line | tr -s ' ' '\n`,\n",
              "      AWS_COMMAND=\"aws cloudwatch put-metric-data --region ",
            { "Ref": "AWS::Region" }, "\n",
              "      MOD=$(( $COUNTER % 2 ))\n",
              "\n",
              "      if [ $MOD -eq 1 ]; then\n",

```

```

        METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
\n",
        "    else\n",
        "        METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`\n",
        "    fi\n",
        "\n",
        "    if [[ -n \"$METRIC_NAME\" && -n \"$METRIC_VALUE\" ]]; then
\n",
        "        INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-
data/instance-id)\n",
        "        CW_JSON_METRIC=\"${CW_JSON_METRIC} { \\\"MetricName\\\": \\\"
\\\"$METRIC_NAME\\\", \\\"Dimensions\\\": [{\\\"Name\\\": \\\"InstanceId\\\", \\\"Value\\\"
\\\": \\\"$INSTANCE_ID\\\"} ], \\\"Value\\\": $METRIC_VALUE },\\\"\\n",
        "        unset METRIC_NAME\n",
        "        unset METRIC_VALUE\n",
        "\n",
        "        METRIC_COUNTER=$((METRIC_COUNTER+1))\n",
        "        if [ $METRIC_COUNTER -eq 20 ]; then\n",
        "            # 20 is max metric collection size, so we have to
submit here\n",
        "            aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\n",
        "\n",
        "            # reset\n",
        "            METRIC_COUNTER=0\n",
        "            CW_JSON_METRIC='' \n",
        "        fi\n",
        "    fi \n",
        "\n",
        "\n",
        "\n",
        "        COUNTER=$((COUNTER+1))\n",
        "    fi\n",
        "\n",
        "    if [[ \"$line\" == \"Client nfs v4:\" ]]; then\n",
        "        # the next line is the good stuff \n",
        "        COUNTER=$((COUNTER+1))\n",
        "    fi\n",
        "done <<< \"$INPUT\"\n",
        "done\n",
        "\n",
        "# submit whatever is left\n",

```

```

        "aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\"
    ] ] },
    "mode": "000755",
    "owner": "ec2-user",
    "group": "ec2-user"
  },
  "/home/ec2-user/crontab" : {
    "content" : { "Fn::Join" : [ "", [
      "** * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
    ] ] },
    "owner": "ec2-user",
    "group": "ec2-user"
  }
},
"commands" : {
  "01_createdir" : {
    "command" : {"Fn::Join" : [ "", [ "mkdir /", { "Ref" :
"MountPoint" }]]]
  }
},
"mount" : {
  "commands" : {
    "01_mount" : {
      "command" : { "Fn::Sub": "sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}"
    },
    "02_permissions" : {
      "command" : {"Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
{ "Ref" : "MountPoint" }]]]
    }
  }
},
"Properties": {
  "AssociatePublicIpAddress" : true,
  "ImageId": {
    "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
      "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
"Arch" ]

```

```

    } ]
  },
  "InstanceType": { "Ref": "InstanceType" },
  "KeyName": { "Ref": "KeyName" },
  "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],
  "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
  "UserData"          : { "Fn::Base64" : { "Fn::Join" : [ "", [
    "#!/bin/bash -xe\n",
    "yum install -y aws-cfn-bootstrap\n",

    "/opt/aws/bin/cfn-init -v ",
    "    --stack ", { "Ref" : "AWS::StackName" },
    "    --resource LaunchConfiguration ",
    "    --configsets MountConfig ",
    "    --region ", { "Ref" : "AWS::Region" }, "\n",

    "crontab /home/ec2-user/crontab\n",

    "/opt/aws/bin/cfn-signal -e $? ",
    "    --stack ", { "Ref" : "AWS::StackName" },
    "    --resource AutoScalingGroup ",
    "    --region ", { "Ref" : "AWS::Region" }, "\n"
  ] ] ] }
  ] ] ]
}
},
"AutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": ["MountTarget", "GatewayToInternet"],
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT15M",
      "Count"   : { "Ref": "AsgMaxSize" }
    }
  }
},
"Properties": {
  "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
  "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
  "MinSize": "1",
  "MaxSize": { "Ref": "AsgMaxSize" },
  "DesiredCapacity": { "Ref": "AsgMaxSize" },
  "Tags": [ {
    "Key": "Name",
    "Value": "EFS FileSystem Mounted Instance",
    "PropagateAtLaunch": "true"
  } ]
}

```

```

    } ]
  }
}
},
"Outputs" : {
  "MountTargetID" : {
    "Description" : "Mount target ID",
    "Value" : { "Ref" : "MountTarget" }
  },
  "FileSystemID" : {
    "Description" : "File system ID",
    "Value" : { "Ref" : "FileSystem" }
  }
}
}
}

```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. ****WARNING**** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.

Parameters:

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.small

AllowedValues:

- t1.micro
- t2.nano
- t2.micro
- t2.small
- t2.medium
- t2.large
- m1.small
- m1.medium
- m1.large
- m1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- m3.medium

- m3.large
- m3.xlarge
- m3.2xlarge
- m4.large
- m4.xlarge
- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- c1.medium
- c1.xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- g2.2xlarge
- g2.8xlarge
- r3.large
- r3.xlarge
- r3.2xlarge
- r3.4xlarge
- r3.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- d2.xlarge
- d2.2xlarge
- d2.4xlarge
- d2.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- cr1.8xlarge
- cc2.8xlarge
- cg1.4xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Type: AWS::EC2::KeyPair::KeyName

Description: Name of an existing EC2 key pair to enable SSH access to the ECS


```
instances
AsgMaxSize:
  Type: Number
  Description: Maximum size and initial desired capacity of Auto Scaling Group
  Default: '2'
SSHLocation:
  Description: The IP address range that can be used to connect to the EC2 instances
    by using SSH
  Type: String
  MinLength: '9'
  MaxLength: '18'
  Default: 0.0.0.0/0
  AllowedPattern: "(\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})/((\\d{1,2})"
  ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
VolumeName:
  Description: The name to be used for the EFS volume
  Type: String
  MinLength: '1'
  Default: myEFSvolume
MountPoint:
  Description: The Linux mount point for the EFS volume
  Type: String
  MinLength: '1'
  Default: myEFSvolume
Mappings:
  AWSInstanceType2Arch:
    t1.micro:
      Arch: HVM64
    t2.nano:
      Arch: HVM64
    t2.micro:
      Arch: HVM64
    t2.small:
      Arch: HVM64
    t2.medium:
      Arch: HVM64
    t2.large:
      Arch: HVM64
    m1.small:
      Arch: HVM64
    m1.medium:
      Arch: HVM64
    m1.large:
      Arch: HVM64
```

```
m1.xlarge:
  Arch: HVM64
m2.xlarge:
  Arch: HVM64
m2.2xlarge:
  Arch: HVM64
m2.4xlarge:
  Arch: HVM64
m3.medium:
  Arch: HVM64
m3.large:
  Arch: HVM64
m3.xlarge:
  Arch: HVM64
m3.2xlarge:
  Arch: HVM64
m4.large:
  Arch: HVM64
m4.xlarge:
  Arch: HVM64
m4.2xlarge:
  Arch: HVM64
m4.4xlarge:
  Arch: HVM64
m4.10xlarge:
  Arch: HVM64
c1.medium:
  Arch: HVM64
c1.xlarge:
  Arch: HVM64
c3.large:
  Arch: HVM64
c3.xlarge:
  Arch: HVM64
c3.2xlarge:
  Arch: HVM64
c3.4xlarge:
  Arch: HVM64
c3.8xlarge:
  Arch: HVM64
c4.large:
  Arch: HVM64
c4.xlarge:
  Arch: HVM64
```

```
c4.2xlarge:
  Arch: HVM64
c4.4xlarge:
  Arch: HVM64
c4.8xlarge:
  Arch: HVM64
g2.2xlarge:
  Arch: HVMG2
g2.8xlarge:
  Arch: HVMG2
r3.large:
  Arch: HVM64
r3.xlarge:
  Arch: HVM64
r3.2xlarge:
  Arch: HVM64
r3.4xlarge:
  Arch: HVM64
r3.8xlarge:
  Arch: HVM64
i2.xlarge:
  Arch: HVM64
i2.2xlarge:
  Arch: HVM64
i2.4xlarge:
  Arch: HVM64
i2.8xlarge:
  Arch: HVM64
d2.xlarge:
  Arch: HVM64
d2.2xlarge:
  Arch: HVM64
d2.4xlarge:
  Arch: HVM64
d2.8xlarge:
  Arch: HVM64
hi1.4xlarge:
  Arch: HVM64
hs1.8xlarge:
  Arch: HVM64
cr1.8xlarge:
  Arch: HVM64
cc2.8xlarge:
  Arch: HVM64
```

AWSRegionArch2AMI:**us-east-1:**

HVM64: ami-0ff8a91507f77f867

HVMG2: ami-0a584ac55a7631c0c

us-west-2:

HVM64: ami-a0cfeed8

HVMG2: ami-0e09505bc235aa82d

us-west-1:

HVM64: ami-0bdb828fd58c52235

HVMG2: ami-066ee5fd4a9ef77f1

eu-west-1:

HVM64: ami-047bb4163c506cd98

HVMG2: ami-0a7c483d527806435

eu-west-2:

HVM64: ami-f976839e

HVMG2: NOT_SUPPORTED

eu-west-3:

HVM64: ami-0ebc281c20e89ba4b

HVMG2: NOT_SUPPORTED

eu-central-1:

HVM64: ami-0233214e13e500f77

HVMG2: ami-06223d46a6d0661c7

ap-northeast-1:

HVM64: ami-06cd52961ce9f0d85

HVMG2: ami-053cdd503598e4a9d

ap-northeast-2:

HVM64: ami-0a10b2721688ce9d2

HVMG2: NOT_SUPPORTED

ap-northeast-3:

HVM64: ami-0d98120a9fb693f07

HVMG2: NOT_SUPPORTED

ap-southeast-1:

HVM64: ami-08569b978cc4dfa10

HVMG2: ami-0be9df32ae9f92309

ap-southeast-2:

HVM64: ami-09b42976632b27e9b

HVMG2: ami-0a9ce9fecc3d1daf8

ap-south-1:

HVM64: ami-0912f71e06545ad88

HVMG2: ami-097b15e89dbdcfcf4

us-east-2:

HVM64: ami-0b59bfac6be064b78

HVMG2: NOT_SUPPORTED

ca-central-1:

```
HVM64: ami-0b18956f
HVMG2: NOT_SUPPORTED
sa-east-1:
  HVM64: ami-07b14488da8ea02a0
  HVMG2: NOT_SUPPORTED
cn-north-1:
  HVM64: ami-0a4eaf6c4454eda75
  HVMG2: NOT_SUPPORTED
cn-northwest-1:
  HVM64: ami-6b6a7d09
  HVMG2: NOT_SUPPORTED
```

Resources:**CloudWatchPutMetricsRole:**

Type: AWS::IAM::Role

Properties:**AssumeRolePolicyDocument:****Statement:**

- Effect: Allow

Principal:**Service:**

- ec2.amazonaws.com

Action:

- sts:AssumeRole

Path: "/"

CloudWatchPutMetricsRolePolicy:

Type: AWS::IAM::Policy

Properties:

PolicyName: CloudWatch_PutMetricData

PolicyDocument:

Version: '2012-10-17'

Statement:

- Sid: CloudWatchPutMetricData

Effect: Allow

Action:

- cloudwatch:PutMetricData

Resource:

- "*"

Roles:

- Ref: CloudWatchPutMetricsRole

CloudWatchPutMetricsInstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Path: "/"

Roles:

- Ref: CloudWatchPutMetricsRole

VPC:

Type: AWS::EC2::VPC

Properties:

EnableDnsSupport: 'true'

EnableDnsHostnames: 'true'

CidrBlock: 10.0.0.0/16

Tags:

- Key: Application

Value:

Ref: AWS::StackId

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:**Tags:**

- Key: Application

Value:

Ref: AWS::StackName

- Key: Network

Value: Public

GatewayToInternet:

Type: AWS::EC2::VPCEGatewayAttachment

Properties:

VpcId:

Ref: VPC

InternetGatewayId:

Ref: InternetGateway

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId:

Ref: VPC

SubnetRouteTableAssoc:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId:

Ref: RouteTable

SubnetId:

Ref: Subnet

InternetGatewayRoute:

Type: AWS::EC2::Route

Properties:

DestinationCidrBlock: 0.0.0.0/0

RouteTableId:

```
    Ref: RouteTable
  GatewayId:
    Ref: InternetGateway
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp:
          Ref: SSHLocation
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
MountTargetSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Security group for mount target
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 2049
        ToPort: 2049
        CidrIp: 0.0.0.0/0
FileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    PerformanceMode: generalPurpose
```

```

    FileSystemTags:
      - Key: Name
        Value:
          Ref: VolumeName
MountTarget:
  Type: AWS::EFS::MountTarget
  Properties:
    FileSystemId:
      Ref: FileSystem
    SubnetId:
      Ref: Subnet
    SecurityGroups:
      - Ref: MountTargetSecurityGroup
LaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        MountConfig:
          - setup
          - mount
      setup:
        packages:
          yum:
            nfs-utils: []
        files:
          "/home/ec2-user/post_nfsstat":
            content: !Sub |
              #!/bin/bash

              INPUT="$(cat)"
              CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
              CW_JSON_CLOSE=' ] }'
              CW_JSON_METRIC=''
              METRIC_COUNTER=0

              for COL in 1 2 3 4 5 6; do

                COUNTER=0
                METRIC_FIELD=$COL
                DATA_FIELD=$(( $COL + ( $COL - 1 ) ) )

                while read line; do
                  if [[ COUNTER -gt 0 ]]; then

```



```

        LINE=`echo $line | tr -s ' ' `
        AWS_COMMAND="aws cloudwatch put-metric-data --region
${AWS::Region}"
        MOD=$(( $COUNTER % 2))

        if [ $MOD -eq 1 ]; then
            METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
        else
            METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
        fi

        if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
            INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)
            CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"
            unset METRIC_NAME
            unset METRIC_VALUE

            METRIC_COUNTER=$((METRIC_COUNTER+1))
            if [ $METRIC_COUNTER -eq 20 ]; then
                # 20 is max metric collection size, so we have to submit here
                aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                # reset
                METRIC_COUNTER=0
                CW_JSON_METRIC=''
            fi
        fi

        COUNTER=$((COUNTER+1))
    fi

    if [[ "$line" == "Client nfs v4:" ]]; then
        # the next line is the good stuff
        COUNTER=$((COUNTER+1))
    fi
done <<< "$INPUT"
done

```

```

        # submit whatever is left
        aws cloudwatch put-metric-data --region ${AWS::Region} --cli-input-json
    "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"
    mode: '000755'
    owner: ec2-user
    group: ec2-user
    "/home/ec2-user/crontab":
    content: "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
    owner: ec2-user
    group: ec2-user
    commands:
    01_createdir:
    command: !Sub "mkdir /${MountPoint}"
    mount:
    commands:
    01_mount:
    command: !Sub >
        mount -t nfs4 -o nfsvers=4.1 ${FileSystem}.efs.
        ${AWS::Region}.amazonaws.com:/ /${MountPoint}
    02_permissions:
    command: !Sub "chown ec2-user:ec2-user /${MountPoint}"
    Properties:
    AssociatePublicIpAddress: true
    ImageId:
    Fn::FindInMap:
    - AWSRegionArch2AMI
    - Ref: AWS::Region
    - Fn::FindInMap:
    - AWSInstanceType2Arch
    - Ref: InstanceType
    - Arch
    InstanceType:
    Ref: InstanceType
    KeyName:
    Ref: KeyName
    SecurityGroups:
    - Ref: InstanceSecurityGroup
    IamInstanceProfile:
    Ref: CloudWatchPutMetricsInstanceProfile
    UserData:
    Fn::Base64: !Sub |
        #!/bin/bash -xe
        yum install -y aws-cfn-bootstrap

```

```
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchConfiguration --configsets MountConfig --region ${AWS::Region}
    crontab /home/ec2-user/crontab
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
AutoScalingGroup --region ${AWS::Region}
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - MountTarget
    - GatewayToInternet
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
      Count:
        Ref: AsgMaxSize
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    LaunchConfigurationName:
      Ref: LaunchConfiguration
    MinSize: '1'
    MaxSize:
      Ref: AsgMaxSize
    DesiredCapacity:
      Ref: AsgMaxSize
    Tags:
      - Key: Name
        Value: EFS FileSystem Mounted Instance
        PropagateAtLaunch: 'true'
Outputs:
  MountTargetID:
    Description: Mount target ID
    Value:
      Ref: MountTarget
  FileSystemID:
    Description: File system ID
    Value:
      Ref: FileSystem
```

Trechos de modelo do Elastic Beanstalk

Com o Elastic Beanstalk, você pode implantar e gerenciar aplicações rapidamente na AWS sem se preocupar com a infraestrutura na qual essas aplicações são executadas. O modelo de exemplo a seguir pode ajudar a descrever os recursos do Elastic Beanstalk em seu modelo do CloudFormation.

PHP de exemplo do Elastic Beanstalk

O modelo de exemplo a seguir implementa um aplicativo Web de exemplo em PHP que é armazenado em um bucket do Amazon S3. O ambiente também é um ambiente de ajuste de escala automático com balanceamento de carga, com um mínimo de duas e um máximo de seis instâncias do Amazon EC2. Ele mostra um ambiente do Elastic Beanstalk que usa uma configuração de lançamento legada. Para obter informações sobre como usar um modelo de lançamento em vez disso, consulte [Modelos de lançamento](#) no Guia do desenvolvedor do AWS Elastic Beanstalk.

Substitua *solution-stack* por um nome de pilha de soluções (versão da plataforma). Para obter uma lista de pilhas de soluções disponíveis, use o comando `aws elasticbeanstalk list-available-solution-stacks` da AWS CLI.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "sampleApplication": {
      "Type": "AWS::ElasticBeanstalk::Application",
      "Properties": {
        "Description": "AWS Elastic Beanstalk Sample Application"
      }
    },
    "sampleApplicationVersion": {
      "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
      "Properties": {
        "ApplicationName": {
          "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Application Version",
        "SourceBundle": {
          "S3Bucket": {
            "Fn::Sub": "elasticbeanstalk-samples-${AWS::Region}"
          },
          "S3Key": "php-newsample-app.zip"
        }
      }
    }
  }
}
```

```

    }
  },
  "sampleConfigurationTemplate": {
    "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
    "Properties": {
      "ApplicationName": {
        "Ref": "sampleApplication"
      },
      "Description": "AWS ElasticBeanstalk Sample Configuration Template",
      "OptionSettings": [
        {
          "Namespace": "aws:autoscaling:asg",
          "OptionName": "MinSize",
          "Value": "2"
        },
        {
          "Namespace": "aws:autoscaling:asg",
          "OptionName": "MaxSize",
          "Value": "6"
        },
        {
          "Namespace": "aws:elasticbeanstalk:environment",
          "OptionName": "EnvironmentType",
          "Value": "LoadBalanced"
        },
        {
          "Namespace": "aws:autoscaling:launchconfiguration",
          "OptionName": "IamInstanceProfile",
          "Value": {
            "Ref": "MyInstanceProfile"
          }
        }
      ],
      "SolutionStackName": "solution-stack"
    }
  },
  "sampleEnvironment": {
    "Type": "AWS::ElasticBeanstalk::Environment",
    "Properties": {
      "ApplicationName": {
        "Ref": "sampleApplication"
      },
      "Description": "AWS ElasticBeanstalk Sample Environment",

```

```

        "TemplateName": {
            "Ref": "sampleConfigurationTemplate"
        },
        "VersionLabel": {
            "Ref": "sampleApplicationVersion"
        }
    },
    "MyInstanceRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ]
                    }
                ]
            },
            "Description": "Beanstalk EC2 role",
            "ManagedPolicyArns": [
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier",
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker",
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier"
            ]
        }
    },
    "MyInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Roles": [
                {
                    "Ref": "MyInstanceRole"
                }
            ]
        }
    }
}

```

```

    }
  }
}
```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  sampleApplication:
    Type: AWS::ElasticBeanstalk::Application
    Properties:
      Description: AWS Elastic Beanstalk Sample Application
  sampleApplicationVersion:
    Type: AWS::ElasticBeanstalk::ApplicationVersion
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Application Version
      SourceBundle:
        S3Bucket: !Sub "elasticbeanstalk-samples-${AWS::Region}"
        S3Key: php-newsample-app.zip
  sampleConfigurationTemplate:
    Type: AWS::ElasticBeanstalk::ConfigurationTemplate
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Configuration Template
      OptionSettings:
        - Namespace: aws:autoscaling:asg
          OptionName: MinSize
          Value: '2'
        - Namespace: aws:autoscaling:asg
          OptionName: MaxSize
          Value: '6'
        - Namespace: aws:elasticbeanstalk:environment
          OptionName: EnvironmentType
          Value: LoadBalanced
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile
          Value: !Ref MyInstanceProfile
      SolutionStackName: solution-stack
  sampleEnvironment:
    Type: AWS::ElasticBeanstalk::Environment
```

```
Properties:
  ApplicationName:
    Ref: sampleApplication
  Description: AWS ElasticBeanstalk Sample Environment
  TemplateName:
    Ref: sampleConfigurationTemplate
  VersionLabel:
    Ref: sampleApplicationVersion
MyInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Description: Beanstalk EC2 role
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier
MyInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref MyInstanceRole
```

Trechos de modelo do Elastic Load Balancing

Para criar um Application Load Balancer, um Network Load Balancer, um Gateway Load Balancer, use os tipos de recursos do V2, que começam com `AWS::ElasticLoadBalancingV2`.

Para criar um Classic Load Balancer, use os tipos de recursos que começam com `AWS::ElasticLoadBalancing`.

Conteúdo

- [Recursos do ELBv2](#)
- [Recursos do Classic Load Balancer](#)

Recursos do ELBv2

Este exemplo define um Application Load Balancer com um receptor HTTP e uma ação padrão que encaminha o tráfego para o grupo-alvo. O balanceador de carga usa as configurações padrão de verificação de integridade. O grupo-alvo tem duas instâncias do EC2 registradas.

YAML

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myHTTPListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup

  myTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      Name: "my-target-group"
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
```

Port: 80

JSON

```
{
  "Resources": {
    "myLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
        "Name": "my-alb",
        "Type": "application",
        "Scheme": "internal",
        "Subnets": [
          {
            "Ref": "subnet-AZ1"
          },
          {
            "Ref": "subnet-AZ2"
          }
        ],
        "SecurityGroups": [
          {
            "Ref": "mySecurityGroup"
          }
        ]
      }
    },
    "myHTTPListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "LoadBalancerArn": {
          "Ref": "myLoadBalancer"
        },
        "Protocol": "HTTP",
        "Port": 80,
        "DefaultActions": [
          {
            "Type": "forward",
            "TargetGroupArn": {
              "Ref": "myTargetGroup"
            }
          }
        ]
      }
    }
  }
}
```

```

    }
  },
  "myTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "Name": "my-target-group",
      "Protocol": "HTTP",
      "Port": 80,
      "TargetType": "instance",
      "VpcId": {
        "Ref": "myVPC"
      },
      "Targets": [
        {
          "Id": {
            "Fn::GetAtt": [
              "Instance1",
              "InstanceId"
            ]
          },
          "Port": 80
        },
        {
          "Id": {
            "Fn::GetAtt": [
              "Instance2",
              "InstanceId"
            ]
          },
          "Port": 80
        }
      ]
    }
  }
}

```

Recursos do Classic Load Balancer

Este exemplo define um Classic Load Balancer com um receptor HTTP e nenhuma instância do EC2 registrada. O balanceador de carga usa as configurações padrão de verificação de integridade.

YAML

```
myLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
```

JSON

```
"myLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
}
```

Este exemplo define um Classic Load Balancer com um receptor HTTP, duas instâncias do EC2 registradas e configurações personalizadas de verificação de integridade.

YAML

```
myClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Instances:
      - Ref: Instance1
      - Ref: Instance2
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
```

```
Protocol: HTTP
HealthCheck:
  Target: HTTP:80/
  HealthyThreshold: '3'
  UnhealthyThreshold: '5'
  Interval: '30'
  Timeout: '5'
```

JSON

```
"myClassicLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Instances" : [
      { "Ref" : "Instance1" },
      { "Ref" : "Instance2" }
    ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
}
```

Trechos de modelo do AWS Identity and Access Management

Esta seção contém trechos de modelos do AWS Identity and Access Management.

Tópicos

- [Declarar um recurso de usuário do IAM](#)
- [Declarar um recurso de chave de acesso do IAM](#)

- [Declarar um recurso de grupo do IAM](#)
- [Adicionar usuários a um grupo](#)
- [Declarar uma política do IAM](#)
- [Declarar uma política do bucket do Amazon S3](#)
- [Declarar uma política de tópico do Amazon SNS](#)
- [Declarar uma política do Amazon SQS](#)
- [Exemplos de modelos de função do IAM](#)

Important

Ao criar ou atualizar uma pilha usando um modelo que contém recursos do IAM, confirme o uso de recursos do IAM. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).

Declarar um recurso de usuário do IAM

Este trecho mostra como declarar um recurso [AWS::IAM::User](#) para criar um usuário do IAM. O usuário é declarado com o caminho ("/") e um perfil de login com a senha (myP@ssW0rd).

O documento de política chamado `giveaccesstoqueueonly` oferece ao usuário permissão para executar todas as ações do Amazon SQS no `myqueue` do recurso de fila do Amazon SQS e nega o acesso a todos os outros recursos de fila do Amazon SQS. A função `Fn::GetAtt` obtém o atributo `Arn` do recurso [AWS::SQS::Queue](#) denominado `myqueue`.

O documento de política chamado `giveaccesstotopiconly` é adicionado ao usuário para que ele tenha permissão para executar todas as ações Amazon SNS no recurso `mytopic` do tópico do Amazon SNS e negar acesso a todos os outros recursos do Amazon SNS. A função `Ref` [AWS::SNS::Topic obtém o Nome de região da Amazon \(ARN\) do mytopic do recurso](#).

JSON

```
"myuser" : {
  "Type" : "AWS::IAM::User",
  "Properties" : {
    "Path" : "/",
    "LoginProfile" : {
```

```

    "Password" : "myP@ssW0rd"
  },
  "Policies" : [ {
    "PolicyName" : "giveaccesstoqueueonly",
    "PolicyDocument" : {
      "Version": "2012-10-17",
      "Statement" : [ {
        "Effect" : "Allow",
        "Action" : [ "sqs:*" ],
        "Resource" : [ {
          "Fn::GetAtt" : [ "myqueue", "Arn" ]
        } ]
      }, {
        "Effect" : "Deny",
        "Action" : [ "sqs:*" ],
        "NotResource" : [ {
          "Fn::GetAtt" : [ "myqueue", "Arn" ]
        } ]
      }
    ]
  }, {
    "PolicyName" : "giveaccesstotopiconly",
    "PolicyDocument" : {
      "Version": "2012-10-17",
      "Statement" : [ {
        "Effect" : "Allow",
        "Action" : [ "sns:*" ],
        "Resource" : [ { "Ref" : "mytopic" } ]
      }, {
        "Effect" : "Deny",
        "Action" : [ "sns:*" ],
        "NotResource" : [ { "Ref" : "mytopic" } ]
      } ]
    }
  } ]
}

```

YAML

```

myuser:
  Type: AWS::IAM::User
  Properties:

```

```
Path: "/"
LoginProfile:
  Password: myP@ssW0rd
Policies:
- PolicyName: giveaccesstoqueueonly
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
    - Effect: Allow
      Action:
      - sqs:*
      Resource:
      - !GetAtt myqueue.Arn
    - Effect: Deny
      Action:
      - sqs:*
      NotResource:
      - !GetAtt myqueue.Arn
- PolicyName: giveaccesstotopiconly
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
    - Effect: Allow
      Action:
      - sns:*
      Resource:
      - !Ref mytopic
    - Effect: Deny
      Action:
      - sns:*
      NotResource:
      - !Ref mytopic
```

Declarar um recurso de chave de acesso do IAM

Este trecho mostra um recurso [AWS::IAM::AccessKey](#). O recurso myaccesskey cria uma chave de acesso e a atribui para um usuário do IAM que é declarado como um recurso [AWS::IAM::User](#) no modelo.

JSON

```
"myaccesskey" : {
```



```
"Type" : "AWS::IAM::AccessKey",
"Properties" : {
  "UserName" : { "Ref" : "myuser" }
}
```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

É possível obter a chave secreta para um recurso `AWS::IAM::AccessKey` usando a função `Fn::GetAtt`. Uma forma de recuperar a chave secreta é colocá-la em um valor `Output`. Você pode obter a chave de acesso usando a função `Ref`. As seguintes declarações do valor `Output` obtêm a chave de acesso e a chave secreta para `myaccesskey`.

JSON

```
"AccessKeyformyaccesskey" : {
  "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
  "Value" : {
    "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
  }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

Você também pode passar a chave de acesso e a chave secreta da AWS para uma instância do Amazon EC2 ou um grupo do Auto Scaling definido no modelo. A declaração [AWS::EC2::Instance](#)

a seguir usa a propriedade `UserData` para passar a chave de acesso e a chave secreta para o recurso `myaccesskey`.

JSON

```
"myinstance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "AvailabilityZone" : "us-east-1a",
    "ImageId" : "ami-0ff8a91507f77f867",
    "UserData" : {
      "Fn::Base64" : {
        "Fn::Join" : [
          "", [
            "ACCESS_KEY=", {
              "Ref" : "myaccesskey"
            },
            "&",
            "SECRET_KEY=",
            {
              "Fn::GetAtt" : [
                "myaccesskey",
                "SecretAccessKey"
              ]
            }
          ]
        ]
      }
    }
  }
}
```

YAML

```
myinstance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: "us-east-1a"
    ImageId: ami-0ff8a91507f77f867
    UserData:
      Fn::Base64: !Sub "ACCESS_KEY=${myaccesskey}&SECRET_KEY=
${myaccesskey.SecretAccessKey}"
```

Declarar um recurso de grupo do IAM

Este trecho mostra um recurso [AWS::IAM::Group](#). O grupo tem um caminho ("/myapplication/"). O documento de política chamado myapppolicy é adicionado ao grupo para permitir que os usuários do grupo executem todas as ações do Amazon SQS no recurso de fila myqueue do Amazon SQS e negar todos os outros recursos do Amazon SQS, exceto myqueue.

Para atribuir uma política a um recurso, o IAM exige o Nome de recurso da Amazon (ARN). No trecho, a função Fn::GetAtt obtém o ARN da fila do recurso [AWS::SQS::Queue](#).

JSON

```
"mygroup" : {
  "Type" : "AWS::IAM::Group",
  "Properties" : {
    "Path" : "/myapplication/",
    "Policies" : [ {
      "PolicyName" : "myapppolicy",
      "PolicyDocument" : {
        "Version": "2012-10-17",
        "Statement" : [ {
          "Effect" : "Allow",
          "Action" : [ "sqs:*" ],
          "Resource" : [ {
            "Fn::GetAtt" : [ "myqueue", "Arn" ]
          } ]
        },
        {
          "Effect" : "Deny",
          "Action" : [ "sqs:*" ],
          "NotResource" : [ { "Fn::GetAtt" : [ "myqueue", "Arn" ] } ]
        }
      ]
    } ]
  }
}
```

YAML

```
mygroup:
  Type: AWS::IAM::Group
  Properties:
```

```
Path: "/myapplication/"
Policies:
- PolicyName: myapppolicy
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
    - Effect: Allow
      Action:
      - sqs:*
      Resource: !GetAtt myqueue.Arn
    - Effect: Deny
      Action:
      - sqs:*
      NotResource: !GetAtt myqueue.Arn
```

Adicionar usuários a um grupo

O recurso [AWS::IAM::UserToGroupAddition](#) adiciona os usuários a um grupo. No trecho a seguir, o recurso `addUserToGroup` adiciona os usuários a seguir a um grupo existente chamado `myexistinggroup2`: o usuário existente `existinguser1` e o usuário `myuser`, que é declarado como um recurso [AWS::IAM::User](#) no modelo.

JSON

```
"addUserToGroup" : {
  "Type" : "AWS::IAM::UserToGroupAddition",
  "Properties" : {
    "GroupName" : "myexistinggroup2",
    "Users" : [ "existinguser1", { "Ref" : "myuser" } ]
  }
}
```

YAML

```
addUserToGroup:
  Type: AWS::IAM::UserToGroupAddition
  Properties:
    GroupName: myexistinggroup2
    Users:
      - existinguser1
      - !Ref myuser
```

Declarar uma política do IAM

Este trecho mostra como criar uma política e aplicá-la a vários grupos usando um recurso [AWS::IAM::Policy](#) chamado `mypolicy`. O recurso `mypolicy` contém uma propriedade `PolicyDocument` que permite as ações `GetObject`, `PutObject` e `PutObjectAcl` sobre os objetos no bucket do S3 representado pelo ARN `arn:aws:s3:::myAWSBucket`. O recurso `mypolicy` aplica a política para um grupo existente chamado `myexistinggroup1` e a um grupo `mygroup` que é declarado no modelo como um recurso [AWS::IAM::Group](#). Este exemplo mostra como aplicar uma política a um grupo usando a propriedade `Groups`; no entanto, você também pode usar a propriedade `Users` para adicionar um documento de política a uma lista de usuários.

JSON

```
"mypolicy" : {
  "Type" : "AWS::IAM::Policy",
  "Properties" : {
    "PolicyName" : "mygrouppolicy",
    "PolicyDocument" : {
      "Version": "2012-10-17",
      "Statement" : [ {
        "Effect" : "Allow",
        "Action" : [
          "s3:GetObject" , "s3:PutObject" , "s3:PutObjectAcl" ],
        "Resource" : "arn:aws:s3:::myAWSBucket/*"
      } ]
    },
    "Groups" : [ "myexistinggroup1", { "Ref" : "mygroup" } ]
  }
}
```

YAML

```
mypolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: mygrouppolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
```

```

- s3:GetObject
- s3:PutObject
- s3:PutObjectAcl
Resource: arn:aws:s3:::myAWSBucket/*
Groups:
- myexistinggroup1
- !Ref mygroup

```

Declarar uma política do bucket do Amazon S3

Este trecho mostra como criar uma política e aplicá-la a um bucket do Amazon S3 usando o recurso [AWS::S3::BucketPolicy](#). O recurso `mybucketpolicy` declara um documento de política que permite que o usuário `user1` do IAM execute a ação `GetObject` em todos os objetos no bucket do S3 aos quais essa política se aplica. No trecho, a função `Fn::GetAtt` obtém o Nome de região da Amazon (ARN) do recurso `user1`. O recurso `mybucketpolicy` aplica a política ao `mybucket` do recurso `AWS::S3::BucketPolicy`. A função `Ref` obtém o nome do bucket do recurso `mybucket`.

JSON

```

"mybucketpolicy" : {
  "Type" : "AWS::S3::BucketPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "ReadAccess",
        "Action" : [ "s3:GetObject" ],
        "Effect" : "Allow",
        "Resource" : { "Fn::Join" : [
          "", [ "arn:aws:s3:::", { "Ref" : "mybucket" } , "/" ]
        ] },
        "Principal" : {
          "AWS" : { "Fn::GetAtt" : [ "user1", "Arn" ] }
        }
      ]
    },
    "Bucket" : { "Ref" : "mybucket" }
  }
}

```

YAML

```
mybucketpolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    PolicyDocument:
      Id: MyPolicy
      Version: '2012-10-17'
      Statement:
        - Sid: ReadAccess
          Action:
            - s3:GetObject
          Effect: Allow
          Resource: !Sub "arn:aws:s3:::${mybucket}/*"
          Principal:
            AWS: !GetAtt user1.Arn
      Bucket: !Ref mybucket
```

Declarar uma política de tópico do Amazon SNS

Este trecho mostra como criar uma política e aplicá-la a um tópico do Amazon SNS usando o recurso [AWS::SNS::TopicPolicy](#). O recurso `mysnspolicy` contém uma propriedade `PolicyDocument` que permite que o recurso [AWS::IAM::User](#) denominado `myuser` execute a ação `Publish` em um recurso [AWS::SNS::Topic](#) denominado `mytopic`. No trecho, a função `Fn::GetAtt` obtém o Nome de região da Amazon (ARN) para o recurso `myuser` e a função `Ref` obtém o Nome de região da Amazon (ARN) para o recurso `mytopic`.

JSON

```
"mysnspolicy" : {
  "Type" : "AWS::SNS::TopicPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyTopicPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "My-statement-id",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : { "Fn::GetAtt" : [ "myuser", "Arn" ] }
        },
        "Action" : "sns:Publish",
```

```

        "Resource" : "*"
      } ]
    },
    "Topics" : [ { "Ref" : "mytopic" } ]
  }
}

```

YAML

```

mysnspolicy:
  Type: AWS::SNS::TopicPolicy
  Properties:
    PolicyDocument:
      Id: MyTopicPolicy
      Version: '2012-10-17'
      Statement:
        - Sid: My-statement-id
          Effect: Allow
          Principal:
            AWS: !GetAtt myuser.Arn
          Action: sns:Publish
          Resource: "*"
    Topics:
      - !Ref mytopic

```

Declarar uma política do Amazon SQS

Este trecho mostra como criar uma política e aplicá-la a uma fila do Amazon SQS usando o recurso [AWS::SQS::QueuePolicy](#). A propriedade `PolicyDocument` permite que o usuário `myapp` existente (especificado pelo seu ARN) execute a ação `SendMessage` em uma fila existente, que é especificada pelo seu URL, e um recurso [AWS::SQS::Queue](#) denominado `myqueue`. A função [Ref](#) obtém o URL para o recurso `myqueue`.

JSON

```

"mysqspolicy" : {
  "Type" : "AWS::SQS::QueuePolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyQueuePolicy",
      "Version": "2012-10-17",
      "Statement" : [ {

```



```

        "Sid" : "Allow-User-SendMessage",
        "Effect" : "Allow",
        "Principal" : {
            "AWS" : "arn:aws:iam::123456789012:user/myapp"
        },
        "Action" : [ "sqs:SendMessage" ],
        "Resource" : "*"
    } ]
},
"Queues" : [
    "https://sqs.us-east-2aws-region.amazonaws.com/123456789012/myexistingqueue",
    { "Ref" : "myqueue" }
]
}
}

```

YAML

```

mysqspolicy:
  Type: AWS::SQS::QueuePolicy
  Properties:
    PolicyDocument:
      Id: MyQueuePolicy
      Version: '2012-10-17'
      Statement:
        - Sid: Allow-User-SendMessage
          Effect: Allow
          Principal:
            AWS: arn:aws:iam::123456789012:user/myapp
          Action:
            - sqs:SendMessage
          Resource: "*"
    Queues:
      - https://sqs.aws-region.amazonaws.com/123456789012/myexistingqueue
      - !Ref myqueue

```

Exemplos de modelos de função do IAM

Esta seção fornece exemplos de modelos do CloudFormation para perfis do IAM para instâncias do EC2.

Para obter mais informações, consulte [Perfis do IAM para o Amazon EC2](#) no Guia do usuário do Amazon EC2.

Função do IAM com o EC2

Neste exemplo, o perfil de instância é referenciado pela propriedade `IamInstanceProfile` da instância do EC2. Tanto a instância quanto a função de política referenciam a política [AWS::IAM::Role](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "myEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Version": "2009-05-15",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "m1.small",
        "Monitoring": "true",
        "DisableApiTermination": "false",
        "IamInstanceProfile": {
          "Ref": "RootInstanceProfile"
        }
      }
    },
    "RootRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [ {
            "Effect": "Allow",
            "Principal": {
              "Service": [ "ec2.amazonaws.com" ]
            },
            "Action": [ "sts:AssumeRole" ]
          } ]
        }
      }
    },
    "RolePolicies": {
      "Type": "AWS::IAM::Policy",
      "Properties": {
```

```

        "PolicyName": "root",
        "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [ {
                "Effect": "Allow",
                "Action": "*",
                "Resource": "*"
            } ]
        },
        "Roles": [ { "Ref": "RootRole" } ]
    }
},
"RootInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [ { "Ref": "RootRole" } ]
    }
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      Monitoring: 'true'
      DisableApiTermination: 'false'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow

```

```
Principal:
  Service:
    - ec2.amazonaws.com
  Action:
    - sts:AssumeRole
  Path: "/"
RolePolicies:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: root
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action: "*"
          Resource: "*"
    Roles:
      - !Ref RootRole
RootInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: "/"
    Roles:
      - !Ref RootRole
```

Perfil do IAM com grupo do Auto Scaling

Neste exemplo, o perfil de instância é referenciado pela propriedade `IamInstanceProfile` de um grupo de uma configuração de execução do Amazon EC2 Auto Scaling.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "myLCOne": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Version": "2009-05-15",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "m1.small",
        "InstanceMonitoring": "true",
        "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
      }
    }
  }
}
```

```

    }
  },
  "myASGrpOne": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Version": "2009-05-15",
    "Properties": {
      "AvailabilityZones": [ "us-east-1a" ],
      "LaunchConfigurationName": { "Ref": "myLCOne" },
      "MinSize": "0",
      "MaxSize": "0",
      "HealthCheckType": "EC2",
      "HealthCheckGracePeriod": "120"
    }
  },
  "RootRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [ {
          "Effect": "Allow",
          "Principal": {
            "Service": [ "ec2.amazonaws.com" ]
          },
          "Action": [ "sts:AssumeRole" ]
        } ]
      },
      "Path": "/"
    }
  },
  "RolePolicies": {
    "Type": "AWS::IAM::Policy",
    "Properties": {
      "PolicyName": "root",
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [ {
          "Effect": "Allow",
          "Action": "*",
          "Resource": "*"
        } ]
      },
      "Roles": [ { "Ref": "RootRole" } ]
    }
  }
}

```

```

    },
    "RootInstanceProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Properties": {
        "Path": "/",
        "Roles": [ { "Ref": "RootRole" } ]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myLCOne:
    Type: AWS::AutoScaling::LaunchConfiguration
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      InstanceMonitoring: 'true'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  myASGrpOne:
    Type: AWS::AutoScaling::AutoScalingGroup
    Version: '2009-05-15'
    Properties:
      AvailabilityZones:
        - "us-east-1a"
      LaunchConfigurationName:
        !Ref myLCOne
      MinSize: '0'
      MaxSize: '0'
      HealthCheckType: EC2
      HealthCheckGracePeriod: '120'
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow

```

```
Principal:
  Service:
    - ec2.amazonaws.com
  Action:
    - sts:AssumeRole
  Path: "/"
RolePolicies:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: root
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action: "*"
          Resource: "*"
    Roles:
      - !Ref RootRole
RootInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: "/"
    Roles:
      - !Ref RootRole
```

AWS LambdaModelo

O modelo a seguir usa uma função do AWS Lambda (Lambda) e um recurso personalizado para anexar um novo grupo de segurança a uma lista de grupos de segurança existentes. Essa função é útil quando você deseja criar uma lista de grupos de segurança dinamicamente, de maneira que a lista inclua grupos de segurança novos e existentes. Por exemplo, é possível passar uma lista de grupos de segurança existentes como um valor de parâmetro, anexar o novo valor à lista e associar todos os valores a uma instância EC2. Para obter mais informações sobre o tipo de recurso de função do Lambda, consulte [AWS::Lambda::Function](#).

No exemplo, quando o CloudFormation cria o recurso personalizado AllSecurityGroups, ele invoca a função do Lambda AppendItemToListFunction. O CloudFormation passa a lista de grupos de segurança existentes e um novo grupo de segurança (NewSecurityGroup) para a função, que anexa o novo grupo de segurança à lista e retorna a lista modificada. O CloudFormation usa a lista modificada para associar todos os grupos de segurança ao recurso MyEC2Instance.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ExistingSecurityGroups": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>"
    },
    "ExistingVPC": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "The VPC ID that includes the security groups in the ExistingSecurityGroups parameter."
    },
    "InstanceType": {
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": [
        "t2.micro",
        "t3.micro"
      ]
    }
  },
  "Resources": {
    "SecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Allow HTTP traffic to the host",
        "VpcId": {
          "Ref": "ExistingVPC"
        },
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIp": "0.0.0.0/0"
          }
        ],
        "SecurityGroupEgress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
```



```

        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
    }
  ]
},
"AllSecurityGroups": {
  "Type": "Custom::Split",
  "Properties": {
    "ServiceToken": {
      "Fn::GetAtt": [
        "AppendItemToListFunction",
        "Arn"
      ]
    },
    "List": {
      "Ref": "ExistingSecurityGroups"
    },
    "AppendedItem": {
      "Ref": "SecurityGroup"
    }
  }
},
"AppendItemToListFunction": {
  "Type": "AWS::Lambda::Function",
  "Properties": {
    "Handler": "index.handler",
    "Role": {
      "Fn::GetAtt": [
        "LambdaExecutionRole",
        "Arn"
      ]
    },
    "Code": {
      "ZipFile": {
        "Fn::Join": [
          "",
          [
            "var response = require('cfn-response');",
            "exports.handler = function(event, context) {",
            "  var responseData = {Value:",
            "event.ResourceProperties.List};",
            "  ",
            "responseData.Value.push(event.ResourceProperties.AppendedItem);",

```

```

        "    response.send(event, context, response.SUCCESS,
responseData);",
        "};"
    ]
}
},
"Runtime": "nodejs20.x"
}
},
"MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "SecurityGroupIds": {
            "Fn::GetAtt": [
                "AllSecurityGroups",
                "Value"
            ]
        },
        "InstanceType": {
            "Ref": "InstanceType"
        }
    }
},
"LambdaExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lambda.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        }
    }
}
]

```

```

    },
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "root",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "logs:*"
                        ],
                        "Resource": "arn:aws:logs:*:*:*"
                    }
                ]
            }
        }
    ]
},
"Outputs": {
    "AllSecurityGroups": {
        "Description": "Security Groups that are associated with the EC2 instance",
        "Value": {
            "Fn::Join": [
                ", ",
                [
                    {
                        "Fn::GetAtt": [
                            "AllSecurityGroups",
                            "Value"
                        ]
                    }
                ]
            ]
        }
    }
}
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

Parameters:**ExistingSecurityGroups:**

Type: List<AWS::EC2::SecurityGroup::Id>

ExistingVPC:

Type: AWS::EC2::VPC::Id

Description: The VPC ID that includes the security groups in the

ExistingSecurityGroups parameter.

InstanceType:

Type: String

Default: t2.micro

AllowedValues:

- t2.micro
- t3.micro

Resources:**SecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP traffic to the host

VpcId: !Ref ExistingVPC

SecurityGroupIngress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- CidrIp: 0.0.0.0/0

SecurityGroupEgress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- CidrIp: 0.0.0.0/0

AllSecurityGroups:

Type: Custom::Split

Properties:

ServiceToken: !GetAtt AppendItemToListFunction.Arn

List: !Ref ExistingSecurityGroups

AppendedItem: !Ref SecurityGroup

AppendItemToListFunction:

Type: AWS::Lambda::Function

Properties:

Handler: index.handler

Role: !GetAtt LambdaExecutionRole.Arn

Code:

ZipFile: !Join

- ''
- - var response = require('cfn-response');

```

- exports.handler = function(event, context) {
- '   var responseData = {Value: event.ResourceProperties.List};'
- '   responseData.Value.push(event.ResourceProperties.AppendedItem);'
- '   response.send(event, context, response.SUCCESS, responseData);'
- '};'

```

Runtime: nodejs20.x

MyEC2Instance:

Type: AWS::EC2::Instance

Properties:

ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}'

SecurityGroupIds: !GetAtt AllSecurityGroups.Value

InstanceType: !Ref InstanceType

LambdaExecutionRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Principal:

Service:

- lambda.amazonaws.com

Action:

- sts:AssumeRole

Path: /

Policies:

- PolicyName: root

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- logs:*

Resource: arn:aws:logs:*:*:*

Outputs:

AllSecurityGroups:

Description: Security Groups that are associated with the EC2 instance

Value: !Join

- ', '

- !GetAtt AllSecurityGroups.Value

Trechos de modelo do Amazon Redshift

O Amazon Redshift é um serviço de data warehouse em escala de petabytes totalmente gerenciado na nuvem. Você pode usar o CloudFormation para provisionar e gerenciar clusters do Amazon Redshift.

Cluster do Amazon Redshift

O modelo de amostra a seguir cria um cluster do Amazon Redshift de acordo com os valores de parâmetro que são especificadas quando a pilha é criada. O parameter group de cluster associado ao cluster do Amazon Redshift permite o registro de atividades do usuário. O modelo também executa o cluster do Amazon Redshift em uma Amazon VPC, que é definida no modelo. A VPC inclui um gateway de Internet que permite acessar os clusters do Amazon Redshift da Internet. No entanto, a comunicação entre o cluster e o gateway de Internet também deve ser ativada, o que é feito pela entrada na tabela de roteamento.

Note

O modelo inclui a condição `IsMultiNodeCluster` para que o parâmetro `NumberOfNodes` seja declarado somente quando o valor do parâmetro `ClusterType` estiver definido como `multi-node`.

O exemplo define os parâmetros `MysqlRootPassword` com a propriedade `NoEcho` definida como `true`. Se você definir o atributo `NoEcho` como `true`, o CloudFormation retornará o valor do parâmetro mascarado como asteriscos (`*****`) para qualquer chamada que descreva a pilha ou os eventos de pilha, exceto informações armazenadas nos locais especificados abaixo.

Important

O uso do atributo `NoEcho` não mascara informações armazenadas no seguinte:

- A seção de modelo de `Metadata`. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção `Metadata`. Para obter mais informações, consulte [Metadados](#).
- A seção de modelo de `Outputs`. Para obter mais informações, consulte [Saídas](#).
- O atributo `Metadata` de uma definição de recurso. Para obter mais informações, consulte o atributo [Metadata](#).

É altamente recomendável não usar esses mecanismos para incluir informações confidenciais, como senhas ou segredos.

Important

Em vez de incorporar informações confidenciais diretamente em modelos do CloudFormation, recomendamos usar os parâmetros dinâmicos no modelo da pilha para fazer referência a informações confidenciais que são armazenadas e gerenciadas de forma externa ao CloudFormation, como no AWS Systems Manager Parameter Store ou no AWS Secrets Manager.

Para obter mais informações, consulte a prática recomendada [Não incorporar credenciais nos seus modelos](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the cluster is
created",
      "Type" : "String",
      "Default" : "dev",
      "AllowedPattern" : "([a-z]|[0-9])+"
    },
    "ClusterType" : {
      "Description" : "The type of cluster",
      "Type" : "String",
      "Default" : "single-node",
      "AllowedValues" : [ "single-node", "multi-node" ]
    },
    "NumberOfNodes" : {
      "Description" : "The number of compute nodes in the cluster. For multi-node
clusters, the NumberOfNodes parameter must be greater than 1",
      "Type" : "Number",
      "Default" : "1"
    },
  },
}
```

```
"NodeType" : {
    "Description" : "The type of node to be provisioned",
    "Type" : "String",
    "Default" : "ds2.xlarge",
    "AllowedValues" : [ "ds2.xlarge", "ds2.8xlarge", "dc1.large", "dc1.8xlarge" ]
},
"MasterUsername" : {
    "Description" : "The user name that is associated with the master user account
for the cluster that is being created",
    "Type" : "String",
    "Default" : "defaultuser",
    "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
},
"MasterUserPassword" : {
    "Description" : "The password that is associated with the master user account for
the cluster that is being created.",
    "Type" : "String",
    "NoEcho" : "true"
},
"InboundTraffic" : {
    "Description" : "Allow inbound traffic to the cluster from this CIDR range.",
    "Type" : "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default" : "0.0.0.0/0",
    "AllowedPattern" : "(\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})/(\\
\\d{1,2})),
    "ConstraintDescription" : "must be a valid CIDR range of the form x.x.x.x/x."
},
"PortNumber" : {
    "Description" : "The port number on which the cluster accepts incoming
connections.",
    "Type" : "Number",
    "Default" : "5439"
}
},
"Conditions" : {
    "IsMultiNodeCluster" : {
        "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
    }
},
"Resources" : {
    "RedshiftCluster" : {
        "Type" : "AWS::Redshift::Cluster",
```



```

    "DependsOn" : "AttachGateway",
    "Properties" : {
      "ClusterType" : { "Ref" : "ClusterType" },
      "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster", { "Ref" :
"NumberOfNodes" }, { "Ref" : "AWS::NoValue" } ]}],
      "NodeType" : { "Ref" : "NodeType" },
      "DBName" : { "Ref" : "DatabaseName" },
      "MasterUsername" : { "Ref" : "MasterUsername" },
      "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
      "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup" },
      "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
      "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
      "PubliclyAccessible" : "true",
      "Port" : { "Ref" : "PortNumber" }
    }
  },
  "RedshiftClusterParameterGroup" : {
    "Type" : "AWS::Redshift::ClusterParameterGroup",
    "Properties" : {
      "Description" : "Cluster parameter group",
      "ParameterGroupFamily" : "redshift-1.0",
      "Parameters" : [{
        "ParameterName" : "enable_user_activity_logging",
        "ParameterValue" : "true"
      }]
    }
  },
  "RedshiftClusterSubnetGroup" : {
    "Type" : "AWS::Redshift::ClusterSubnetGroup",
    "Properties" : {
      "Description" : "Cluster subnet group",
      "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
    }
  },
  "VPC" : {
    "Type" : "AWS::EC2::VPC",
    "Properties" : {
      "CidrBlock" : "10.0.0.0/16"
    }
  },
  "PublicSubnet" : {
    "Type" : "AWS::EC2::Subnet",
    "Properties" : {
      "CidrBlock" : "10.0.0.0/24",

```

```
    "VpcId" : { "Ref" : "VPC" }
  },
  "SecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Security group",
      "SecurityGroupIngress" : [ {
        "CidrIp" : { "Ref" : "InboundTraffic" },
        "FromPort" : { "Ref" : "PortNumber" },
        "ToPort" : { "Ref" : "PortNumber" },
        "IpProtocol" : "tcp"
      } ],
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "myInternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway"
  },
  "AttachGateway" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "InternetGatewayId" : { "Ref" : "myInternetGateway" }
    }
  },
  "PublicRouteTable" : {
    "Type" : "AWS::EC2::RouteTable",
    "Properties" : {
      "VpcId" : {
        "Ref" : "VPC"
      }
    }
  },
  "PublicRoute" : {
    "Type" : "AWS::EC2::Route",
    "DependsOn" : "AttachGateway",
    "Properties" : {
      "RouteTableId" : {
        "Ref" : "PublicRouteTable"
      },
      "DestinationCidrBlock" : "0.0.0.0/0",
      "GatewayId" : {
        "Ref" : "myInternetGateway"
      }
    }
  }
}
```

```

    }
  }
},
"PublicSubnetRouteTableAssociation" : {
  "Type" : "AWS::EC2::SubnetRouteTableAssociation",
  "Properties" : {
    "SubnetId" : {
      "Ref" : "PublicSubnet"
    },
    "RouteTableId" : {
      "Ref" : "PublicRouteTable"
    }
  }
}
},
"Outputs" : {
  "ClusterEndpoint" : {
    "Description" : "Cluster endpoint",
    "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Port" ] } ] ] }
  },
  "ClusterName" : {
    "Description" : "Name of cluster",
    "Value" : { "Ref" : "RedshiftCluster" }
  },
  "ParameterGroupName" : {
    "Description" : "Name of parameter group",
    "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
  },
  "RedshiftClusterSubnetGroupName" : {
    "Description" : "Name of cluster subnet group",
    "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
  },
  "RedshiftClusterSecurityGroupName" : {
    "Description" : "Name of cluster security group",
    "Value" : { "Ref" : "SecurityGroup" }
  }
}
}
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  DatabaseName:
    Description: The name of the first database to be created when the cluster is
      created
    Type: String
    Default: dev
    AllowedPattern: "([a-z]|[0-9])+"
  ClusterType:
    Description: The type of cluster
    Type: String
    Default: single-node
    AllowedValues:
      - single-node
      - multi-node
  NumberOfNodes:
    Description: The number of compute nodes in the cluster. For multi-node clusters,
      the NumberOfNodes parameter must be greater than 1
    Type: Number
    Default: '1'
  NodeType:
    Description: The type of node to be provisioned
    Type: String
    Default: ds2.xlarge
    AllowedValues:
      - ds2.xlarge
      - ds2.8xlarge
      - dc1.large
      - dc1.8xlarge
  MasterUsername:
    Description: The user name that is associated with the master user account for
      the cluster that is being created
    Type: String
    Default: defaultuser
    AllowedPattern: "([a-z])([a-z]|[0-9])*"
  MasterUserPassword:
    Description: The password that is associated with the master user account for
      the cluster that is being created.
    Type: String
    NoEcho: 'true'
  InboundTraffic:
    Description: Allow inbound traffic to the cluster from this CIDR range.
```

```
Type: String
MinLength: '9'
MaxLength: '18'
Default: 0.0.0.0/0
AllowedPattern: "(\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}/(\d{1,2})"
ConstraintDescription: must be a valid CIDR range of the form x.x.x.x/x.
PortNumber:
  Description: The port number on which the cluster accepts incoming connections.
  Type: Number
  Default: '5439'
Conditions:
  IsMultiNodeCluster:
    Fn::Equals:
      - Ref: ClusterType
      - multi-node
Resources:
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    DependsOn: AttachGateway
    Properties:
      ClusterType:
        Ref: ClusterType
      NumberOfNodes:
        Fn::If:
          - IsMultiNodeCluster
          - Ref: NumberOfNodes
          - Ref: AWS::NoValue
      NodeType:
        Ref: NodeType
      DBName:
        Ref: DatabaseName
      MasterUsername:
        Ref: MasterUsername
      MasterUserPassword:
        Ref: MasterUserPassword
      ClusterParameterGroupName:
        Ref: RedshiftClusterParameterGroup
      VpcSecurityGroupIds:
        - Ref: SecurityGroup
      ClusterSubnetGroupName:
        Ref: RedshiftClusterSubnetGroup
      PubliclyAccessible: 'true'
      Port:
        Ref: PortNumber
```

```
RedshiftClusterParameterGroup:
  Type: AWS::Redshift::ClusterParameterGroup
  Properties:
    Description: Cluster parameter group
    ParameterGroupFamily: redshift-1.0
    Parameters:
      - ParameterName: enable_user_activity_logging
        ParameterValue: 'true'
RedshiftClusterSubnetGroup:
  Type: AWS::Redshift::ClusterSubnetGroup
  Properties:
    Description: Cluster subnet group
    SubnetIds:
      - Ref: PublicSubnet
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
PublicSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock: 10.0.0.0/24
    VpcId:
      Ref: VPC
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group
    SecurityGroupIngress:
      - CidrIp:
          Ref: InboundTraffic
        FromPort:
          Ref: PortNumber
        ToPort:
          Ref: PortNumber
        IpProtocol: tcp
    VpcId:
      Ref: VPC
myInternetGateway:
  Type: AWS::EC2::InternetGateway
AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId:
```

```
    Ref: VPC
  InternetGatewayId:
    Ref: myInternetGateway
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: VPC
PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId:
      Ref: PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId:
      Ref: myInternetGateway
PublicSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId:
      Ref: PublicSubnet
    RouteTableId:
      Ref: PublicRouteTable
Outputs:
  ClusterEndpoint:
    Description: Cluster endpoint
    Value: !Sub "${RedshiftCluster.Endpoint.Address}:${RedshiftCluster.Endpoint.Port}"
  ClusterName:
    Description: Name of cluster
    Value:
      Ref: RedshiftCluster
  ParameterGroupName:
    Description: Name of parameter group
    Value:
      Ref: RedshiftClusterParameterGroup
  RedshiftClusterSubnetGroupName:
    Description: Name of cluster subnet group
    Value:
      Ref: RedshiftClusterSubnetGroup
  RedshiftClusterSecurityGroupName:
    Description: Name of cluster security group
    Value:
```

Ref: SecurityGroup

Consulte também

[AWS::Redshift::Cluster](#)

Trechos de modelo do Amazon RDS

Tópicos

- [Recurso de instância de banco de dados do Amazon RDS](#)
- [Recurso de instância de banco de dados do Oracle do Amazon RDS](#)
- [Recurso DBSecurityGroup do Amazon RDS para intervalo de CIDR](#)
- [DBSecurityGroup do Amazon RDS com um grupo de segurança do Amazon EC2](#)
- [Security groups da VPC múltiplos](#)
- [Instância de banco de dados do Amazon RDS em um grupo de segurança da VPC](#)

Recurso de instância de banco de dados do Amazon RDS

Este exemplo mostra um recurso de instância de banco de dados do Amazon RDS especificando uma senha de usuário mestre gerenciada. Para obter mais informações, consulte [Gerenciamento de senhas com o AWS Secrets Manager](#) no Guia do usuário do Amazon RDS e [Gerenciamento de senhas com o AWS Secrets Manager](#) no Guia do usuário do Aurora. Como a propriedade `EngineVersion` opcional não é especificada, a versão do mecanismo padrão é usada para essa instância de banco de dados. Para obter detalhes sobre a versão do mecanismo padrão e outras configurações padrão, consulte [CreateDBInstance](#). A propriedade `DBSecurityGroups` autoriza a entrada na rede para os recursos `AWS::RDS::DBSecurityGroup` chamados `MyDbSecurityByEC2SecurityGroup` e `MyDbSecurityByCIDRIPGroup`. Para obter detalhes, consulte [AWS::RDS::DBInstance](#). O recurso de instância de banco de dados também tem um atributo `DeletionPolicy` definido como `Snapshot`. Com a `DeletionPolicy` do `Snapshot` definida, o CloudFormation capturará um snapshot dessa instância de banco de dados antes de excluí-lo durante a exclusão da pilha.

JSON

```
"MyDB" : {  
  "Type" : "AWS::RDS::DBInstance",  
  "Properties" : {
```



```

    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "MasterUsername" : "MyName",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}

```

YAML

```

MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    MasterUsername: MyName
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
    DeletionPolicy: Snapshot

```

Recurso de instância de banco de dados do Oracle do Amazon RDS

Este exemplo cria um recurso de instância Oracle Database DB com uma senha de usuário mestre gerenciada. Para obter mais informações, consulte [Gerenciamento de senhas com o AWS Secrets Manager](#) no Guia do usuário do Amazon RDS. O exemplo especifica o Engine como oracle-ee com um modelo de traga sua própria licença. Para obter detalhes sobre as configurações para instâncias de banco de dados do Oracle Database, consulte [CreateDBInstance](#). A propriedade DBSecurityGroups autoriza o ingresso na rede dos recursos AWS::RDS::DBSecurityGroup chamados MyDbSecurityByEC2SecurityGroup e MyDbSecurityByCIDRIPGroup. Para obter detalhes, consulte [AWS::RDS::DBInstance](#). O recurso de instância de banco de dados também tem um

atributo `DeletionPolicy` definido como `Snapshot`. Com a `DeletionPolicy` do `Snapshot` definida, o CloudFormation capturará um snapshot dessa instância de banco de dados antes de excluí-lo durante a exclusão da pilha.

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "oracle-ee",
    "LicenseModel" : "bring-your-own-license",
    "MasterUsername" : "master",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: oracle-ee
    LicenseModel: bring-your-own-license
    MasterUsername: master
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
    DeletionPolicy: Snapshot
```

Recurso DBSecurityGroup do Amazon RDS para intervalo de CIDR

Este exemplo mostra um recurso DBSecurityGroup do Amazon RDS com uma autorização de entrada para o intervalo de CIDR especificado no formato ddd.ddd.ddd.ddd/dd. Para obter detalhes, consulte [AWS::RDS::DBSecurityGroup](#) e [Ingress](#).

JSON

```
"MyDbSecurityByCIDRIPGroup" : {
  "Type" : "AWS::RDS::DBSecurityGroup",
  "Properties" : {
    "GroupDescription" : "Ingress for CIDRIP",
    "DBSecurityGroupIngress" : {
      "CIDRIP" : "192.168.0.0/32"
    }
  }
}
```

YAML

```
MyDbSecurityByCIDRIPGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    GroupDescription: Ingress for CIDRIP
    DBSecurityGroupIngress:
      CIDRIP: "192.168.0.0/32"
```

DBSecurityGroup do Amazon RDS com um grupo de segurança do Amazon EC2

Este exemplo mostra um recurso [AWS::RDS::DBSecurityGroup](#) com autorização de entrada de um grupo de segurança do Amazon EC2 referenciado por MyEc2SecurityGroup.

Para fazer isso, defina um grupo de segurança do EC2 e, depois, use a função intrínseca Ref para fazer referência ao grupo de segurança do EC2 no DBSecurityGroup.

JSON

```
"DBInstance" : {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
    "DBName" : { "Ref" : "DBName" },
```

```

    "Engine"           : "MySQL",
    "MasterUsername"   : { "Ref" : "DBUsername" },
    "DBInstanceClass"  : { "Ref" : "DBClass" },
    "DBSecurityGroups" : [ { "Ref" : "DBSecurityGroup" } ],
    "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
    "MasterUserPassword": { "Ref" : "DBPassword" }
  }
},

"DBSecurityGroup": {
  "Type": "AWS::RDS::DBSecurityGroup",
  "Properties": {
    "DBSecurityGroupIngress": {
      "EC2SecurityGroupName": {
        "Fn::GetAtt": ["WebServerSecurityGroup", "GroupName"]
      }
    },
    "GroupDescription" : "Frontend Access"
  }
},

"WebServerSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
    "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
    "SecurityGroupIngress" : [
      { "IpProtocol" : "tcp", "FromPort" : 80, "ToPort" : 80, "CidrIp" :
"0.0.0.0/0"},
      { "IpProtocol" : "tcp", "FromPort" : 22, "ToPort" : 22, "CidrIp" : "0.0.0.0/0"}
    ]
  }
}
}

```

YAML

Este exemplo é extraído do seguinte exemplo completo: [Drupal_Single_Instance_With_RDS.template](#)

```

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MasterUsername:

```

```

    Ref: DBUsername
DBInstanceClass:
    Ref: DBClass
DBSecurityGroups:
  - Ref: DBSecurityGroup
AllocatedStorage:
    Ref: DBAllocatedStorage
MasterUserPassword:
    Ref: DBPassword
DBSecurityGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    DBSecurityGroupIngress:
      EC2SecurityGroupName:
        Ref: WebServerSecurityGroup
      GroupDescription: Frontend Access
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0

```

Security groups da VPC múltiplos

Este exemplo mostra um recurso [AWS::RDS::DBSecurityGroup](#) com autorização de entrada para vários grupos de segurança de VPC do Amazon EC2 em [AWS::RDS::DBSecurityGroupIngress](#).

JSON

```

{
  "Resources" : {
    "DBInstance" : {
      "Type" : "AWS::RDS::DBInstance",
      "Properties" : {
        "AllocatedStorage" : "5",

```

```

        "DBInstanceClass" : "db.t2.small",
        "DBName" : { "Ref": "MyDBName" },
        "DBSecurityGroups" : [ { "Ref" : "DbSecurityByEC2SecurityGroup" } ],
        "DBSubnetGroupName" : { "Ref" : "MyDBSubnetGroup" },
        "Engine" : "MySQL",
        "MasterUserPassword": { "Ref" : "MyDBPassword" },
        "MasterUsername" : { "Ref" : "MyDBUsername" }
    },
    "DeletionPolicy" : "Snapshot"
},
"DbSecurityByEC2SecurityGroup" : {
    "Type" : "AWS::RDS::DBSecurityGroup",
    "Properties" : {
        "GroupDescription" : "Ingress for Amazon EC2 security group",
        "EC2VpcId" : { "Ref" : "MyVPC" },
        "DBSecurityGroupIngress" : [ {
            "EC2SecurityGroupId" : "sg-b0ff1111",
            "EC2SecurityGroupOwnerId" : "111122223333"
        }, {
            "EC2SecurityGroupId" : "sg-ffd72222",
            "EC2SecurityGroupOwnerId" : "111122223333"
        } ]
    }
}
}
}
}
}

```

YAML

```

Resources:
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      AllocatedStorage: '5'
      DBInstanceClass: db.t2.small
      DBName:
        Ref: MyDBName
      DBSecurityGroups:
        - Ref: DbSecurityByEC2SecurityGroup
      DBSubnetGroupName:
        Ref: MyDBSubnetGroup
      Engine: MySQL
      MasterUserPassword:

```

```

    Ref: MyDBPassword
  MasterUsername:
    Ref: MyDBUsername
  DeletionPolicy: Snapshot
  DbSecurityByEC2SecurityGroup:
    Type: AWS::RDS::DBSecurityGroup
    Properties:
      GroupDescription: Ingress for Amazon EC2 security group
      EC2VpcId:
        Ref: MyVPC
      DBSecurityGroupIngress:
        - EC2SecurityGroupId: sg-b0ff1111
          EC2SecurityGroupOwnerId: '111122223333'
        - EC2SecurityGroupId: sg-ffd72222
          EC2SecurityGroupOwnerId: '111122223333'

```

Instância de banco de dados do Amazon RDS em um grupo de segurança da VPC

Este exemplo mostra uma instância do banco de dados do Amazon RDS associada a um security group da VPC do Amazon EC2.

JSON

```

{
  "DBEC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Open database for access",
      "SecurityGroupIngress": [{
        "IpProtocol": "tcp",
        "FromPort": 3306,
        "ToPort": 3306,
        "SourceSecurityGroupName": { "Ref": "WebServerSecurityGroup" }
      }]
    }
  },
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName": { "Ref": "DBName" },
      "Engine": "MySQL",
      "MultiAZ": { "Ref": "MultiAZDatabase" },
      "MasterUsername": { "Ref": "DBUser" },

```

```

    "DBInstanceClass" : { "Ref" : "DBClass" },
    "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
    "MasterUserPassword": { "Ref" : "DBPassword" },
    "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "DBEC2SecurityGroup", "GroupId" ] } ]
  }
}
}

```

YAML

```

DBEC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Open database for access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 3306
        ToPort: 3306
        SourceSecurityGroupName:
          Ref: WebServerSecurityGroup
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MultiAZ:
      Ref: MultiAZDatabase
    MasterUsername:
      Ref: DBUser
    DBInstanceClass:
      Ref: DBClass
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword
    VPCSecurityGroups:
      - !GetAtt DBEC2SecurityGroup.GroupId

```

Trechos de modelos do Route 53

Tópicos

- [Conjunto de registro de recurso do Amazon Route 53 usando ID ou nome de zona hospedada](#)
- [Usar RecordSetGroup para configurar conjuntos de registros de recurso ponderados](#)
- [Usar RecordSetGroup para configurar um conjunto de registros de recursos de alias](#)
- [Conjunto de registros de recursos de alias para uma distribuição do CloudFront](#)

Conjunto de registro de recurso do Amazon Route 53 usando ID ou nome de zona hospedada

Ao criar um conjunto de registro de recurso do Amazon Route 53, você deve especificar a zona hospedada em que deseja adicioná-lo. O CloudFormation oferece duas maneiras de especificar uma zona hospedada:

- Você pode especificar explicitamente a zona hospedada usando a propriedade `HostedZoneId`.
- Você pode fazer com que o CloudFormation encontre a zona hospedada usando a propriedade `HostedZoneName`. Se você usar a propriedade `HostedZoneName` e houver várias zonas hospedadas com o mesmo nome, o CloudFormation não criará a pilha.

Adição de RecordSet usando `HostedZoneId`

Este exemplo adiciona um conjunto de registro de recurso do Amazon Route 53 que contém um registro SPF para o nome de domínio `mysite.example.com` que usa a propriedade `HostedZoneId` para especificar a zona hospedada.

JSON

```
"myDNSRecord" : {
  "Type" : "AWS::Route53::RecordSet",
  "Properties" :
  {
    "HostedZoneId" : "Z3DG6IL3SJCGPX",
    "Name" : "mysite.example.com.",
    "Type" : "SPF",
    "TTL" : "900",
    "ResourceRecords" : [ "\"v=spf1 ip4:192.168.0.1/16 -all\"" ]
  }
}
```

YAML

```
myDNSRecord:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: Z3DG6IL3SJCGPX
    Name: mysite.example.com.
    Type: SPF
    TTL: '900'
    ResourceRecords:
      - '"v=spf1 ip4:192.168.0.1/16 -all"'
```

Adição de RecordSet usando HostedZoneName

Este exemplo adiciona um conjunto de registros de recurso do Amazon Route 53 para o nome de domínio "mysite.example.com" usando a propriedade HostedZoneName para especificar a zona hospedada.

JSON

```
"myDNSRecord2" : {
  "Type" : "AWS::Route53::RecordSet",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Name" : "mysite.example.com.",
    "Type" : "A",
    "TTL" : "900",
    "ResourceRecords" : [
      "192.168.0.1",
      "192.168.0.2"
    ]
  }
}
```

YAML

```
myDNSRecord2:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneName: example.com.
    Name: mysite.example.com.
```

```
Type: A
TTL: '900'
ResourceRecords:
- 192.168.0.1
- 192.168.0.2
```

Usar RecordSetGroup para configurar conjuntos de registros de recurso ponderados

Este exemplo usa um [AWS::Route53::RecordSetGroup](#) para configurar dois registros CNAME para o "example.com". zona hospedada. A propriedade RecordSets contém os conjuntos de registros CNAME para o nome DNS do "mysite.example.com". Cada conjunto de registros contém um identificador (SetIdentifier) e um peso (Weight). A proporção do tráfego de Internet que é roteada para os recursos é baseada nos seguintes cálculos:

- Frontend One: $140 / (140 + 60) = 140 / 200 = 70\%$
- Frontend Two: $60 / (140 + 60) = 60 / 200 = 30\%$

Para obter mais informações sobre conjuntos de registros de recursos ponderados, consulte [Roteamento ponderado](#) no Guia do desenvolvedor do Amazon Route 53.

JSON

```
"myDNSOne" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Weighted RR for my frontends.",
    "RecordSets" : [
      {
        "Name" : "mysite.example.com.",
        "Type" : "CNAME",
        "TTL" : "900",
        "SetIdentifier" : "Frontend One",
        "Weight" : "140",
        "ResourceRecords" : ["example-ec2.amazonaws.com"]
      },
      {
        "Name" : "mysite.example.com.",
        "Type" : "CNAME",
        "TTL" : "900",
        "SetIdentifier" : "Frontend Two",
```

```

        "Weight" : "60",
        "ResourceRecords" : ["example-ec2-larger.amazonaws.com"]
    }
}
}

```

YAML

```

myDNSOne:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Weighted RR for my frontends.
    RecordSets:
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend One
        Weight: '140'
        ResourceRecords:
          - example-ec2.amazonaws.com
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend Two
        Weight: '60'
        ResourceRecords:
          - example-ec2-larger.amazonaws.com

```

Usar RecordSetGroup para configurar um conjunto de registros de recursos de alias

Os exemplos a seguir usam um [AWS::Route53::RecordSetGroup](#) para configurar um conjunto de registros de recurso de alias chamado `example.com` que roteia o tráfego para um load balancer do ELB Versão 1 (Classic) e um load balancer (aplicativo ou rede) da versão 2. A propriedade [AliasTarget](#) especifica o ID da zona hospedada e o nome do DNS para o LoadBalancer `myELB` usando a função intrínseca `GetAtt`. O `GetAtt` recupera propriedades diferentes do recurso `myELB`, dependendo de você estar roteando o tráfego para um balanceador de carga versão 1 ou versão 2:

- Load balancer da versão 1: `CanonicalHostedZoneNameID` e `DNSName`
- Load balancer da versão 2: `CanonicalHostedZoneID` e `DNSName`

Para obter mais informações sobre conjuntos de registros de recurso de alias, consulte [Escolher entre registros de alias e de não alias](#), no Guia do Desenvolvedor do Route 53.

JSON para load balancer da versão 1

```
"myELB" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
},
"myDNS" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
    "RecordSets" : [
      {
        "Name" : "example.com.",
        "Type" : "A",
        "AliasTarget" : {
          "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
"CanonicalHostedZoneNameID"] },
          "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
        }
      }
    ]
  }
}
```

YAML para load balancer da versão 1

```
myELB:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
```

```

- LoadBalancerPort: '80'
  InstancePort: '80'
  Protocol: HTTP
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
  RecordSets:
    - Name: example.com.
      Type: A
      AliasTarget:
        HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneNameID'
        DNSName: !GetAtt 'myELB.DNSName'

```

JSON para load balancer da versão 2

```

"myELB" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "Subnets" : [
      {"Ref": "SubnetAZ1"},
      {"Ref" : "SubnetAZ2"}
    ]
  }
},
"myDNS" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
    "RecordSets" : [
      {
        "Name" : "example.com.",
        "Type" : "A",
        "AliasTarget" : {
          "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
"CanonicalHostedZoneID"] },
          "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
        }
      }
    ]
  }
}

```

```
}
```

YAML para load balancer da versão 2

```
myELB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets:
      - Ref: SubnetAZ1
      - Ref: SubnetAZ2
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
      - Name: example.com.
        Type: A
        AliasTarget:
          HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneID'
          DNSName: !GetAtt 'myELB.DNSName'
```

Conjunto de registros de recursos de alias para uma distribuição do CloudFront

O exemplo a seguir cria um registro de alias A que aponta um nome de domínio personalizado para uma distribuição do CloudFront existente. Presume-se que `myHostedZoneID` seja uma referência a um recurso `AWS::Route53::HostedZone` atual no mesmo modelo ou um parâmetro. `myCloudFrontDistribution` refere-se a um recurso `AWS::CloudFront::Distribution` dentro do mesmo modelo. O registro de alias usa o ID de zona hospedada padrão do CloudFront (`Z2FDTNDATAQYW2`) e resolve automaticamente o nome de domínio da distribuição usando `Fn::GetAtt`. Essa configuração permite que o tráfego da Web seja roteado do domínio personalizado para a distribuição do CloudFront sem exigir um endereço IP.

Note

Quando você cria conjuntos de registros de recursos de alias, você deve especificar `Z2FDTNDATAQYW2` para a propriedade `HostedZoneId`. Os conjuntos de registros de recursos do CloudFront não podem ser criados em uma zona privada.

JSON

```
{
  "myDNS": {
    "Type": "AWS::Route53::RecordSetGroup",
    "Properties": {
      "HostedZoneId": {
        "Ref": "myHostedZoneID"
      },
      "RecordSets": [
        {
          "Name": {
            "Ref": "myRecordSetDomainName"
          },
          "Type": "A",
          "AliasTarget": {
            "HostedZoneId": "Z2FDTNDATAQYW2",
            "DNSName": {
              "Fn::GetAtt": [
                "myCloudFrontDistribution",
                "DomainName"
              ]
            },
            "EvaluateTargetHealth": false
          }
        }
      ]
    }
  }
}
```

YAML

```
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneId: !Ref myHostedZoneID
    RecordSets:
      - Name: !Ref myRecordSetDomainName
        Type: A
        AliasTarget:
          HostedZoneId: Z2FDTNDATAQYW2
          DNSName: !GetAtt
```



```
- myCloudFrontDistribution
- DomainName
EvaluateTargetHealth: false
```

Trechos de modelo do Amazon S3

Use os modelos de exemplo do Amazon S3 para ajudar a descrever seus buckets do Amazon S3 com o CloudFormation. Para obter mais exemplos, consulte a seção [Exemplos](#) no recurso `AWS::S3::Bucket`.

Tópicos

- [Criar um bucket do Amazon S3 com padrões](#)
- [Criação de um bucket do Amazon S3 para hospedagem de site e com uma `DeletionPolicy`](#)
- [Criar um site estático usando um domínio personalizado](#)

Criar um bucket do Amazon S3 com padrões

Este exemplo usa um `AWS::S3::Bucket` para criar um bucket com as configurações padrão.

JSON

```
"myS3Bucket" : {
  "Type" : "AWS::S3::Bucket"
}
```

YAML

```
MyS3Bucket:
  Type: AWS::S3::Bucket
```

Criação de um bucket do Amazon S3 para hospedagem de site e com uma **DeletionPolicy**

Este exemplo cria um bucket como um site e desabilita o bloqueio de acesso público (são necessárias permissões públicas de leitura para buckets configurados para a hospedagem de sites). Em seguida, uma política pública para buckets é adicionada ao bucket. Como esse recurso de bucket tem um atributo `DeletionPolicy` definido como `Retain`, o CloudFormation não exclui

esse bucket ao excluir a pilha. A seção Output usa `Fn::GetAtt` para recuperar os atributos `WebsiteURL` e `DomainName` do recurso `S3Bucket`.

Note

Os exemplos apresentados a seguir pressupõem que as configurações do bloqueio de acesso público `BlockPublicPolicy` e `RestrictPublicBuckets` foram desabilitadas no nível da conta.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "S3Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "PublicAccessBlockConfiguration": {
          "BlockPublicAcls": false,
          "BlockPublicPolicy": false,
          "IgnorePublicAcls": false,
          "RestrictPublicBuckets": false
        },
        "WebsiteConfiguration": {
          "IndexDocument": "index.html",
          "ErrorDocument": "error.html"
        }
      },
      "DeletionPolicy": "Retain",
      "UpdateReplacePolicy": "Retain"
    },
    "BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "PolicyDocument": {
          "Id": "MyPolicy",
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "PublicReadForGetBucketObjects",
              "Effect": "Allow",
```

```

        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": {
            "Fn::Join": [
                "",
                [
                    "arn:aws:s3:::",
                    {
                        "Ref": "S3Bucket"
                    },
                    "/*"
                ]
            ]
        }
    ],
    "Bucket": {
        "Ref": "S3Bucket"
    }
}
},
"Outputs": {
    "WebsiteURL": {
        "Value": {
            "Fn::GetAtt": [
                "S3Bucket",
                "WebsiteURL"
            ]
        },
        "Description": "URL for website hosted on S3"
    },
    "S3BucketSecureURL": {
        "Value": {
            "Fn::Join": [
                "",
                [
                    "https://",
                    {
                        "Fn::GetAtt": [
                            "S3Bucket",
                            "DomainName"
                        ]
                    }
                ]
            ]
        }
    }
}

```

```

        }
      ]
    ],
    "Description": "Name of S3 bucket to hold website content"
  }
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: error.html
      DeletionPolicy: Retain
      UpdateReplacePolicy: Retain
  BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      PolicyDocument:
        Id: MyPolicy
        Version: 2012-10-17
        Statement:
          - Sid: PublicReadForGetBucketObjects
            Effect: Allow
            Principal: '*'
            Action: 's3:GetObject'
            Resource: !Join
              - ''
              - - 'arn:aws:s3:::'
                - !Ref S3Bucket
                - /*
            Bucket: !Ref S3Bucket

```

Outputs:**WebsiteURL:**

Value: !GetAtt

- S3Bucket
- WebsiteURL

Description: URL for website hosted on S3

S3BucketSecureURL:

Value: !Join

- ''
- - 'https://'
- !GetAtt
- S3Bucket
- DomainName

Description: Name of S3 bucket to hold website content

Criar um site estático usando um domínio personalizado

Você pode usar o Route 53 com um domínio registrado. O exemplo a seguir pressupõe que você já criou uma zona hospedada no Route 53 para seu domínio. O exemplo cria dois buckets para hospedagem de sites. O bucket raiz hospeda o conteúdo, e o outro bucket redireciona as solicitações `www.domainname.com` para o bucket raiz. Os conjuntos de registros mapeiam o nome do domínio para os endpoints do Amazon S3.

Também será necessário adicionar uma política de bucket, como mostrado nos exemplos acima.

Para obter mais informações sobre como usar um domínio personalizado, consulte [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#) no Guia do usuário do Amazon Simple Storage Service.

Note

Os exemplos apresentados a seguir pressupõem que as configurações do bloqueio de acesso público `BlockPublicPolicy` e `RestrictPublicBuckets` foram desabilitadas no nível da conta.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings" : {
```

```

    "RegionMap" : {
        "us-east-1" : { "S3hostedzoneID" : "Z3AQBSTGFYJSTF", "websiteendpoint" :
"s3-website-us-east-1.amazonaws.com" },
        "us-west-1" : { "S3hostedzoneID" : "Z2F56UZL2M1ACD", "websiteendpoint" :
"s3-website-us-west-1.amazonaws.com" },
        "us-west-2" : { "S3hostedzoneID" : "Z3BJ6K6RIION7M", "websiteendpoint" :
"s3-website-us-west-2.amazonaws.com" },
        "eu-west-1" : { "S3hostedzoneID" : "Z1BKCTXD74EZPE", "websiteendpoint" :
"s3-website-eu-west-1.amazonaws.com" },
        "ap-southeast-1" : { "S3hostedzoneID" : "Z300J2DXBE1FTB",
"websiteendpoint" : "s3-website-ap-southeast-1.amazonaws.com" },
        "ap-southeast-2" : { "S3hostedzoneID" : "Z1WCIGYICN2BYD",
"websiteendpoint" : "s3-website-ap-southeast-2.amazonaws.com" },
        "ap-northeast-1" : { "S3hostedzoneID" : "Z2M4EHUR26P7ZW",
"websiteendpoint" : "s3-website-ap-northeast-1.amazonaws.com" },
        "sa-east-1" : { "S3hostedzoneID" : "Z31GFT0UA1I2HV", "websiteendpoint" :
"s3-website-sa-east-1.amazonaws.com" }
    }
},
"Parameters": {
    "RootDomainName": {
        "Description": "Domain name for your website (example.com)",
        "Type": "String"
    }
},
"Resources": {
    "RootBucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "BucketName" : {"Ref":"RootDomainName"},
            "PublicAccessBlockConfiguration": {
                "BlockPublicAcls": false,
                "BlockPublicPolicy": false,
                "IgnorePublicAcls": false,
                "RestrictPublicBuckets": false
            },
            "WebsiteConfiguration": {
                "IndexDocument": "index.html",
                "ErrorDocument": "404.html"
            }
        }
    },
    "WWWBucket": {
        "Type": "AWS::S3::Bucket",

```

```

    "Properties": {
      "BucketName": {
        "Fn::Join": ["", ["www.", {"Ref": "RootDomainName"}]]
      },
      "AccessControl": "BucketOwnerFullControl",
      "WebsiteConfiguration": {
        "RedirectAllRequestsTo": {
          "HostName": {"Ref": "RootBucket"}
        }
      }
    },
    "myDNS": {
      "Type": "AWS::Route53::RecordSetGroup",
      "Properties": {
        "HostedZoneName": {
          "Fn::Join": ["", [{"Ref": "RootDomainName"}, "."]]
        },
        "Comment": "Zone apex alias.",
        "RecordSets": [
          {
            "Name": {"Ref": "RootDomainName"},
            "Type": "A",
            "AliasTarget": {
              "HostedZoneId": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "S3hostedzoneID"]},
              "DNSName": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "websiteendpoint"]}
            }
          },
          {
            "Name": {
              "Fn::Join": ["", ["www.", {"Ref": "RootDomainName"}]]
            },
            "Type": "CNAME",
            "TTL" : "900",
            "ResourceRecords" : [
              {"Fn::GetAtt":["WWWBucket", "DomainName"]}
            ]
          }
        ]
      }
    }
  },
},

```

```

    "Outputs": {
      "WebsiteURL": {
        "Value": {"Fn::GetAtt": ["RootBucket", "WebsiteURL"]},
        "Description": "URL for website hosted on S3"
      }
    }
  }
}

```

YAML

```

Parameters:
  RootDomainName:
    Description: Domain name for your website (example.com)
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      S3hostedzoneID: Z3AQBSTGFYJSTF
      websiteendpoint: s3-website-us-east-1.amazonaws.com
    us-west-1:
      S3hostedzoneID: Z2F56UZL2M1ACD
      websiteendpoint: s3-website-us-west-1.amazonaws.com
    us-west-2:
      S3hostedzoneID: Z3BJ6K6RIION7M
      websiteendpoint: s3-website-us-west-2.amazonaws.com
    eu-west-1:
      S3hostedzoneID: Z1BKCTXD74EZPE
      websiteendpoint: s3-website-eu-west-1.amazonaws.com
    ap-southeast-1:
      S3hostedzoneID: Z300J2DXBE1FTB
      websiteendpoint: s3-website-ap-southeast-1.amazonaws.com
    ap-southeast-2:
      S3hostedzoneID: Z1WCIGYICN2BYD
      websiteendpoint: s3-website-ap-southeast-2.amazonaws.com
    ap-northeast-1:
      S3hostedzoneID: Z2M4EHUR26P7ZW
      websiteendpoint: s3-website-ap-northeast-1.amazonaws.com
    sa-east-1:
      S3hostedzoneID: Z31GFT0UA1I2HV
      websiteendpoint: s3-website-sa-east-1.amazonaws.com
Resources:
  RootBucket:
    Type: AWS::S3::Bucket

```


Properties:

BucketName: !Ref RootDomainName

PublicAccessBlockConfiguration:

BlockPublicAcls: false

BlockPublicPolicy: false

IgnorePublicAcls: false

RestrictPublicBuckets: false

WebsiteConfiguration:

IndexDocument: index.html

ErrorDocument: 404.html

WWWBucket:

Type: AWS::S3::Bucket

Properties:

BucketName: !Sub

- www.\${Domain}

- Domain: !Ref RootDomainName

AccessControl: BucketOwnerFullControl

WebsiteConfiguration:

RedirectAllRequestsTo:

HostName: !Ref RootBucket

myDNS:

Type: AWS::Route53::RecordSetGroup

Properties:

HostedZoneName: !Sub

- \${Domain}.

- Domain: !Ref RootDomainName

Comment: Zone apex alias.

RecordSets:

- Name: !Ref RootDomainName

Type: A

AliasTarget:

HostedZoneId: !FindInMap [RegionMap, !Ref 'AWS::Region', S3hostedzoneID]

DNSName: !FindInMap [RegionMap, !Ref 'AWS::Region', websiteendpoint]

- Name: !Sub

- www.\${Domain}

- Domain: !Ref RootDomainName

Type: CNAME

TTL: 900

ResourceRecords:

- !GetAtt WWWBucket.DomainName

Outputs:**WebsiteURL:**

Value: !GetAtt RootBucket.WebsiteURL

Description: URL for website hosted on S3

Trechos de modelo do Amazon SNS

Este exemplo mostra um recurso de tópico do Amazon SNS. Ele requer um endereço de e-mail válido.

JSON

```
"MySNSTopic" : {
  "Type" : "AWS::SNS::Topic",
  "Properties" : {
    "Subscription" : [ {
      "Endpoint" : "add valid email address",
      "Protocol" : "email"
    } ]
  }
}
```

YAML

```
MySNSTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: "add valid email address"
        Protocol: email
```

Trechos de modelo do Amazon SQS

Este exemplo mostra uma fila do Amazon SQS.

JSON

```
"MyQueue" : {
  "Type" : "AWS::SQS::Queue",
  "Properties" : {
    "VisibilityTimeout" : "value"
  }
}
```

YAML

```
MyQueue:
  Type: AWS::SQS::Queue
  Properties:
    VisibilityTimeout: value
```

Trechos de modelo do Amazon Timestream

O Amazon Timestream para InfluxDB torna fácil para os desenvolvedores de aplicações e as equipes de DevOps executarem bancos de dados InfluxDB totalmente gerenciados na AWS para aplicações de série temporal em tempo real usando APIs de código aberto. Você pode criar rapidamente um banco de dados InfluxDB que lida com workloads de série temporal complexas. Com algumas simples chamadas de API, você pode configurar, migrar, operar e escalar um banco de dados InfluxDB na AWS com correções, backups e recuperação de software automatizados. Você também pode encontrar esses exemplos em [awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb](https://github.com/aws-labs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb) no GitHub.

Tópicos

- [Exemplo mínimo usando valores padrão](#)
- [Exemplo mais completo com parâmetros](#)

Esses modelos do CloudFormation criam os seguintes recursos que são necessários para a criação, a conexão e o monitoramento bem-sucedido de uma instância do Amazon Timestream para InfluxDB:

Amazon VPC

- VPC
- Uma ou mais Subnet
- InternetGateway
- RouteTable
- SecurityGroup

Amazon S3

- Bucket

Amazon Timestream

- InfluxDBInstance

Exemplo mínimo usando valores padrão

Esse exemplo implanta uma instância multi-AZ acessível ao público usando os valores padrão quando possível.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBPassword"
          ]
        }
      ],
      "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
      }
    }
  },
  "Parameters": {
    "DbInstanceName": {
      "Description": "The name that uniquely identifies the DB instance when interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a prefix included in the endpoint. DB instance names must be unique per customer and per Region.",
      "Type": "String",
      "Default": "mydbinstance",
      "MinLength": 3,
      "MaxLength": 40,
      "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
    },
    "InfluxDBPassword": {
      "Description": "The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative
```

tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.",

```

    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,
    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  }
},
"Resources": {
  "VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {"CidrBlock": "10.0.0.0/16"}
  },
  "InternetGateway": {"Type": "AWS::EC2::InternetGateway"},
  "InternetGatewayAttachment": {
    "Type": "AWS::EC2::VPCGatewayAttachment",
    "Properties": {
      "InternetGatewayId": {"Ref": "InternetGateway"},
      "VpcId": {"Ref": "VPC"}
    }
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          {"Fn::GetAZs": ""}
        ]
      },
      "CidrBlock": {
        "Fn::Select": [
          0,
          {
            "Fn::Cidr": [
              {
                "Fn::GetAtt": [
                  "VPC",
                  "CidrBlock"
                ]
              }
            ]
          }
        ],
        2,

```

```

        12
      ]
    }
  ]
},
  "MapPublicIpOnLaunch": true
}
},
"Subnet2": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "VpcId": {"Ref": "VPC"},
    "AvailabilityZone": {
      "Fn::Select": [
        1,
        {"Fn::GetAZs": ""}
      ]
    },
    "CidrBlock": {
      "Fn::Select": [
        1,
        {
          "Fn::Cidr": [
            {
              "Fn::GetAtt": [
                "VPC",
                "CidrBlock"
              ]
            },
            2,
            12
          ]
        }
      ]
    },
    "MapPublicIpOnLaunch": true
  }
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
    "VpcId": {"Ref": "VPC"}
  }
},

```

```
"DefaultRoute": {
  "Type": "AWS::EC2::Route",
  "DependsOn": "InternetGatewayAttachment",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "DestinationCidrBlock": "0.0.0.0/0",
    "GatewayId": {"Ref": "InternetGateway"}
  }
},
"Subnet1RouteTableAssociation": {
  "Type": "AWS::EC2::SubnetRouteTableAssociation",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "SubnetId": {"Ref": "Subnet1"}
  }
},
"Subnet2RouteTableAssociation": {
  "Type": "AWS::EC2::SubnetRouteTableAssociation",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "SubnetId": {"Ref": "Subnet2"}
  }
},
"InfluxDBSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupName": "influxdb-sg",
    "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
    "VpcId": {"Ref": "VPC"}
  }
},
"InfluxDBSecurityGroupIngress": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties": {
    "GroupId": {"Ref": "InfluxDBSecurityGroup"},
    "IpProtocol": "tcp",
    "CidrIp": "0.0.0.0/0",
    "FromPort": 8086,
    "ToPort": 8086
  }
},
"InfluxDBLogsS3Bucket": {
  "Type": "AWS::S3::Bucket",
  "DeletionPolicy": "Retain"
```

```

    },
    "InfluxDBLogsS3BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "s3:PutObject",
              "Effect": "Allow",
              "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*"},
              "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
            },
            {
              "Action": "s3:*",
              "Effect": "Deny",
              "Resource": [
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
              ],
              "Principal": "*",
              "Condition": {
                "Bool": {"aws:SecureTransport": false}
              }
            }
          ]
        }
      }
    },
    "DbInstance": {
      "Type": "AWS::Timestream::InfluxDBInstance",
      "DependsOn": "InfluxDBLogsS3BucketPolicy",
      "Properties": {
        "AllocatedStorage": 20,
        "DbInstanceType": "db.influx.medium",
        "Name": {"Ref": "DbInstanceName"},
        "Password": {"Ref": "InfluxDBPassword"},
        "PubliclyAccessible": true,
        "DeploymentType": "WITH_MULTIAZ_STANDBY",
        "VpcSecurityGroupIds": [
          {"Ref": "InfluxDBSecurityGroup"}
        ]
      }
    }
  }
}

```



```

        "VpcSubnetIds": [
            {"Ref": "Subnet1"},
            {"Ref": "Subnet2"}
        ],
        "LogDeliveryConfiguration": {
            "S3Configuration": {
                "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
                "Enabled": true
            }
        }
    },
    "Outputs": {
        "VPC": {
            "Description": "A reference to the VPC used to create network resources",
            "Value": {"Ref": "VPC"}
        },
        "Subnets": {
            "Description": "A list of the subnets created",
            "Value": {
                "Fn::Join": [
                    ",",
                    [
                        {"Ref": "Subnet1"},
                        {"Ref": "Subnet2"}
                    ]
                ]
            }
        },
        "Subnet1": {
            "Description": "A reference to the subnet in the 1st Availability Zone",
            "Value": {"Ref": "Subnet1"}
        },
        "Subnet2": {
            "Description": "A reference to the subnet in the 2nd Availability Zone",
            "Value": {"Ref": "Subnet2"}
        },
        "InfluxDBSecurityGroup": {
            "Description": "Security group with port 8086 ingress rule",
            "Value": {"Ref": "InfluxDBSecurityGroup"}
        },
        "InfluxDBLogsS3Bucket": {
            "Description": "S3 Bucket containing InfluxDB logs from the DB instance",

```

```

    "Value": {"Ref": "InfluxDBLogsS3Bucket"}
  },
  "DbInstance": {
    "Description": "A reference to the Timestream for InfluxDB DB instance",
    "Value": {"Ref": "DbInstance"}
  },
  "InfluxAuthParametersSecretArn": {
    "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
    "Value": {
      "Fn::GetAtt": [
        "DbInstance",
        "InfluxAuthParametersSecretArn"
      ]
    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",
              "Endpoint"
            ]
          },
          ":8086"
        ]
      ]
    }
  }
}

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:

```

```

ParameterGroups:
  -
    Label:
      default: "Amazon Timestream for InfluxDB Configuration"
    Parameters:
      - DbInstanceName
      - InfluxDBPassword
ParameterLabels:
  VPCCIDR:
    default: VPC CIDR

```

Parameters:

DbInstanceName:

Description: The name that uniquely identifies the DB instance when interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a prefix included in the endpoint. DB instance names must be unique per customer and per Region.

Type: String

Default: mydbinstance

MinLength: 3

MaxLength: 40

AllowedPattern: `^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$`

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

Type: String

NoEcho: true

MinLength: 8

MaxLength: 64

AllowedPattern: `^[a-zA-Z0-9]+$`

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/16

InternetGateway:

Type: AWS::EC2::InternetGateway

InternetGatewayAttachment:

Type: AWS::EC2::VPCElasticNetworkInterfaceAttachment

Properties:

InternetGatewayId: !Ref InternetGateway

```
VpcId: !Ref VPC
Subnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
    MapPublicIpOnLaunch: true
Subnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
    MapPublicIpOnLaunch: true
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
DefaultRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
Subnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
```

```

Properties:
  GroupId: !Ref InfluxDBSecurityGroup
  IpProtocol: tcp
  CidrIp: 0.0.0.0/0
  FromPort: 8086
  ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
          Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
          Principal:
            Service: timestream-influxdb.amazonaws.com
        - Action: "s3:*"
          Effect: Deny
          Resource:
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
          Principal: "*"
          Condition:
            Bool:
              aws:SecureTransport: false
DbInstance:
  Type: AWS::Timestream::InfluxDBInstance
  DependsOn: InfluxDBLogsS3BucketPolicy
  Properties:
    AllocatedStorage: 20
    DbInstanceType: db.influx.medium
    Name: !Ref DbInstanceName
    Password: !Ref InfluxDBPassword
    PubliclyAccessible: true
    DeploymentType: WITH_MULTIAZ_STANDBY
    VpcSecurityGroupIds:
      - !Ref InfluxDBSecurityGroup
    VpcSubnetIds:
      - !Ref Subnet1

```

```
- !Ref Subnet2
LogDeliveryConfiguration:
  S3Configuration:
    BucketName: !Ref InfluxDBLogsS3Bucket
    Enabled: true
```

Outputs:

```
# Network Resources
```

```
VPC:
```

```
  Description: A reference to the VPC used to create network resources
```

```
  Value: !Ref VPC
```

```
Subnets:
```

```
  Description: A list of the subnets created
```

```
  Value: !Join [",", [!Ref Subnet1, !Ref Subnet2]]
```

```
Subnet1:
```

```
  Description: A reference to the subnet in the 1st Availability Zone
```

```
  Value: !Ref Subnet1
```

```
Subnet2:
```

```
  Description: A reference to the subnet in the 2nd Availability Zone
```

```
  Value: !Ref Subnet2
```

```
InfluxDBSecurityGroup:
```

```
  Description: Security group with port 8086 ingress rule
```

```
  Value: !Ref InfluxDBSecurityGroup
```

```
# Timestream for InfluxDB Resources
```

```
InfluxDBLogsS3Bucket:
```

```
  Description: S3 Bucket containing InfluxDB logs from the DB instance
```

```
  Value: !Ref InfluxDBLogsS3Bucket
```

```
DbInstance:
```

```
  Description: A reference to the Timestream for InfluxDB DB instance
```

```
  Value: !Ref DbInstance
```

```
InfluxAuthParametersSecretArn:
```

```
  Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password."
```

```
  Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
```

```
Endpoint:
```

```
  Description: The endpoint URL to connect to InfluxDB
```

```
  Value: !Join ["/", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]
```

Exemplo mais completo com parâmetros

Esse exemplo de modelo altera dinamicamente os recursos de rede de acordo com os parâmetros fornecidos. Os parâmetros incluem PubliclyAccessible e DeploymentType.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Network Configuration"},
          "Parameters": ["VPCCIDR"]
        },
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBUsername",
            "InfluxDBPassword",
            "InfluxDBOrganization",
            "InfluxDBBucket",
            "DbInstanceType",
            "DbStorageType",
            "AllocatedStorage",
            "PubliclyAccessible",
            "DeploymentType"
          ]
        }
      ],
      "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
      }
    }
  },
  "Parameters": {
    "VPCCIDR": {
      "Description": "Please enter the IP range (CIDR notation) for the new VPC",
      "Type": "String",
      "Default": "10.0.0.0/16"
    },
    "DbInstanceName": {
```

```

    "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
    "Type": "String",
    "Default": "mydbinstance",
    "MinLength": 3,
    "MaxLength": 40,
    "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
  },
  "InfluxDBUsername": {
    "Description": "The username of the initial admin user created in InfluxDB. Must
start with a letter and can't end with a hyphen or contain two consecutive hyphens.
For example, my-user1. This username will allow you to access the InfluxDB UI to
perform various administrative tasks and also use the InfluxDB CLI to create an
operator token. These attributes will be stored in a Secret created in AWS Secrets
Manager in your account.",
    "Type": "String",
    "Default": "admin",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBPassword": {
    "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,
    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  },
  "InfluxDBOrganization": {
    "Description": "The name of the initial organization for the initial admin user
in InfluxDB. An InfluxDB organization is a workspace for a group of users.",
    "Type": "String",
    "Default": "org",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBBucket": {
    "Description": "The name of the initial InfluxDB bucket. All InfluxDB data
is stored in a bucket. A bucket combines the concept of a database and a retention

```



```

period (the duration of time that each data point persists). A bucket belongs to an
organization.",
  "Type": "String",
  "Default": "bucket",
  "MinLength": 2,
  "MaxLength": 64,
  "AllowedPattern": "^[^_\\\\""][^\\\\""]*$"
},
"DeploymentType": {
  "Description": "Specifies whether the Timestream for InfluxDB is deployed as
Single-AZ or with a MultiAZ Standby for High availability",
  "Type": "String",
  "Default": "WITH_MULTIAZ_STANDBY",
  "AllowedValues": [
    "SINGLE_AZ",
    "WITH_MULTIAZ_STANDBY"
  ]
},
"AllocatedStorage": {
  "Description": "The amount of storage to allocate for your DB storage type in GiB
(gibibytes).",
  "Type": "Number",
  "Default": 400,
  "MinValue": 20,
  "MaxValue": 16384
},
"DbInstanceType": {
  "Description": "The Timestream for InfluxDB DB instance type to run InfluxDB
on.",
  "Type": "String",
  "Default": "db.influx.medium",
  "AllowedValues": [
    "db.influx.medium",
    "db.influx.large",
    "db.influx.xlarge",
    "db.influx.2xlarge",
    "db.influx.4xlarge",
    "db.influx.8xlarge",
    "db.influx.12xlarge",
    "db.influx.16xlarge"
  ]
},
"DbStorageType": {

```

```

    "Description": "The Timestream for InfluxDB DB storage type to read and write
InfluxDB data.",
    "Type": "String",
    "Default": "InfluxIOIncludedT1",
    "AllowedValues": [
        "InfluxIOIncludedT1",
        "InfluxIOIncludedT2",
        "InfluxIOIncludedT3"
    ]
},
"PubliclyAccessible": {
    "Description": "Configures the DB instance with a public IP to facilitate
access.",
    "Type": "String",
    "Default": true,
    "AllowedValues": [
        true,
        false
    ]
}
},
"Conditions": {
    "IsMultiAZ": {
        "Fn::Equals": [
            {"Ref": "DeploymentType"},
            "WITH_MULTIAZ_STANDBY"
        ]
    },
    "IsPublic": {
        "Fn::Equals": [
            {"Ref": "PubliclyAccessible"},
            true
        ]
    }
},
"Resources": {
    "VPC": {
        "Type": "AWS::EC2::VPC",
        "Properties": {
            "CidrBlock": {"Ref": "VPCCIDR"}
        }
    },
    "InternetGateway": {
        "Type": "AWS::EC2::InternetGateway",

```

```

    "Condition": "IsPublic"
  },
  "InternetGatewayAttachment": {
    "Type": "AWS::EC2::VPCGatewayAttachment",
    "Condition": "IsPublic",
    "Properties": {
      "InternetGatewayId": {"Ref": "InternetGateway"},
      "VpcId": {"Ref": "VPC"}
    }
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          {"Fn::GetAZs": ""}
        ]
      },
      "CidrBlock": {
        "Fn::Select": [
          0,
          {
            "Fn::Cidr": [
              {
                "Fn::GetAtt": [
                  "VPC",
                  "CidrBlock"
                ]
              },
              2,
              12
            ]
          }
        ]
      },
      "MapPublicIpOnLaunch": {
        "Fn::If": [
          "IsPublic",
          true,
          false
        ]
      }
    }
  }
}

```

```

    }
  },
  "Subnet2": {
    "Type": "AWS::EC2::Subnet",
    "Condition": "IsMultiAZ",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          1,
          {"Fn::GetAZs": ""}
        ]
      },
    },
    "CidrBlock": {
      "Fn::Select": [
        1,
        {
          "Fn::Cidr": [
            {
              "Fn::GetAtt": [
                "VPC",
                "CidrBlock"
              ]
            },
            2,
            12
          ]
        }
      ]
    },
  },
  "MapPublicIpOnLaunch": {
    "Fn::If": [
      "IsPublic",
      true,
      false
    ]
  }
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
    "VpcId": {"Ref": "VPC"}
  }
}

```

```
    },
    "DefaultRoute": {
      "Type": "AWS::EC2::Route",
      "Condition": "IsPublic",
      "DependsOn": "InternetGatewayAttachment",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
      }
    },
    "Subnet1RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
      }
    },
    "Subnet2RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Condition": "IsMultiAZ",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet2"}
      }
    },
    "InfluxDBSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupName": "influxdb-sg",
        "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
        "VpcId": {"Ref": "VPC"}
      }
    },
    "InfluxDBSecurityGroupIngress": {
      "Type": "AWS::EC2::SecurityGroupIngress",
      "Properties": {
        "GroupId": {"Ref": "InfluxDBSecurityGroup"},
        "IpProtocol": "tcp",
        "CidrIp": "0.0.0.0/0",
        "FromPort": 8086,
        "ToPort": 8086
      }
    },
  },
```

```

    "InfluxDBLogsS3Bucket": {
      "Type": "AWS::S3::Bucket",
      "DeletionPolicy": "Retain"
    },
    "InfluxDBLogsS3BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "s3:PutObject",
              "Effect": "Allow",
              "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/
*"}},
              "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
            },
            {
              "Action": "s3:*",
              "Effect": "Deny",
              "Resource": [
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
                {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
              ],
              "Principal": "*",
              "Condition": {
                "Bool": {"aws:SecureTransport": false}
              }
            }
          ]
        }
      }
    },
    "DbInstance": {
      "Type": "AWS::Timestream::InfluxDBInstance",
      "DependsOn": "InfluxDBLogsS3BucketPolicy",
      "Properties": {
        "DbStorageType": {"Ref": "DbStorageType"},
        "AllocatedStorage": {"Ref": "AllocatedStorage"},
        "DbInstanceType": {"Ref": "DbInstanceType"},
        "Name": {"Ref": "DbInstanceName"},
        "Username": {"Ref": "InfluxDBUsername"},
        "Password": {"Ref": "InfluxDBPassword"},

```

```

    "Organization": {"Ref": "InfluxDBOrganization"},
    "Bucket": {"Ref": "InfluxDBBucket"},
    "PubliclyAccessible": {
      "Fn::If": [
        "IsPublic",
        true,
        false
      ]
    },
    "DeploymentType": {"Ref": "DeploymentType"},
    "VpcSecurityGroupIds": [
      {"Ref": "InfluxDBSecurityGroup"}
    ],
    "VpcSubnetIds": {
      "Fn::If": [
        "IsMultiAZ",
        [
          {"Ref": "Subnet1"},
          {"Ref": "Subnet2"}
        ],
        [
          {"Ref": "Subnet1"}
        ]
      ]
    },
    "LogDeliveryConfiguration": {
      "S3Configuration": {
        "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
        "Enabled": true
      }
    }
  },
  "Outputs": {
    "VPC": {
      "Description": "A reference to the VPC used to create network resources",
      "Value": {"Ref": "VPC"}
    },
    "Subnets": {
      "Description": "A list of the subnets created",
      "Value": {
        "Fn::If": [
          "IsMultiAZ",

```

```

        {
            "Fn::Join": [
                ",",
                [
                    {"Ref": "Subnet1"},
                    {"Ref": "Subnet2"}
                ]
            ],
            {"Ref": "Subnet1"}
        ]
    },
    "Subnet1": {
        "Description": "A reference to the subnet in the 1st Availability Zone",
        "Value": {"Ref": "Subnet1"}
    },
    "Subnet2": {
        "Condition": "IsMultiAZ",
        "Description": "A reference to the subnet in the 2nd Availability Zone",
        "Value": {"Ref": "Subnet2"}
    },
    "InfluxDBSecurityGroup": {
        "Description": "Security group with port 8086 ingress rule",
        "Value": {"Ref": "InfluxDBSecurityGroup"}
    },
    "InfluxDBLogsS3Bucket": {
        "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
        "Value": {"Ref": "InfluxDBLogsS3Bucket"}
    },
    "DbInstance": {
        "Description": "A reference to the Timestream for InfluxDB DB instance",
        "Value": {"Ref": "DbInstance"}
    },
    "InfluxAuthParametersSecretArn": {
        "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
        "Value": {
            "Fn::GetAtt": [
                "DbInstance",
                "InfluxAuthParametersSecretArn"
            ]
        }
    }
}

```



```

    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",
              "Endpoint"
            ]
          },
          ":8086"
        ]
      ]
    }
  }
}

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Network Configuration"
        Parameters:
          - VPCCIDR
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBUsername
          - InfluxDBPassword
          - InfluxDBOrganization
          - InfluxDBBucket
          - DbInstanceType

```

- DbStorageType
- AllocatedStorage
- PubliclyAccessible
- DeploymentType

ParameterLabels:

VPCCIDR:

default: VPC CIDR

Parameters:

Network Configuration

VPCCIDR:

Description: Please enter the IP range (CIDR notation) for the new VPC

Type: String

Default: 10.0.0.0/16

Timestream for InfluxDB Configuration

DbInstanceName:

Description: The name that uniquely identifies the DB instance when interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a prefix included in the endpoint. DB instance names must be unique per customer and per Region.

Type: String

Default: mydbinstance

MinLength: 3

MaxLength: 40

AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*\$

InfluxDB initial user configurations

InfluxDBUsername:

Description: The username of the initial admin user created in InfluxDB. Must start with a letter and can't end with a hyphen or contain two consecutive hyphens. For example, my-user1. This username will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

Type: String

Default: admin

MinLength: 1

MaxLength: 64

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS in your account.

Type: String

NoEcho: true

MinLength: 8

MaxLength: 64

AllowedPattern: `^[a-zA-Z0-9]+$`

InfluxDBOrganization:

Description: The name of the initial organization for the initial admin user in InfluxDB. An InfluxDB organization is a workspace for a group of users.

Type: String

Default: org

MinLength: 1

MaxLength: 64

InfluxDBBucket:

Description: The name of the initial InfluxDB bucket. All InfluxDB data is stored in a bucket. A bucket combines the concept of a database and a retention period (the duration of time that each data point persists). A bucket belongs to an organization.

Type: String

Default: bucket

MinLength: 2

MaxLength: 64

AllowedPattern: `^[^_"]"[^"]]*$`

DeploymentType:

Description: Specifies whether the Timestream for InfluxDB is deployed as Single-AZ or with a MultiAZ Standby for High availability

Type: String

Default: WITH_MULTIAZ_STANDBY

AllowedValues:

- SINGLE_AZ
- WITH_MULTIAZ_STANDBY

AllocatedStorage:

Description: The amount of storage to allocate for your DB storage type in GiB (gibibytes).

Type: Number

Default: 400

MinValue: 20

MaxValue: 16384

DbInstanceType:

Description: The Timestream for InfluxDB DB instance type to run InfluxDB on.

Type: String

Default: db.influx.medium

AllowedValues:

- db.influx.medium
- db.influx.large
- db.influx.xlarge
- db.influx.2xlarge
- db.influx.4xlarge

- db.influx.8xlarge
- db.influx.12xlarge
- db.influx.16xlarge

DbStorageType:

Description: The Timestream for InfluxDB DB storage type to read and write InfluxDB data.

Type: String

Default: InfluxIOIncludedT1

AllowedValues:

- InfluxIOIncludedT1
- InfluxIOIncludedT2
- InfluxIOIncludedT3

PubliclyAccessible:

Description: Configures the DB instance with a public IP to facilitate access.

Type: String

Default: true

AllowedValues:

- true
- false

Conditions:

IsMultiAZ: !Equals [!Ref DeploymentType, WITH_MULTIAZ_STANDBY]

IsPublic: !Equals [!Ref PubliclyAccessible, true]

Resources:**VPC:**

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VPCCIDR

InternetGateway:

Type: AWS::EC2::InternetGateway

Condition: IsPublic

InternetGatewayAttachment:

Type: AWS::EC2::VPCElasticNetworkInterfaceAttachment

Condition: IsPublic

Properties:

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref VPC

Subnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [0, !GetAZs '']

CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12]]

```
    MapPublicIpOnLaunch: !If [IsPublic, true, false]
Subnet2:
  Type: AWS::EC2::Subnet
  Condition: IsMultiAZ
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
    MapPublicIpOnLaunch: !If [IsPublic, true, false]
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
DefaultRoute:
  Type: AWS::EC2::Route
  Condition: IsPublic
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
Subnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Condition: IsMultiAZ
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !Ref InfluxDBSecurityGroup
    IpProtocol: tcp
    CidrIp: 0.0.0.0/0
```

```

    FromPort: 8086
    ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
          Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
          Principal:
            Service: timestream-influxdb.amazonaws.com
        - Action: "s3:*"
          Effect: Deny
          Resource:
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
          Principal: "*"
          Condition:
            Bool:
              aws:SecureTransport: false
DbInstance:
  Type: AWS::Timestream::InfluxDBInstance
  DependsOn: InfluxDBLogsS3BucketPolicy
  Properties:
    DbStorageType: !Ref DbStorageType
    AllocatedStorage: !Ref AllocatedStorage
    DbInstanceType: !Ref DbInstanceType
    Name: !Ref DbInstanceName
    Username: !Ref InfluxDBUsername
    Password: !Ref InfluxDBPassword
    Organization: !Ref InfluxDBOrganization
    Bucket: !Ref InfluxDBBucket
    PubliclyAccessible: !If [IsPublic, true, false]
    DeploymentType: !Ref DeploymentType
    VpcSecurityGroupIds:
      - !Ref InfluxDBSecurityGroup
    VpcSubnetIds: !If
      - IsMultiAZ

```

```

-
  - !Ref Subnet1
  - !Ref Subnet2
-
  - !Ref Subnet1
LogDeliveryConfiguration:
  S3Configuration:
    BucketName: !Ref InfluxDBLogsS3Bucket
    Enabled: true

```

Outputs:

Network Resources

VPC:

Description: A reference to the VPC used to create network resources

Value: !Ref VPC

Subnets:

Description: A list of the subnets created

Value: !If

- IsMultiAZ
- !Join [",", [!Ref Subnet1, !Ref Subnet2]]
- !Ref Subnet1

Subnet1:

Description: A reference to the subnet in the 1st Availability Zone

Value: !Ref Subnet1

Subnet2:

Condition: IsMultiAZ

Description: A reference to the subnet in the 2nd Availability Zone

Value: !Ref Subnet2

InfluxDBSecurityGroup:

Description: Security group with port 8086 ingress rule

Value: !Ref InfluxDBSecurityGroup

Timestream for InfluxDB Resources

InfluxDBLogsS3Bucket:

Description: S3 Bucket containing InfluxDB logs from the DB instance

Value: !Ref InfluxDBLogsS3Bucket

DbInstance:

Description: A reference to the Timestream for InfluxDB DB instance

Value: !Ref DbInstance

InfluxAuthParametersSecretArn:

Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret containing the initial InfluxDB authorization parameters. The secret value is a JSON formatted key-value pair holding InfluxDB authorization values: organization, bucket, username, and password."

```
Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
Endpoint:
  Description: The endpoint URL to connect to InfluxDB
  Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]
```

Implantar pilhas baseadas em Windows usando o CloudFormation

Esta página fornece links para documentação de referência técnica dos recursos do CloudFormation comumente usados em implantações baseadas em Windows.

O CloudFormation oferece suporte para a implantação e o gerenciamento de pilhas Microsoft Windows via Infrastructure as Code (IaC). Você pode usar o CloudFormation para provisionamento automatizado de instâncias do EC2 baseadas em Windows, SQL Server no Amazon RDS e Microsoft Active Directory via Directory Service.

A AWS fornece Imagens de máquina da Amazon (AMIs) pré-configuradas e projetadas especificamente para plataformas Windows para ajudar você a implantar rapidamente aplicações no Amazon EC2. Essas AMIs incluem configurações padrão da Microsoft e personalizações específicas para a AWS. Com o CloudFormation, é possível escolher uma AMI apropriada, iniciar uma instância e acessá-la usando a Conexão de Área de Trabalho Remota, exatamente da mesma forma que faria com qualquer outro Windows Server. As AMIs contêm componentes essenciais de software, incluindo o EC2Launch (as versões variam de acordo com a edição do Windows Server), AWS Systems Manager, CloudFormation, Ferramentas da AWS para PowerShell e vários drivers de rede, armazenamento e vídeo para garantir performance e compatibilidade otimizadas com os serviços da AWS. Para obter mais informações, consulte a [Referência da AMI do Windows da AWS](#).

O CloudFormation também oferece suporte a ferramentas de configuração de software, como scripts UserData, que podem executar comandos do PowerShell ou comandos em lote quando uma instância do EC2 é inicializada pela primeira vez. Ele também oferece scripts auxiliares (cfn-init, cfn-signal, cfn-get-metadata e cfn-hup) e oferece suporte a metadados AWS::CloudFormation::Init para gerenciar pacotes, arquivos e serviços em instâncias Windows.

Para ambientes corporativos, o CloudFormation permite unir domínios, gerenciar licenças do Windows via modelos de licenciamento do EC2 e lidar de forma segura com credenciais via AWS Secrets Manager. Combinado com modelos controlados por versão e implantações repetíveis, o CloudFormation ajuda as organizações a manter ambientes Windows consistentes, seguros e escaláveis em várias contas e Regiões da AWS.

Para obter detalhes sobre os recursos do CloudFormation comumente usados em implantações baseadas em Windows, consulte os tópicos de referência técnica a seguir.

Tipo de recurso	Descrição
AWS::EC2::Instance	Para iniciar instâncias do EC2 do Windows.
AWS::EC2::SecurityGroup	Para definir regras de firewall para workloads do Windows.
AWS::AutoScaling::AutoScalingGroup	Para escalar instâncias do EC2 do Windows.
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftAD	Para implantar o Microsoft Active Directory.
AWS::FSx::FileSystem	Para implantar o FSx for Windows File Server.
AWS::RDS::DBInstance	Para provisionamento de SQL Server no Amazon RDS.
AWS::CloudFormation::Init	Usado nos metadados do EC2 para configurar instâncias. Para obter mais informações, consulte Fazer o bootstrap de pilhas do CloudFormation baseadas em Windows .
AWS::SecretsManager::Secret	Para gerenciar credenciais e senhas do Windows com segurança.
AWS::SSM::Parameter	Para armazenar valores de configuração com segurança.
AWS::IAM::InstanceProfile	Para conceder permissões a aplicações em execução em instâncias do EC2.
AWS::IAM::Perfil	

Fazer o bootstrap de pilhas do CloudFormation baseadas em Windows

Este tópico descreve como fazer bootstrap de uma pilha do Windows e solucionar problemas na criação da pilha.

Tópicos

- [Dados do usuário em instâncias do EC2](#)
- [CloudFormation scripts auxiliares](#)
- [Exemplo de bootstrap de uma pilha do Windows](#)
- [Escapar barras invertidas nos caminhos dos arquivos do Windows](#)
- [Gerenciar serviços da Windows](#)
- [Como solucionar problemas de criação de pilhas](#)

Dados do usuário em instâncias do EC2

Os dados do usuário são um recurso do Amazon EC2 que permite que você passe scripts ou informações de configuração para uma instância do EC2 quando ela é inicializada.

Para instâncias do EC2 para Windows:

- Você pode usar scripts em lote (usando tags `<script>`) ou scripts PowerShell (usando tags `<powershell>`).
- A execução de scripts é realizada por EC2Launch.

Important

Se você estiver criando sua própria AMI do Windows para uso com o CloudFormation, verifique se o EC2Launch v2 está configurado corretamente. O EC2Launch v2 é necessário para que as ferramentas de bootstrap do CloudFormation inicializem e configurem adequadamente as instâncias Windows durante a criação da pilha. Para obter mais informações, consulte [Usar o agente do EC2Launch v2 para realizar tarefas durante a inicialização da instância do EC2 do Windows](#) no Guia do usuário do Amazon EC2. Para obter informações sobre as AMIs da AWS para Windows disponíveis, consulte [AWS Windows AMI Reference](#).

CloudFormation scripts auxiliares

Os scripts auxiliares são utilitários para configurar instâncias durante o processo de bootstrap. Usados com os dados do usuário do Amazon EC2, eles oferecem opções de configuração avançadas.

O CloudFormation fornece os seguintes scripts auxiliares do Python que podem ser usados para instalar software e iniciar serviços em uma instância do Amazon EC2 criada por você como parte da pilha:

- `cfn-init`: usado para recuperar e interpretar os metadados de recursos, instalar pacotes, criar arquivos e iniciar serviços.
- `cfn-signal`: usado para sinalizar com uma `CreationPolicy` para que você possa sincronizar outros recursos da pilha quando o recurso ou a aplicação exigidos estiverem prontos.
- `cfn-get-metadata`: usado para recuperar metadados de um recurso ou caminho para uma chave específica.
- `cfn-hup`: usado para verificar se há atualizações para os metadados e executar hooks personalizados quando alterações são detectadas.

Você chama os scripts diretamente do seu modelo. Os scripts funcionam em conjunto com os metadados de recurso que estão definidos no mesmo modelo. Os scripts são executados na instância do Amazon EC2 durante o processo de criação da pilha.

Para obter mais informações, consulte [Referência de scripts auxiliares do CloudFormation](#) no Guia de referência de modelos do CloudFormation.

Exemplo de bootstrap de uma pilha do Windows

Vamos examinar exemplos de trechos de um modelo do Windows Server que executa as seguintes ações:

- Lança uma instância do EC2 denominada `TestInstance` em uma AMI do Windows Server 2022.
- Cria um arquivo de teste simples para verificar se `cfn-init` está funcionando.
- Configura `cfn-hup` para gerenciamento contínuo das configurações.
- Usa uma `CreationPolicy` para garantir que a instância sinalize conclusões bem-sucedidas.

O script auxiliar `cfn-init` é usado para realizar cada uma dessas ações com base em informações do recurso `AWS::CloudFormation::Init` no modelo.

A seção `AWS::CloudFormation::Init` é denominada `TestInstance` e começa com a declaração a seguir.

```
TestInstance:
```

```
Type: AWS::EC2::Instance
Metadata:
  AWS::CloudFormation::Init:
    configSets:
      default:
        - create_files
        - start_services
```

A seguir, a seção `files` de `AWS::CloudFormation::Init` é declarada.

```
create_files:
  files:
    c:\cfn\test.txt:
      content: !Sub |
        Hello from ${AWS::StackName}
    c:\cfn\cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackName}
        region=${AWS::Region}
        interval=2
    c:\cfn\hooks.d\cfn-auto-reloader.conf:
      content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.TestInstance.Metadata.AWS::CloudFormation::Init
        action=cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --
region ${AWS::Region}
```

Três arquivos são criados aqui e colocados no diretório `C:\cfn` da instância do servidor:

- `test.txt`, um arquivo de teste simples que verifica se `cfn-init` está funcionando corretamente e pode criar arquivos com conteúdo dinâmico.
- `cfn-hup.conf`, o arquivo de configuração de `cfn-hup` com um intervalo de verificação de 2 minutos.
- `cfn-auto-reloader.conf`, o arquivo de configuração do hook usado por `cfn-hup` para iniciar uma atualização (chamando `cfn-init`) quando os metadados em `AWS::CloudFormation::Init` são alterados.

A próxima seção `start_services` configura serviços do Windows.

```

start_services:
  services:
    windows:
      cfn-hup:
        enabled: true
        ensureRunning: true
        files:
          - c:\cfn\cfn-hup.conf
          - c:\cfn\hooks.d\cfn-auto-reloader.conf

```

Esta seção garante que o serviço `cfn-hup` seja iniciado e que será reiniciado automaticamente se os arquivos de configuração forem modificados. O serviço monitora alterações nos metadados do CloudFormation e executa `cfn-init` novamente quando atualizações são detectadas.

A próxima seção é `Properties`.

```

TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Metadata:
    AWS::CloudFormation::Init:
      # ... metadata configuration ...
  Properties:
    InstanceType: t2.large
    ImageId: '{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base}}'
    SecurityGroupIds:
      - !Ref InstanceSecurityGroup
    KeyName: !Ref KeyPairName
    UserData:
      Fn::Base64: !Sub |
        <powershell>
          cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --region
${AWS::Region}
          cfn-signal.exe -e $lastexitcode --stack ${AWS::StackName} --resource
TestInstance --region ${AWS::Region}
        </powershell>

```

Nesta seção, a propriedade `UserData` contém um script PowerShell que será executado por `EC2Launch`, entre tags `<powershell>`. O script executa `cfn-init` com o `configSet default` e

depois usa `cfn-signal` para retornar o código de saída ao CloudFormation. A `CreationPolicy` é usada para garantir que a instância esteja configurada corretamente antes que a criação da pilha seja considerada concluída.

A propriedade `ImageId` usa um parâmetro público do Systems Manager Parameter Store para recuperar automaticamente o ID da AMI mais recente do Windows Server 2022 . Essa abordagem elimina a necessidade de mapeamentos de AMIs específicas da região e garante que você sempre obtenha a AMI mais recente. O uso de parâmetros do Systems Manager para IDs de AMIs é uma prática recomendada para manter as referências de AMIs atualizadas. Se você planejar se conectar à instância, certifique-se de a propriedade `SecurityGroupIds` referencie um grupo de segurança que permita acesso por RDP.

A `CreationPolicy` é declarada como parte das propriedades do recurso e especifica um tempo limite. O comando `cfn-signal` nos dados do usuário sinaliza quando a configuração da instância foi concluída:

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Properties:
    # ... other properties ...
```

Como o processo de inicialização é mínimo e apenas cria arquivos e inicia serviços, a `CreationPolicy` aguarda 20 minutos (PT20M) antes de atingir o tempo limite. O tempo limite é especificado usando o formato de duração ISO 8601. Observe que as instâncias do Windows geralmente demoram mais para serem iniciadas que as instâncias do Linux, por isso, teste cuidadosamente para determinar os melhores valores de tempo limite para as suas necessidades.

Se tudo correr bem, a `CreationPolicy` será concluída com êxito e você poderá acessar a instância do Windows Server usando seu endereço IP público. Depois que a criação da pilha é concluída, o ID e endereço IP da instância são exibidos na guia Saídas do console do CloudFormation.

```
Outputs:
  InstanceId:
    Value: !Ref TestInstance
    Description: Instance ID of the Windows Server
  PublicIP:
```

```
Value: !GetAtt TestInstance.PublicIp
Description: Public IP address of the Windows Server
```

Você também pode verificar manualmente se a inicialização funcionou corretamente conectando-se à instância por RDP e verificando se o arquivo `C:\cfn\test.txt` existe e contém o conteúdo esperado. Para obter informações sobre conexão a instâncias do Windows, consulte [Conexão com a instância do Windows usando um cliente RDP](#) no Manual do usuário do Amazon EC2.

Escapar barras invertidas nos caminhos dos arquivos do Windows

Ao referenciar os caminhos do Windows nos modelos do CloudFormation, lembre-se sempre de escapar adequadamente as barras invertidas (\) de acordo com o formato do modelo sendo usado.

- Para modelos JSON, você deve usar barras invertidas duplas nos caminhos dos arquivos do Windows porque o JSON trata a barra invertida como um caractere de escape. A primeira barra invertida escapa da segunda, resultando na interpretação de uma única barra invertida literal.

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\SharePoint\\SharePointFoundation2010.exe /extract:C:\\SharePoint
\\SPF2010 /quiet /log:C:\\SharePoint\\SharePointFoundation2010-extract.log"
  }
}
```

- Para modelos YAML, barras invertidas simples normalmente são suficientes.

```
commands:
  1-extract:
    command: C:\SharePoint\SharePointFoundation2010.exe /extract:C:\SharePoint
\SPF2010 /quiet /log:C:\SharePoint\SharePointFoundation2010-extract.log
```

Gerenciar serviços da Windows

Os serviços do Windows são gerenciados da mesma forma que os serviços do Linux, exceto pelo uso de uma chave `windows` em vez de `sysvinit`. O exemplo a seguir iniciará o serviço `cfn-hup`, definirá o serviço como automático e o reiniciará se `cfn-init` modificar os arquivos de configuração `c:\cfn\cfn-hup.conf` ou `c:\cfn\hooks.d\cfn-auto-reloader.conf`.

```
services:
```

```
windows:
  cfn-hup:
    enabled: true
    ensureRunning: true
    files:
      - c:\cfn\cfn-hup.conf
      - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

É possível gerenciar outros serviços do Windows da mesma maneira, usando o nome, não o nome de exibição, para referenciar o serviço.

Como solucionar problemas de criação de pilhas

Em caso de falha na pilha durante a criação, o comportamento padrão é de reversão em caso de falha. Embora normalmente seja um bom padrão, pois evita cobranças desnecessários, isso dificulta a depuração por causa da falha na criação da pilha.

Para desativar esse comportamento ao criar ou atualizar sua pilha com o console do CloudFormation, escolha a opção Preservar recursos provisionados com sucesso em Opções de falha de pilha. Para obter mais informações, consulte [Escolha como lidar com falhas ao provisionar recursos](#). Isso permite a você fazer login na instância e visualizar os arquivos de log para identificar os problemas encontrados na execução dos scripts de startup.

Os logs importantes a serem observados são:

- O log de configuração do EC2 em %ProgramData%\Amazon\EC2Launch\log\agent.log
- O log de cfn-init em C:\cfn\log\cfn-init.log (verifique os códigos de saída e as mensagens de erro para pontos de falha específicos)

Para obter mais logs, consulte os seguintes tópicos no Guia do usuário do Amazon EC2:

- [EC2Launch Estrutura de diretórios do](#)
- [EC2Launch Estrutura de diretório do v](#)

Para obter mais informações sobre como solucionar problemas de inicialização, consulte [How do I troubleshoot helper scripts that won't bootstrap in a CloudFormation stack with Windows instances?](#).

Amplie os recursos do seu modelo com os tipos de recursos fornecidos pelo CloudFormation

O CloudFormation oferece vários tipos de recursos que você pode usar em seu modelo de pilha para estender suas capacidades além daquelas de um simples modelo de pilha.

Esses tipos de recursos incluem:

Tipo de recurso	Descrição	Documentação
Recursos personalizados	O tipo de recurso <code>AWS::CloudFormation::CustomResource</code> permite criar recursos personalizados capazes de executar tarefas de provisionamento específicas ou incluir recursos que não estão disponíveis como tipos de recursos do CloudFormation.	Recursos personalizados
Macros	O tipo de recurso <code>AWS::CloudFormation::Macro</code> define um trecho reutilizável de código que pode executar o processamento personalizado nos modelos do CloudFormation. Macros podem modificar seus modelos, gerar recursos adicionais ou realizar outras operações personalizadas durante a criação ou as atualizações da pilha.	Macros de modelo
Pilhas aninhadas	O tipo de recurso <code>AWS::CloudFormation::Stack</code> permite que você crie pilhas aninhadas em seus modelos do CloudFormation para obter arquiteturas de pilha mais modulares e reutilizáveis.	Pilhas aninhadas
StackSet	O tipo de recurso <code>AWS::CloudFormation::StackSet</code> cria ou atualiza um CloudFormation StackSet, que é um contêiner para pilhas que pode ser implantado em várias regiões e Contas da AWS.	Gerenciar pilhas com StackSets

Tipo de recurso	Descrição	Documentação
Condição de espera	O tipo de recurso <code>AWS::CloudFormation::WaitCondition</code> pausa a criação ou atualização da pilha até que uma condição específica seja atendida, como a conclusão bem-sucedida de um processo de longa execução ou a disponibilidade de recursos externos.	Condições de espera
Processamento de condição de espera	O tipo de recurso <code>AWS::CloudFormation::WaitConditionHandle</code> funciona junto com o tipo de recurso <code>AWS::CloudFormation::WaitCondition</code> . Ele fornece um URL pré-assinado que é usado para enviar sinais indicando que uma determinada condição foi atendida. Esses sinais permitem que o processo de criação ou atualização da pilha continue.	Condições de espera

Crie uma lógica de provisionamento personalizada com recursos personalizados

Os recursos personalizados fornecem uma maneira de gravar uma lógica de provisionamento personalizada nos modelos do CloudFormation e fazer com que o CloudFormation a execute sempre que você criar, atualizar (se você alterou o recurso personalizado) ou excluir uma pilha. Isso pode ser útil quando os requisitos de provisionamento envolvem uma lógica complexa ou fluxos de trabalho que não podem ser expressos com os tipos de recursos integrados do CloudFormation.

Por exemplo, talvez você deseje incluir recursos que não estão disponíveis como tipos de recursos do CloudFormation. Você pode incluir esses recursos usando recursos personalizados. Dessa forma, você ainda poderá gerenciar todos os recursos relacionados em uma única pilha.

Para definir um recurso personalizado em seu modelo do CloudFormation, você usa o tipo de recurso `AWS::CloudFormation::CustomResource` ou `Custom::MyCustomResourceTypeName`. Os recursos personalizados requerem uma propriedade,

o token de serviço, que especifica para qual local o CloudFormation envia as solicitações, como um tópico do Amazon SNS ou uma função do Lambda.

Os tópicos apresentados a seguir fornecem informações sobre como usar os recursos personalizados.

Tópicos

- [Como os recursos personalizados funcionam](#)
- [Tempo limite de resposta](#)
- [Referência personalizada de solicitações e respostas de recursos do CloudFormation](#)
- [Recursos personalizados baseados no Amazon SNS](#)
- [Recursos personalizados baseados no Lambda](#)

Note

O registro e os recursos personalizados do CloudFormation oferecem seus próprios benefícios. Os recursos personalizados oferecem os seguintes benefícios:

- Você não precisa registrar o recurso.
- Você pode incluir um recurso inteiro como parte de um modelo sem registrá-lo.
- Aceita as operações Create, Update e Delete

As vantagens que os recursos baseados no registro oferecem são as seguintes:

- Oferece suporte à modelagem, ao provisionamento e ao gerenciamento de recursos de aplicações de terceiros
- Suporta as operações Create, Read, Update, Delete e List (CRUDL)
- Suporta detecção de desvio em tipos de recursos privados e de terceiros

Ao contrário dos recursos personalizados, os recursos baseados em registro não precisarão associar um tópico do Amazon SNS ou uma função Lambda para executar operações CRUDL. Para obter mais informações, consulte [Gerenciar extensões com o registro do CloudFormation](#).

Como os recursos personalizados funcionam

O processo geral para configurar um novo recurso personalizado inclui as etapas a seguir. Essas etapas envolvem dois perfis: o provedor de recursos personalizados que é o proprietário do recurso personalizado e o desenvolvedor de modelos que cria um modelo que inclui um tipo de recurso personalizado. Eles podem ser a mesma pessoa, mas, se não forem, o provedor de recursos personalizados deverá trabalhar com o desenvolvedor de modelos.

1. O provedor de recursos personalizados escreve uma lógica que determina como lidar com as solicitações do CloudFormation e realizar ações no recurso personalizado.
2. O provedor de recursos personalizados cria o tópico do Amazon SNS ou a função do Lambda para a qual o CloudFormation pode enviar solicitações. O tópico do Amazon SNS ou a função do Lambda devem estar na mesma região em que a pilha será criada.
3. O provedor de recursos personalizados fornece o ARN do tópico do Amazon SNS ou o ARN da função do Lambda ao desenvolvedor de modelos.
4. O desenvolvedor de modelos define o recurso personalizado no modelo do CloudFormation. Isso inclui um token de serviço e todos os parâmetros de dados de entrada. O token de serviço e a estrutura dos dados de entrada são definidos pelo provedor de recursos personalizados. O token de serviço especifica o ARN do tópico do Amazon SNS ou o ARN da função do Lambda e sempre é obrigatório, mas os dados de entrada são opcionais, dependendo do recurso personalizado.

Sempre que alguém usa o modelo para criar, atualizar ou excluir o recurso personalizado, o CloudFormation envia uma solicitação para o token de serviço especificado e espera por uma resposta antes de prosseguir com a operação da pilha.

Este é o resumo do fluxo de criação de uma pilha usando o modelo:

1. O CloudFormation envia uma solicitação para o token de serviço especificado. A solicitação inclui informações como o tipo de solicitação e um URL do bucket do Amazon S3 pré-assinado, para o qual o recurso personalizado envia as respostas. Para obter mais informações sobre o que está incluído na solicitação, consulte [Referência personalizada de solicitações e respostas de recursos do CloudFormation](#).

O exemplo de dados a seguir mostra o que o CloudFormation inclui em uma solicitação de Create. Neste exemplo, ResourceProperties permite que o CloudFormation crie uma carga útil personalizada para enviar à função do Lambda.

```
{
  "RequestType" : "Create",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::TestResource",
  "LogicalResourceId" : "MyTestResource",
  "ResourceProperties" : {
    "Name" : "Value",
    "List" : [ "1", "2", "3" ]
  }
}
```

2. O provedor de recursos personalizados realiza o processamento da solicitação do CloudFormation e retorna uma resposta de SUCCESS ou de FAILED para o URL assinado previamente. O custom resource provider fornece a resposta em um arquivo formatado em JSON e o carrega no URL do S3 pré-assinado. Para obter mais informações, consulte [Carregar objetos com URLs pré-assinados](#) no Guia do Usuário do Amazon Simple Storage Service.

Na resposta, o custom resource provider também pode incluir pares de nome e valor que o template developer pode acessar. Por exemplo, a resposta poderá incluir dados de saída se a solicitação tiver sido bem-sucedida ou uma mensagem de erro se a solicitação tiver falhado. Para obter mais informações sobre respostas, consulte [Referência personalizada de solicitações e respostas de recursos do CloudFormation](#).

Important

Se os pares de nome/valor contiverem informações confidenciais, você deverá utilizar o campo NoEcho para mascarar a saída do recurso personalizado. Caso contrário, os valores estarão visíveis por meio de APIs que exibem valores de propriedades (como DescribeStackEvents).

Para obter mais informações sobre como usar NoEcho para mascarar informações confidenciais, consulte a prática recomendada [Não incorporar credenciais em seus modelos](#).

O custom resource provider é responsável por escutar e responder à solicitação. Por exemplo, para notificações do Amazon SNS, o provedor de recursos personalizados deve atuar

como receptor e responder às notificações enviadas para um ARN de tópico específico. O CloudFormation espera e atua como receptor para uma resposta no local do URL assinado previamente.

Os seguintes dados de exemplo mostram o que um recurso personalizado pode incluir em uma resposta:

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MyTestResource",
  "PhysicalResourceId" : "TestResource1",
  "Data" : {
    "OutputName1" : "Value1",
    "OutputName2" : "Value2",
  }
}
```

3. Após obter uma resposta de SUCCESS, o CloudFormation prossegue com a operação de pilha. Caso uma resposta FAILED ou nenhuma resposta seja retornada, a operação falhará. Todos os dados de saída do recurso personalizado são armazenados no local do URL pré-assinado. O desenvolvedor de modelos pode recuperar os dados usando a função [Fn::GetAtt](#).

Note

Se você usar o AWS PrivateLink, os recursos personalizados da VPC deverão ter acesso aos buckets do S3 específicos para o CloudFormation. Os recursos personalizados devem enviar respostas a um URL pré-assinado do Amazon S3. Se não puderem enviar respostas para o Amazon S3, o CloudFormation não receberá uma resposta e há falha na operação de pilha. Para obter mais informações, consulte [Acessar o CloudFormation usando um endpoint de interface \(AWS PrivateLink\)](#).

Tempo limite de resposta

O tempo limite padrão para o recurso personalizado é de 3600 segundos (1 hora). Se nenhuma resposta for recebida durante esse período, a operação de pilha falhará.

É possível ajustar o valor do tempo limite com base em quanto tempo você espera que o recurso personalizado leve para responder. Por exemplo, ao provisionar um recurso personalizado que invoca uma função do Lambda que deve responder em cinco minutos, você pode definir um tempo limite de cinco minutos no modelo de pilha especificando a propriedade `ServiceTimeout`. Para obter mais informações, consulte [Referência personalizada de solicitações e respostas de recursos do CloudFormation](#). Dessa forma, se houver um erro na função do Lambda que provoque sua paralisação, o CloudFormation determinará que houve falha na operação de pilha após cinco minutos, em vez de esperar a hora inteira.

Porém, tenha cuidado para não definir um valor de tempo limite muito pequeno. Para evitar que o tempo limite se esgote inesperadamente, certifique-se de que o recurso personalizado tenha tempo suficiente para realizar as ações necessárias e retornar uma resposta.

Referência personalizada de solicitações e respostas de recursos do CloudFormation

O CloudFormation gerencia recursos personalizados por meio de um protocolo de solicitação-resposta que se comunica com seu provedor de recursos personalizados. Cada solicitação inclui um tipo de solicitação (`Create`, `Update` ou `Delete`) e segue esse fluxo de trabalho de alto nível:

1. Um desenvolvedor de modelos define um recurso personalizado com um `ServiceToken` e `ServiceTimeout` no modelo e inicia uma operação de pilha.
2. O CloudFormation envia uma solicitação JSON ao provedor de recursos personalizados por meio do SNS ou do Lambda.
3. O provedor de recursos personalizados processa a solicitação e retorna uma resposta JSON para um URL pré-assinado do bucket do Amazon S3 antes que o período de tempo limite expire.
4. O CloudFormation lê a resposta e prossegue para a operação da pilha. Se nenhuma resposta for recebida antes do tempo limite terminar, a solicitação será considerada malsucedida e a operação de pilha falhará.

Para obter mais informações, consulte [Como os recursos personalizados funcionam](#).

Esta seção descreve a estrutura, os parâmetros e as respostas esperadas para cada tipo de solicitação.

Note

O tamanho total do corpo da resposta não pode exceder 4096 bytes.

Configuração do modelo

Ao definir um recurso personalizado em um modelo, o desenvolvedor do modelo usa [AWS::CloudFormation::CustomResource](#) com as seguintes propriedades:

ServiceToken

Um ARN de tópico do Amazon SNS ou um ARN de função do Lambda da mesma região que a pilha.

Obrigatório: Sim

Tipo: string

ServiceTimeout

O tempo máximo, em segundos, antes que a operação de um recurso personalizado atinja o tempo limite. Esse valor deve estar entre 1 e 3600. Padrão: 3600 segundos (1 hora).

Obrigatório: não

Tipo: string

São compatíveis propriedades adicionais de recursos. As propriedades do recurso serão incluídas como `ResourceProperties` na solicitação. O provedor de recursos personalizados deve determinar quais propriedades são válidas e seus valores aceitáveis.

Objeto de solicitação

Create

Quando o desenvolvedor de modelos cria uma pilha que contém um recurso personalizado, o CloudFormation envia uma solicitação com `RequestType` definido como `Create`.

Crie solicitações que contenha os seguintes campos:

RequestType

`Create`.

Obrigatório: Sim

Tipo: string

RequestId

Um ID exclusivo para a solicitação.

Combinar `StackId` com `RequestId` forma um valor que é possível utilizar para identificar exclusivamente uma solicitação em um recurso personalizado específico.

Obrigatório: Sim

Tipo: string

StackId

O nome do recurso da Amazon (ARN) que identifica a pilha que contém o recurso personalizado.

Combinar `StackId` com `RequestId` forma um valor que é possível utilizar para identificar exclusivamente uma solicitação em um recurso personalizado específico.

Obrigatório: Sim

Tipo: string

ResponseURL

O URL de resposta identifica um bucket do S3 pré-assinado que recebe respostas do provedor de recursos personalizados para o CloudFormation.

Obrigatório: Sim

Tipo: string

ResourceType

O tipo de recurso escolhido pelo desenvolvedor do modelo do recurso personalizado no modelo do CloudFormation. Os nomes de tipos de recursos personalizados podem incluir até 60 caracteres, incluindo alfanuméricos e os seguintes caracteres: `_@-`.

Obrigatório: Sim

Tipo: string

LogicalResourceId

O nome do modelo escolhido pelo desenvolvedor (ID lógico) do recurso personalizado no modelo do CloudFormation.

Obrigatório: Sim

Tipo: string

ResourceProperties

Esse campo apresenta o conteúdo do objeto Properties enviado pelo desenvolvedor de modelos. Seu conteúdo é definido pelo provedor de recursos personalizados.

Obrigatório: não

Tipo: objeto JSON

Exemplo

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-create-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Update

Quando o desenvolvedor de modelos faz alterações nas propriedades de uma pilha que contém um recurso personalizado e atualiza a pilha, o CloudFormation envia uma solicitação ao provedor de recursos personalizados com RequestType definido como Update. Isso significa que o código do recurso personalizado não precisa detectar as alterações nos recursos, pois sabe que as propriedades foram alteradas quando o tipo de solicitação é Update.

Atualize solicitações que contenham os seguintes campos:

RequestType

Update.

Obrigatório: Sim

Tipo: string

RequestId

Um ID exclusivo para a solicitação.

Combinar `StackId` com `RequestId` forma um valor que é possível utilizar para identificar exclusivamente uma solicitação em um recurso personalizado específico.

Obrigatório: Sim

Tipo: string

StackId

O nome do recurso da Amazon (ARN) que identifica a pilha que contém o recurso personalizado.

Combinar `StackId` com `RequestId` forma um valor que é possível utilizar para identificar exclusivamente uma solicitação em um recurso personalizado específico.

Obrigatório: Sim

Tipo: string

ResponseURL

O URL de resposta identifica um bucket do S3 pré-assinado que recebe respostas do provedor de recursos personalizados para o CloudFormation.

Obrigatório: Sim

Tipo: string

ResourceType

O tipo de recurso escolhido pelo desenvolvedor do modelo do recurso personalizado no modelo do CloudFormation. Os nomes de tipos de recursos personalizados podem incluir até 60 caracteres, incluindo alfanuméricos e os seguintes caracteres: `_@-`. Não é possível alterar o tipo durante uma atualização.

Obrigatório: Sim

Tipo: string

LogicalResourceId

O nome do modelo escolhido pelo desenvolvedor (ID lógico) do recurso personalizado no modelo do CloudFormation.

Obrigatório: Sim

Tipo: string

PhysicalResourceId

Um ID físico definido por um provedor de recursos personalizados exclusivo para esse provedor.

Obrigatório: Sim

Tipo: string

ResourceProperties

Esse campo apresenta o conteúdo do objeto `Properties` enviado pelo desenvolvedor de modelos. Seu conteúdo é definido pelo provedor de recursos personalizados.

Obrigatório: não

Tipo: objeto JSON

OldResourceProperties

Usado apenas para solicitações `Update`. Os valores da propriedade do recurso que foram declarados anteriormente pelo desenvolvedor do modelo no modelo do CloudFormation.

Obrigatório: sim

Tipo: objeto JSON

Exemplo

```
{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
```

```
"ResponseURL" : "pre-signed-url-for-update-response",
"ResourceType" : "Custom::MyCustomResourceType",
"LogicalResourceId" : "resource-logical-id",
"PhysicalResourceId" : "provider-defined-physical-id",
"ResourceProperties" : {
  "key1" : "new-string",
  "key2" : [ "new-list" ],
  "key3" : { "key4" : "new-map" }
},
"OldResourceProperties" : {
  "key1" : "string",
  "key2" : [ "list" ],
  "key3" : { "key4" : "map" }
}
}
```

Delete

Quando o desenvolvedor de modelos exclui uma pilha ou remove um recurso personalizado do modelo, e depois atualiza a pilha, o CloudFormation envia uma solicitação com `RequestType` definido como `Delete`.

Exclua solicitações que contenha os seguintes campos:

RequestType

Delete.

Obrigatório: Sim

Tipo: string

RequestId

Um ID exclusivo para a solicitação.

Obrigatório: Sim

Tipo: string

StackId

O nome do recurso da Amazon (ARN) que identifica a pilha que contém o recurso personalizado.

Obrigatório: Sim

Tipo: string

ResponseURL

O URL de resposta identifica um bucket do S3 pré-assinado que recebe respostas do provedor de recursos personalizados para o CloudFormation.

Obrigatório: Sim

Tipo: string

ResourceType

O tipo de recurso escolhido pelo desenvolvedor do modelo do recurso personalizado no modelo do CloudFormation. Os nomes de tipos de recursos personalizados podem incluir até 60 caracteres, incluindo alfanuméricos e os seguintes caracteres: `_@-`.

Obrigatório: Sim

Tipo: string

LogicalResourceId

O nome do modelo escolhido pelo desenvolvedor (ID lógico) do recurso personalizado no modelo do CloudFormation.

Obrigatório: Sim

Tipo: string

PhysicalResourceId

Um ID físico definido por um provedor de recursos personalizados exclusivo para esse provedor.

Obrigatório: Sim

Tipo: string

ResourceProperties

Esse campo apresenta o conteúdo do objeto `Properties` enviado pelo desenvolvedor de modelos. Seu conteúdo é definido pelo provedor de recursos personalizados.

Obrigatório: não

Tipo: objeto JSON

Exemplo

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-delete-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Objeto da resposta

O provedor de recursos personalizados envia uma resposta para o URL pré-assinado para todos os tipos de solicitação. Se o provedor de recursos personalizados não enviar uma resposta, o CloudFormation vai esperar até que a operação atinja o tempo limite.

A resposta deve ser um objeto JSON com os seguintes campos:

Status

Deve ser SUCCESS ou FAILED.

Obrigatório: Sim

Tipo: string

RequestId

Um ID exclusivo para a solicitação. Copie esse valor exatamente como ele aparece na solicitação.

Obrigatório: Sim

Tipo: string

StackId

O nome do recurso da Amazon (ARN) que identifica a pilha que contém o recurso personalizado. Copie esse valor exatamente como ele aparece na solicitação.

Obrigatório: Sim

Tipo: string

LogicalResourceId

O nome do modelo escolhido pelo desenvolvedor (ID lógico) do recurso personalizado no modelo do CloudFormation. Copie esse valor exatamente como ele aparece na solicitação.

Obrigatório: Sim

Tipo: string

PhysicalResourceId

Esse valor deve ser um identificador exclusivo do fornecedor de recursos personalizados e pode ter até 1 KB de tamanho. O valor deve ser uma string não vazia e deve ser idêntico para todas as respostas do mesmo recurso.

Ao atualizar recursos personalizados, o valor retornado de `PhysicalResourceId` determina o comportamento da atualização. Se o valor retornado permanecer o mesmo, o CloudFormation vai considerar a atualização normal. Se o valor mudar, o CloudFormation vai interpretar a atualização como uma substituição e enviará uma solicitação de exclusão para o recurso antigo. Para obter mais informações, consulte [AWS::CloudFormation::CustomResource](#).

Obrigatório: Sim

Tipo: string

Reason

Descreve o motivo de uma resposta de falha.

Obrigatório se Status for FAILED. Caso contrário, será opcional.

Obrigatório: Condicional

Tipo: string

NoEcho

Indica se é necessário mascarar a saída do recurso personalizado quando ela for recuperada usando a função `Fn::GetAtt`. Se definido como `true`, todos os valores retornados serão mascarados com asteriscos (`*****`), exceto os armazenados na seção *Metadata* do modelo. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção *Metadata*. O valor padrão é `false`.

Para obter mais informações sobre como usar NoEcho para mascarar informações confidenciais, consulte a prática recomendada [Não incorporar credenciais em seus modelos](#).

Disponível somente para respostas `Create` e `Update`. Não é compatível com respostas `Delete`.

Obrigatório: não

Tipo: booleano

Data

Os pares de nome/valor definidos pelo provedor de recursos personalizados a serem enviados com a resposta. É possível acessar os valores fornecidos aqui pelo nome no modelo com `Fn::GetAtt`.

Disponível somente para respostas `Create` e `Update`. Não é compatível com respostas `Delete`.

Important

Se os pares de nome/valor contiverem informações confidenciais, você deverá utilizar o campo NoEcho para mascarar a saída do recurso personalizado. Caso contrário, os valores estarão visíveis por meio de APIs que exibem valores de propriedades (como `DescribeStackEvents`).

Obrigatório: não

Tipo: objeto JSON

Exemplos de respostas de sucesso

Resposta **Create** e **Update**

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "NoEcho": true,
  "Data": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

DeleteResposta do

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id"
}
```

Exemplo de resposta com falha

```
{
  "Status": "FAILED",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "Reason": "Required failure reason string"
}
```

Recursos personalizados baseados no Amazon SNS

O tópico apresentado a seguir mostra como configurar um recurso personalizado com um token de serviço que especifica o tópico do Amazon SNS para o qual o CloudFormation envia solicitações.

Além disso, você aprende a sequência de eventos e de mensagens enviadas e recebidas como resultado da criação, da atualização e da exclusão da pilha de recursos personalizada.

Com recursos personalizados e o Amazon SNS, você pode habilitar cenários como adicionar novos recursos a uma pilha e injetar dados dinâmicos em uma pilha. Por exemplo, quando você cria uma pilha, o CloudFormation pode enviar uma solicitação `Create` para um tópico monitorado por uma aplicação em execução em uma instância do Amazon EC2. A notificação do Amazon SNS aciona a aplicação para realizar tarefas de provisionamento adicionais, como recuperar um grupo de IPs elásticos na lista de permissão. Depois que todas essas ações forem concluídas, a aplicação enviará uma resposta (e quaisquer dados de saída) que notifica o CloudFormation para dar prosseguimento à operação de pilha.

Quando você especifica um tópico do Amazon SNS como o destino de um recurso personalizado, o CloudFormation envia mensagens para o tópico do SNS especificado durante as operações de pilha que envolvem o recurso personalizado. Para realizar o processamento dessas mensagens e executar as ações necessárias, você deve ter um endpoint com suporte que seja assinante do tópico do SNS.

Para obter uma introdução aos recursos personalizados e como eles funcionam, consulte [Como os recursos personalizados funcionam](#). Para obter mais informações sobre o Amazon SNS e como ele funciona, consulte o [Guia do desenvolvedor do Amazon Simple Notification Service](#).

Uso do Amazon SNS para criar recursos personalizados

Tópicos

- [Etapa 1: Criar a pilha](#)
- [Etapa 2: Atualizações de pilha](#)
- [Etapa 3: Exclusão da pilha](#)

Etapa 1: Criar a pilha

1. O desenvolvedor do modelo cria uma pilha do CloudFormation que contém um recurso personalizado.

No exemplo de modelo apresentado abaixo, usamos o nome do tipo de recurso personalizado `Custom::SeleniumTester` para o recurso personalizado com ID lógico `MySeleniumTest`. Os nomes de tipo de recurso personalizados devem ser alfanuméricos e podem ter um comprimento máximo de 60 caracteres.

O tipo de recurso personalizado é declarado com um token de serviço, propriedades opcionais específicas do provedor e atributos de [Fn::GetAtt](#) opcionais que são definidos pelo provedor de recursos personalizados. Essas propriedades e os atributos podem ser usados para passar informações do template developer para o custom resource provider, e vice-versa. O token de serviço especifica um tópico do Amazon SNS configurado pelo provedor de recursos.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
      "Type": "Custom::SeleniumTester",
      "Version" : "1.0",
      "Properties" : {
        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
"http://search.mysite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
      }
    }
  },
  "Outputs" : {
    "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    },
    "numRespondents" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
    }
  }
}
```

Note

Os nomes e os valores dos dados acessados com `Fn::GetAtt` são retornados pelo provedor de recursos personalizados durante a resposta do provedor para o CloudFormation. Caso o custom resource provider seja um terceiro, o template developer deverá obter os nomes desses valores de retorno do custom resource provider.

2. O CloudFormation envia uma notificação do Amazon SNS para o provedor de recursos com um "RequestType" : "Create" que contém informações sobre a pilha, as propriedades do recurso personalizado do modelo da pilha e um URL de S3 para a resposta.

O tópico SNS usado para enviar a notificação é incorporado no modelo na propriedade `ServiceToken`. Para evitar usar um valor codificado, um template developer pode usar um parâmetro de modelo, de maneira que o valor seja informado no momento em que a pilha é iniciada.

O exemplo a seguir mostra uma solicitação `Create` de recurso personalizado que inclui um nome de tipo de recurso personalizado `Custom::SeleniumTester`, criado com um `LogicalResourceId` de `MySeleniumTester`:

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}
```

Para obter informações detalhadas sobre o objeto de solicitação para as solicitações `Create`, consulte o tópico [Referência de solicitações e respostas](#).

3. O custom resource provider processa os dados enviados pelo template developer e determina se a solicitação `Create` foi bem-sucedida. O provedor de recursos usa o URL do S3 enviado pelo CloudFormation para enviar uma resposta de `SUCCESS` ou de `FAILED`.

O CloudFormation espera diferentes campos de resposta com base no tipo de resposta. Para obter informações sobre os campos de resposta para um tipo de solicitação em específico, consulte a documentação para esse tipo de solicitação na seção [Referência de solicitações e respostas](#).

Em resposta a uma solicitação de criação ou atualização, o provedor personalizado de recursos pode retornar elementos de dados no campo `Data` da resposta. Esses são os pares de nome/valor, e os nomes correspondem aos atributos `Fn::GetAtt` usados com o recurso personalizado no modelo de pilha. Os valores são os dados retornados quando o `Template Developer` chama `Fn::GetAtt` no recurso com o nome do atributo.

Este é um exemplo de uma resposta de recurso personalizado:

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "Data" : {
    "resultsPage" : "http://www.myexampledomain/test-results/guid",
    "lastUpdate" : "2012-11-14T03:30Z"
  }
}
```

Para obter informações detalhadas sobre o objeto de resposta para as solicitações `Create`, consulte o tópico [Referência de solicitações e respostas](#).

Os campos `StackId`, `RequestId` e `LogicalResourceId` devem ser copiados na íntegra da solicitação.

4. O CloudFormation declara o status da pilha como `CREATE_COMPLETE` ou `CREATE_FAILED`. Caso a pilha tenha sido criada com êxito, o desenvolvedor de modelos pode usar os valores de saída do recurso personalizado criado acessando-os com [Fn::GetAtt](#).

Por exemplo, o modelo de recurso personalizado usado na ilustração utilizou `Fn::GetAtt` para copiar saídas de recurso para as saídas de pilha:

```
"Outputs" : {
  "topItem" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
  },
  "numRespondents" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
  }
}
```

```
}
```

Etapa 2: Atualizações de pilha

Para atualizar uma pilha existente, é necessário enviar um modelo que especifique as atualizações para as propriedades dos recursos na pilha, conforme mostrado no exemplo abaixo. O CloudFormation atualiza somente os recursos que têm alterações especificadas no modelo. Para obter mais informações, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

É possível atualizar recursos personalizados que exigem uma substituição do recurso físico subjacente. Quando você atualiza um recurso personalizado em um modelo do CloudFormation, o CloudFormation envia uma solicitação de atualização a esse recurso personalizado. Caso um recurso personalizado exija uma substituição, o novo recurso personalizado deve enviar uma resposta com o novo ID físico. Quando o CloudFormation recebe a resposta, ele compara o `PhysicalResourceId` entre os recursos personalizados antigos e novos. Se os recursos forem diferentes, o CloudFormation reconhecerá a atualização como uma substituição e enviará uma solicitação de exclusão para o recurso antigo, conforme mostrado na [Etapa 3: Exclusão da pilha](#).

Note

Se você não fez alterações no recurso personalizado, o CloudFormation não enviará solicitações para ele durante uma atualização de pilha.

1. O template developer inicia uma atualização para a pilha que contém um recurso personalizado. Durante uma atualização, o template developer pode especificar novas propriedades no modelo da pilha.

Este é um exemplo de um Update para o modelo de pilha que usa um tipo de recurso personalizado:

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
      "Type": "Custom::SeleniumTester",
      "Version" : "1.0",
      "Properties" : {
```

```

        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
            "http://search.mysite.com",
            "http://mynewsite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
    }
},
"Outputs" : {
    "topItem" : {
        "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    },
    "numRespondents" : {
        "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
    }
}
}

```

2. O CloudFormation envia uma notificação do Amazon SNS para o provedor de recursos com um "RequestType" : "Update" que contém informações semelhantes às da chamada Create, exceto por o campo OldResourceProperties conter as propriedades de recurso anteriores e ResourceProperties conter as propriedades de recurso atualizadas (caso haja alguma).

Este é um exemplo de uma solicitação Update:

```

{
    "RequestType" : "Update",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
    "ResponseURL" : "http://pre-signed-S3-url-for-response",
    "ResourceType" : "Custom::SeleniumTester",
    "LogicalResourceId" : "MySeleniumTester",
    "PhysicalResourceId" : "Tester1",
    "ResourceProperties" : {
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
            "http://mynewsite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
    },
    "OldResourceProperties" : {

```



```
"seleniumTester" : "SeleniumTest()",
"endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
"frequencyOfTestsPerHour" : [ "3", "2", "4" ]
}
}
```

Para obter informações detalhadas sobre o objeto de solicitação para as solicitações Update, consulte o tópico [Referência de solicitações e respostas](#).

3. O provedor de recursos personalizados realiza o processamento dos dados enviados pelo CloudFormation. O recurso personalizado executa a atualização e envia uma resposta de SUCCESS ou de FAILED para o URL do S3. Em seguida, o CloudFormation compara os PhysicalResourceIDs de recursos personalizados antigos e novos. Se os recursos forem diferentes, o CloudFormation reconhecerá que a atualização requer uma substituição e enviará uma solicitação de exclusão para o recurso antigo. O exemplo a seguir demonstra a resposta do custom resource provider a uma solicitação Update.

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester2"
}
```

Para obter informações detalhadas sobre o objeto de resposta para as solicitações Update, consulte o tópico [Referência de solicitações e respostas](#).

Os campos StackId, RequestId e LogicalResourceId devem ser copiados na íntegra da solicitação.

4. O CloudFormation declara o status da pilha como UPDATE_COMPLETE ou UPDATE_FAILED. Em caso de falha na atualização, a pilha é revertida. Caso a pilha tenha sido atualizada com êxito, o template developer poderá acessar todos os novos valores de saída do recurso personalizado criado com Fn::GetAtt.

Etapa 3: Exclusão da pilha

1. O desenvolvedor do modelo exclui uma pilha que contém um recurso personalizado. O CloudFormation obtém as propriedades atuais especificadas no modelo de pilha em conjunto com o tópico do SNS e se prepara para fazer uma solicitação ao provedor de recursos personalizados.
2. O CloudFormation envia uma notificação do Amazon SNS para o provedor de recursos com um "RequestType" : "Delete" que contém informações atuais sobre a pilha, as propriedades do recurso personalizado do modelo da pilha e um URL de S3 para a resposta.

Sempre que você exclui uma pilha ou faz uma atualização que remove ou substitui o recurso personalizado, o CloudFormation compara o `PhysicalResourceId` entre os recursos personalizados antigos e novos. Se os recursos forem diferentes, o CloudFormation reconhecerá a atualização como uma substituição e enviará uma solicitação de exclusão para o recurso antigo (`OldPhysicalResource`), conforme mostrado no exemplo a seguir de uma solicitação `Delete`.

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com",
      "http://mynewsite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
  }
}
```

Para obter informações detalhadas sobre o objeto de solicitação para as solicitações `Delete`, consulte o tópico [Referência de solicitações e respostas](#).

`DescribeStackResource`, `DescribeStackResource` e `ListStackResources` exibem o nome definido pelo usuário caso ele tenha sido especificado.

3. O provedor de recursos personalizados realiza o processamento dos dados enviados pelo CloudFormation e determina se a solicitação `Delete` ocorreu com êxito. O provedor de recursos

usa o URL do S3 enviado pelo CloudFormation para enviar uma resposta de SUCCESS ou de FAILED. Para excluir com êxito uma pilha com um recurso personalizado, o custom resource provider devem responder com êxito a uma solicitação de exclusão.

Este é um exemplo de uma resposta do custom resource provider a uma solicitação Delete:

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1"
}
```

Para obter informações detalhadas sobre o objeto de resposta para as solicitações Delete, consulte o tópico [Referência de solicitações e respostas](#).

Os campos StackId, RequestId e LogicalResourceId devem ser copiados na íntegra da solicitação.

4. O CloudFormation declara o status da pilha como DELETE_COMPLETE ou DELETE_FAILED.

Recursos personalizados baseados no Lambda

Ao associar uma função do Lambda com um recurso personalizado, a função é invocada sempre que o recurso personalizado é criado, atualizado ou excluído. O CloudFormation chama uma API do Lambda para invocar a função e transferir todos os dados da solicitação (como o tipo de solicitação e as propriedades do recurso) para a função. A capacidade e a personalização das funções do Lambda em associação com o CloudFormation possibilitam uma ampla variedade de cenários, como a pesquisa dinâmica de IDs de AMI durante a criação da pilha ou a implementação e o uso de funções utilitárias, como as funções de reversão de string.

Para obter uma introdução aos recursos personalizados e como eles funcionam, consulte [Como os recursos personalizados funcionam](#).

Tópicos

- [Instruções passo a passo: criar um mecanismo de atraso com um recurso personalizado apoiado pelo Lambda](#)

- [cfn-responseMódulo](#)

Instruções passo a passo: criar um mecanismo de atraso com um recurso personalizado apoiado pelo Lambda

Este passo a passo mostra como configurar e executar um recurso personalizado com suporte do Lambda usando um modelo de exemplo do CloudFormation. Este modelo cria um mecanismo de atraso que pausa as implantações de pilha por um período especificado. Isso pode ser útil quando você precisa introduzir atrasos deliberados durante o provisionamento de recursos, como ao esperar que os recursos se estabilizem antes que os recursos dependentes sejam criados.

 Note

Embora os recursos personalizados baseados em Lambda tenham sido recomendados anteriormente para recuperar IDs de AMI, recomendamos utilizar parâmetros do AWS Systems Manager. Essa abordagem torna seus modelos mais reutilizáveis e fáceis de manter. Para obter mais informações, consulte [Obter um valor em texto simples do Systems Manager Parameter Store](#).

Tópicos

- [Visão geral](#)
- [Modelo de exemplo](#)
- [Modelo de passo a passo](#)
- [Pré-requisitos](#)
- [Iniciar a pilha](#)
- [Limpar os recursos](#)
- [Informações relacionadas](#)

Visão geral

O modelo de pilha de exemplo usado neste passo a passo cria um recurso personalizado baseado no Lambda. Esse recurso personalizado introduz um atraso configurável (60 segundos por padrão) durante a criação da pilha. O atraso ocorre durante as atualizações da pilha somente quando as propriedades do recurso personalizado são modificadas.

Este modelo fornece os seguintes recursos:

- um recurso personalizado,
- uma função do Lambda e
- um perfil do IAM que permite que o Lambda grave logs no CloudWatch.

Ele também define duas saídas:

- O tempo que a função de fato esperou.
- Um identificador único gerado durante cada execução da função do Lambda.

Note

O CloudFormation é um serviço gratuito, mas o Lambda cobra com base no número de solicitações para suas funções e no tempo durante o qual seu código é executado. Para obter mais informações sobre preços do Lambda, consulte [Preços do AWS Lambda](#).

Modelo de exemplo

É possível ver o modelo de amostra de recurso personalizado apoiado pelo Lambda com o mecanismo de atraso abaixo:

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
          }]
        },
        "Path": "/",
```

```

    "Policies": [{
      "PolicyName": "AllowLogs",
      "PolicyDocument": {
        "Statement": [{
          "Effect": "Allow",
          "Action": ["logs:*"],
          "Resource": "*"
        }]
      }
    }],
  },
  "CFNWaiter": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Runtime": "python3.9",
      "Timeout": 900,
      "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
      "Code": {
        "ZipFile": { "Fn::Join": ["\n", [
          "from time import sleep",
          "import json",
          "import cfnresponse",
          "import uuid",
          "",
          "def handler(event, context):",
          "    wait_seconds = 0",
          "    id = str(uuid.uuid1())",
          "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
          "        wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout", 0))",
          "        sleep(wait_seconds)",
          "        response = {",
          "            \"TimeWaited\": wait_seconds,",
          "            \"Id\": id ",
          "        }",
          "        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,",
          "\"Waiter-\"+id)"]
        ]}
      }
    }
  },
  "CFNWaiterCustomResource": {

```

```

    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  },
  "Outputs": {
    "TimeWaited": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
      "Export": { "Name": "TimeWaited" }
    },
    "WaiterId": {
      "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
      "Export": { "Name": "WaiterId" }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: "2010-09-09"
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"

```

```
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile:
        !Sub |
        from time import sleep
        import json
        import cfnresponse
        import uuid

        def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
                wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
                sleep(wait_seconds)
            response = {
                "TimeWaited": wait_seconds,
                "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    ServiceTimeout: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
```


Modelo de passo a passo

Os trechos a seguir explicam partes relevantes do modelo de amostra para ajudá-lo a entender como a função do Lambda está associada a um recurso personalizado e a entender a saída.

CFNWaiter do resource [AWS::Lambda::Function](#)

O recurso `AWS::Lambda::Function` especifica o código-fonte, o nome do handler, o ambiente de runtime e o nome do recurso da Amazon (ARN) do perfil de execução da função.

A propriedade `Handler` é definida como `index.handler`, pois usa um código-fonte do Python. Para mais informações sobre identificadores de manipuladores aceitos ao usar códigos-fonte de funções embutidas, consulte [AWS::Lambda::Function Code](#).

O `Runtime` é especificado como `python3.9` porque o arquivo de origem é um código do Python.

O `Timeout` é definido como 900 segundos.

A propriedade `Role` usa a função `Fn::GetAtt` para obter o ARN do perfil de execução `LambdaExecutionRole` declarado no recurso `AWS::IAM::Role` do modelo.

A propriedade `Code` define o código da função inline com uma função do Python. A função do Python no modelo de amostra faz o seguinte:

- Criar um ID exclusivo utilizando o UUID
- Verificar se a solicitação é de criação ou atualização
- Suspende pela duração especificada por `ServiceTimeout` durante solicitações `Create` ou `Update`
- Retorna o tempo de espera e o ID exclusivo

JSON

```
...
  "CFNWaiter": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Runtime": "python3.9",
      "Timeout": 900,
      "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
```

```

"Code": {
  "ZipFile": { "Fn::Join": ["\n", [
    "from time import sleep",
    "import json",
    "import cfnresponse",
    "import uuid",
    "",
    "def handler(event, context):",
    "  wait_seconds = 0",
    "  id = str(uuid.uuid1())",
    "  if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
    "    wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout", 0))",
    "    sleep(wait_seconds)",
    "    response = {",
    "      \"TimeWaited\": wait_seconds,",
    "      \"Id\": id ",
    "    }",
    "    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,",
    "\"Waiter-\"+id)"
  ]]}
}
},
},
...

```

YAML

```

...
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile:
        !Sub |
          from time import sleep
          import json
          import cfnresponse
          import uuid

```

```

def handler(event, context):
    wait_seconds = 0
    id = str(uuid.uuid1())
    if event["RequestType"] in ["Create", "Update"]:
        wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
        sleep(wait_seconds)
    response = {
        "TimeWaited": wait_seconds,
        "Id": id
    }
    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
...

```

LambdaExecutionRole do recurso [AWS::IAM::Role](#)

O recurso `AWS::IAM::Role` cria um perfil de execução para a função do Lambda, que inclui uma política de função de suposição que permite que o Lambda a use. Ele também contém uma política que permite o acesso aos logs do CloudWatch.

JSON

```

...
"LambdaExecutionRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Statement": [{
        "Effect": "Allow",
        "Principal": { "Service": ["lambda.amazonaws.com"] },
        "Action": ["sts:AssumeRole"]
      }]
    },
    "Path": "/",
    "Policies": [{
      "PolicyName": "AllowLogs",
      "PolicyDocument": {
        "Statement": [{
          "Effect": "Allow",
          "Action": ["logs:*"],
          "Resource": "*"
        }]
      }
    }]
  }
}

```

```

    ...
  },
}
}]
}
}]

```

YAML

```

...
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "lambda.amazonaws.com"
          Action:
            - "sts:AssumeRole"
    Path: "/"
    Policies:
      - PolicyName: "AllowLogs"
        PolicyDocument:
          Statement:
            - Effect: "Allow"
              Action:
                - "logs:*"
              Resource: "*"

```

Recurso [AWS::CloudFormation::CustomResource](#) CFNWaiterCustomResource

O recurso personalizado se vincula à função do Lambda com seu ARN usando `!GetAtt CFNWaiter.Arn`. Ela implementará um tempo de espera de 60 segundos para operações de criação e atualização, conforme definido em `ServiceTimeout`. O recurso somente será chamado para uma operação de atualização se as propriedades forem modificadas.

JSON

```
...
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  },
  ...
```

YAML

```
...
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    ServiceTimeout: 60
...
```

Outputs

As Outputs desse modelo são TimeWaited e WaiterId. O valor TimeWaited usa uma função Fn::GetAtt para fornecer a quantidade de tempo que o recurso de espera realmente aguardou. WaiterId usa uma função Fn::GetAtt para fornecer o ID exclusivo que foi gerado e associado à execução.

JSON

```
...
"Outputs": {
  "TimeWaited": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
    "Export": { "Name": "TimeWaited" }
  },
  "WaiterId": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
    "Export": { "Name": "WaiterId" }
  }
}
```

```
    }  
  }  
}  
...
```

YAML

```
...  
Outputs:  
  TimeWaited:  
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited  
    Export:  
      Name: TimeWaited  
  WaiterId:  
    Value: !GetAtt CFNWaiterCustomResource.Id  
    Export:  
      Name: WaiterId  
...
```

Pré-requisitos

Você deve ter permissões do IAM para usar todos os serviços correspondentes, como o Lambda e o CloudFormation.

Iniciar a pilha

Para criar a pilha

1. Encontre o modelo de sua preferência (YAML ou JSON) na seção [Modelo de exemplo](#) e salve-o na máquina com o nome `samplelambdabackedcustomresource.template`.
2. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation/>.
3. Na página Pilhas, escolha Criar pilha no canto superior direito e depois Com novos recursos (padrão).
4. Em Pré-requisito: preparar modelo, escolha Escolher um modelo existente.
5. Para Especificar modelo, selecione Carregar um arquivo de modelo e depois Escolher arquivo.
6. Selecione o arquivo de modelo `samplelambdabackedcustomresource.template` salvou anteriormente.
7. Escolha Próximo.

8. Para Nome da pilha, digite **SampleCustomResourceStack** e escolha Próximo.
9. Para esta demonstração, você não precisa adicionar tags ou especificar configurações avançadas, portanto escolha Próximo.
10. Certifique-se de que o nome da pilha esteja correto e escolha Criar.

Pode levar alguns minutos para que o CloudFormation crie a pilha. Para monitorar o progresso, visualize os eventos da pilha. Para obter mais informações, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Se a criação da pilha for bem-sucedida, todos os recursos da pilha, como a função do Lambda e o recurso personalizado, foram criados. Você usou com êxito uma função do Lambda e um recurso personalizado.

Se a função do Lambda retornar um erro, visualize os logs da função no [console](#) do CloudWatch Logs. O nome do stream de logs é o ID físico do recurso personalizado, que pode ser encontrado visualizando os recursos da pilha. Para obter mais informações, consulte [Visualizar dados de log](#), no Guia do usuário do Amazon CloudWatch.

Limpar os recursos

Exclua a pilha para limpar todos os recursos de pilha criados; assim, você não será cobrado por recursos desnecessários.

Para excluir a pilha

1. No console do CloudFormation, escolha a pilha SampleCustomResourceStack.
2. Escolha Ações e, em seguida, Excluir pilha.
3. Na mensagem de confirmação, escolha Sim, excluir.

Todos os recursos que você criou serão excluídos.

Agora que você sabe como criar e usar o recurso personalizado apoiado pelo Lambda, pode usar o modelo de amostra e o código deste passo a passo para criar e experimentar outras pilhas e funções.

Informações relacionadas

- [Referência de recursos personalizados do CloudFormation](#)

- [AWS::CloudFormation::CustomResource](#)

cfn-responseMódulo

No modelo do CloudFormation, é possível especificar uma função do Lambda como o destino de um recurso personalizado. Ao usar a propriedade `ZipFile` para especificar o código-fonte da [função](#), você pode carregar o módulo `cfn-response` com a finalidade de enviar respostas da função do Lambda para um recurso personalizado. O módulo `cfn-response` corresponde a uma biblioteca que simplifica o envio de respostas para o recurso personalizado que invocou a função do Lambda. O módulo contém um método `send` que envia um [objeto de resposta](#) para um recurso personalizado por meio de um URL do Amazon S3 assinado previamente (o `ResponseURL`).

O módulo `cfn-response` está disponível somente quando você usa a propriedade `ZipFile` para gravar o código-fonte. Ele não está disponível para o código-fonte armazenado nos buckets do Amazon S3. Para código em buckets do , você deve criar suas próprias funções para enviar respostas.

Note

Depois de executar o método `send` a função do Lambda é encerrada, assim qualquer coisa que você grave após esse método é ignorada.

Carregar o módulo **cfn-response**

Para funções Node.js, use a função `require()` para carregar o módulo `cfn-response`. Para esclarecer, o código de exemplo a seguir cria um objeto `cfn-response` com o nome `response`:

```
var response = require('cfn-response');
```

Para Python, use a instrução `import` para carregar o módulo `cfnresponse`, conforme mostrado no seguinte exemplo:

Note

Use essa exata declaração de importação. Se você usar outras variantes da instrução “`import`”, o CloudFormation não incluirá o módulo de resposta.


```
import cfnresponse
```

sendParâmetros do método

Você pode usar os seguintes parâmetros com o método `send`.

`event`

Os campos em uma [solicitação de recurso personalizado](#).

`context`

Um objeto, específico às funções do Lambda que você pode usar para especificar quando a função e qualquer retorno de chamada tiverem concluído a execução ou para acessar informações no ambiente de execução do Lambda. Para obter mais informações, consulte [Construir funções do Lambda com Node.js](#) no Guia do desenvolvedor do AWS Lambda.

`responseStatus`

Se a função foi concluída com êxito. Use as constantes do módulo `cfnresponse` para especificar o status: `SUCCESS` para execuções bem-sucedidas e `FAILED` para execuções com falha.

`responseData`

O campo `Data` de um [objeto de resposta](#) de um recurso personalizado. Os dados são uma lista de pares de nome-valor.

`physicalResourceId`

Opcional. O identificador exclusivo do recurso personalizado que invocou a função. Por padrão, o módulo usa o nome do fluxo de log do Amazon CloudWatch Logs que está associado à função do Lambda.

O valor retornado para um `PhysicalResourceId` pode alterar as operações de atualização de recursos personalizados. Se o valor retornado for o mesmo, a atualização é considerada normal. Se o valor retornado for diferente, o CloudFormation reconhecerá a atualização como uma substituição e enviará uma solicitação de exclusão para o recurso antigo. Para obter mais informações, consulte [AWS::CloudFormation::CustomResource](#).

`noEcho`

Opcional. Indica se é necessário mascarar a saída do recurso personalizado quando ela for recuperada usando a função `Fn::GetAtt`. Se definido como `true`, todos os valores

retornados serão mascarados com asteriscos (****), exceto informações armazenadas nos locais especificados abaixo. Por padrão, esse valor é `false`.

Important

O uso do atributo `NoEcho` não mascara informações armazenadas no seguinte:

- A seção de modelo de Metadata. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção Metadata. Para obter mais informações, consulte [Metadados](#).
- A seção de modelo de Outputs. Para obter mais informações, consulte [Saídas](#).
- O atributo Metadata de uma definição de recurso. Para obter mais informações, consulte o atributo [Metadata](#).

É altamente recomendável não usar esses mecanismos para incluir informações confidenciais, como senhas ou segredos.

Para obter mais informações sobre como usar `NoEcho` para mascarar informações confidenciais, consulte a prática recomendada [Não incorporar credenciais em seus modelos](#).

Exemplos

Node.js

No exemplo Node.js a seguir, a função do Lambda incorporada obtém um valor de entrada e multiplica por 5. As funções em linha são especialmente úteis para funções menores, pois permitem que você especifique o código-fonte diretamente no modelo, em vez de criar um pacote e carregá-lo em um bucket do Amazon S3. A função usa o método `cfn-response send` para enviar o resultado de volta para o recurso personalizado que a chamou.

JSON

```
"ZipFile": { "Fn::Join": ["", [
  "var response = require('cfn-response');",
  "exports.handler = function(event, context) {",
  "  var input = parseInt(event.ResourceProperties.Input);",
  "  var responseData = {Value: input * 5};",
  "  response.send(event, context, response.SUCCESS, responseData);",
  "};"
```

```
]]}
```

YAML

```
ZipFile: >
  var response = require('cfn-response');
  exports.handler = function(event, context) {
    var input = parseInt(event.ResourceProperties.Input);
    var responseData = {Value: input * 5};
    response.send(event, context, response.SUCCESS, responseData);
  };
```

Python

No exemplo de Python a seguir, a função do Lambda em linha pega um valor inteiro e o multiplica por 5.

JSON

```
"ZipFile" : { "Fn::Join" : ["\n", [
  "import json",
  "import cfnresponse",
  "def handler(event, context):",
  "   responseValue = int(event['ResourceProperties']['Input']) * 5",
  "   responseData = {}",
  "   responseData['Data'] = responseValue",
  "   cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,",
  "\"CustomResourcePhysicalID\"")
]]}
```

YAML

```
ZipFile: |
  import json
  import cfnresponse
  def handler(event, context):
    responseValue = int(event['ResourceProperties']['Input']) * 5
    responseData = {}
    responseData['Data'] = responseValue
    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
"CustomResourcePhysicalID")
```

Código-fonte do módulo

Tópicos

- [Código-fonte assíncrono do Node.js](#)
- [Código fonte do Node.js](#)
- [Código-fonte Python](#)

Código-fonte assíncrono do Node.js

Veja a seguir o código-fonte do módulo de resposta das funções do Node.js caso o handler seja assíncrono. Analise-o para entender o que o módulo faz e obter ajuda com a implementação de suas próprias funções de resposta.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
            Data: responseData
        });

        console.log("Response body:\n", responseBody);

        var https = require("https");
        var url = require("url");

        var parsedUrl = url.parse(event.ResponseURL);
```

```

    var options = {
      hostname: parsedUrl.hostname,
      port: 443,
      path: parsedUrl.path,
      method: "PUT",
      headers: {
        "content-type": "",
        "content-length": responseBody.length
      }
    };

    var request = https.request(options, function(response) {
      console.log("Status code: " + parseInt(response.statusCode));
      resolve(context.done());
    });

    request.on("error", function(error) {
      console.log("send(..) failed executing https.request(..): " +
        maskCredentialsAndSignature(error));
      reject(context.done(error));
    });

    request.write(responseBody);
    request.end();
  })
}

function maskCredentialsAndSignature(message) {
  return message.replace(/X-Amz-Credential=[^&\s]+/i, 'X-Amz-Credential=*****')
    .replace(/X-Amz-Signature=[^&\s]+/i, 'X-Amz-Signature=*****');
}

```

Código fonte do Node.js

Veja a seguir o código-fonte do módulo de resposta das funções do Node.js caso o handler não seja assíncrono. Analise-o para entender o que o módulo faz e obter ajuda com a implementação de suas próprias funções de resposta.

```

// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

```

```
exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream: " + context.logStreamName,
        PhysicalResourceId: physicalResourceId || context.logStreamName,
        StackId: event.StackId,
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        NoEcho: noEcho || false,
        Data: responseData
    });

    console.log("Response body:\n", responseBody);

    var https = require("https");
    var url = require("url");

    var parsedUrl = url.parse(event.ResponseURL);
    var options = {
        hostname: parsedUrl.hostname,
        port: 443,
        path: parsedUrl.path,
        method: "PUT",
        headers: {
            "content-type": "",
            "content-length": responseBody.length
        }
    };

    var request = https.request(options, function(response) {
        console.log("Status code: " + parseInt(response.statusCode));
        context.done();
    });

    request.on("error", function(error) {
        console.log("send(..) failed executing https.request(..): " +
            maskCredentialsAndSignature(error));
        context.done();
    });

    request.write(responseBody);
```

```
request.end();  
}
```

Código-fonte Python

Veja a seguir o código-fonte do módulo de resposta das funções do Python:

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.  
# SPDX-License-Identifier: MIT-0  
  
from __future__ import print_function  
import urllib3  
import json  
import re  
  
SUCCESS = "SUCCESS"  
FAILED = "FAILED"  
  
http = urllib3.PoolManager()  
  
def send(event, context, responseStatus, responseData, physicalResourceId=None,  
        noEcho=False, reason=None):  
    responseUrl = event['ResponseURL']  
  
    responseBody = {  
        'Status' : responseStatus,  
        'Reason' : reason or "See the details in CloudWatch Log Stream:  
{}".format(context.log_stream_name),  
        'PhysicalResourceId' : physicalResourceId or context.log_stream_name,  
        'StackId' : event['StackId'],  
        'RequestId' : event['RequestId'],  
        'LogicalResourceId' : event['LogicalResourceId'],  
        'NoEcho' : noEcho,  
        'Data' : responseData  
    }  
  
    json_responseBody = json.dumps(responseBody)  
  
    print("Response body:")  
    print(json_responseBody)  
  
    headers = {  
        'content-type' : '',
```

```
        'content-length' : str(len(json_responseBody))
    }

    try:
        response = http.request('PUT', responseUrl, headers=headers,
body=json_responseBody)
        print("Status code:", response.status)

    except Exception as e:

        print("send(..) failed executing http.request(..):",
mask_credentials_and_signature(e))

def mask_credentials_and_signature(message):
    message = re.sub(r'X-Amz-Credential=[^\s]+', 'X-Amz-Credential=*****', message,
flags=re.IGNORECASE)
    return re.sub(r'X-Amz-Signature=[^\s]+', 'X-Amz-Signature=*****', message,
flags=re.IGNORECASE)
```

Executar processamento personalizado em modelos do CloudFormation com macros de modelo

As macros permitem realizar o processamento personalizado em modelos, desde ações simples, como operações de localizar e substituir, até transformações extensas de modelos inteiros.

Para ter uma ideia das possibilidades, considere as transformações `AWS::Include` e `AWS::Serverless`, ambas macros hospedadas pelo CloudFormation:

- A [transformação `AWS::Include`](#) permite que você insira trechos de modelos clichê em seus modelos.
- A [transformação `AWS::Serverless`](#) usa um modelo inteiro escrito na sintaxe do AWS Serverless Application Model (AWS SAM) e o transforma e expande em um modelo compatível com o CloudFormation. Para obter mais informações sobre aplicações sem servidor e o AWS SAM, consulte o [Guia do desenvolvedor do AWS Serverless Application Model](#).

Tópicos

- [Faturamento](#)

- [Exemplos de macro](#)
- [Recursos relacionados](#)
- [Visão geral das macros do CloudFormation](#)
- [Criar uma definição de macro do CloudFormation](#)
- [Exemplo de macro simples de substituição de string](#)
- [Solucionar problemas do modelo processado](#)

Faturamento

Quando uma macro é executada, o proprietário da função Lambda é cobrado por todos os custos relacionados à execução dessa função.

As transformações `AWS::Include` e `AWS::Serverless` são macros hospedadas pelo CloudFormation. Não há custo pelo uso desses recursos.

Exemplos de macro

Além dos exemplos desta seção, você pode encontrar exemplos de macro, incluindo código-fonte e modelos, em nosso [repositório do GitHub](#). Esses exemplos são fornecidos "como estão" para fins de instrução.

Recursos relacionados

- [AWS::CloudFormation::Macro](#)
- [Seção Transform do modelo do CloudFormation](#)
- [Fn::Transform](#)
- [AWS::Serverless Transformação](#)
- [AWS::Include Transformação](#)

Visão geral das macros do CloudFormation

Há duas etapas principais para processar modelos usando macros: criar a macro propriamente dita e, em seguida, usar essa macro para realizar o processamento nos seus modelos.

Para criar uma definição de macro, você deve criar o seguinte:

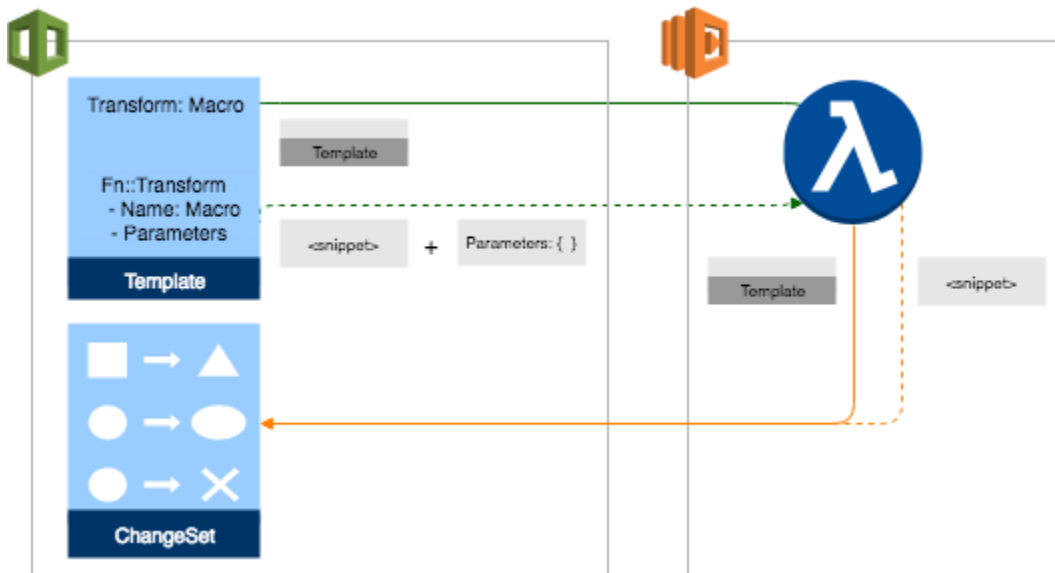
- Uma função do Lambda para processar o modelo. Essa função Lambda aceita um trecho inteiro ou um modelo inteiro e quaisquer parâmetros adicionais que você definir. Ele retorna o trecho de modelo processado ou todo o modelo como uma resposta.
- Um recurso do tipo [AWS::CloudFormation::Macro](#), que permite que os usuários chamem a função do Lambda de dentro de modelos do CloudFormation. Esse recurso especifica o ARN da função Lambda a ser chamada para esta macro, além de propriedades opcionais adicionais para ajudar na depuração. Para criar esse recurso dentro de uma conta, crie um modelo de pilha que inclua o recurso `AWS::CloudFormation::Macro` e, depois, crie uma pilha ou um conjunto de pilhas com permissões autogerenciadas por meio do modelo. CloudFormation No momento, o StackSets não suporta a criação ou atualização de conjuntos de pilhas com permissões gerenciadas por serviço de modelos que fazem referência a macros do .

Para usar uma macro, faça referência a ela no seu modelo:

- Para processar uma seção, ou parte, de um modelo, referencie a macro em uma função `Fn::Transform` localizada em relação ao conteúdo do modelo que você deseja transformar. Ao usar `Fn::Transform`, você também pode transmitir os parâmetros especificados.
- Para processar um modelo inteiro, faça referência à macro na seção [Transform](#) do modelo.

Depois, normalmente você cria um conjunto de alterações e o executa. (O processamento de macros pode adicionar vários recursos dos quais você pode não estar ciente. Para garantir que você conheça todas as alterações apresentadas pelas macros, é altamente recomendável usar conjuntos de alterações.) O CloudFormation transmite o conteúdo do modelo especificado, além de qualquer outro parâmetro especificado, para a função do Lambda especificada no recurso de macro. A função Lambda retorna o conteúdo do modelo processado, seja um trecho ou um modelo inteiro.

Depois que todas as macros do modelo tiverem sido chamadas, o CloudFormation gerará um conjunto de alterações que inclui o conteúdo do modelo processado. Após revisar o conjunto de alterações, execute-o para aplicar as alterações.



Como criar pilhas diretamente

Para criar ou atualizar uma pilha usando um modelo que faz referência a macros, normalmente você cria um conjunto de alterações e o executa. Um conjunto de alterações descreve as ações que o CloudFormation executará com base no modelo processado. O processamento de macros pode adicionar vários recursos dos quais você não tem conhecimento. Para garantir que você esteja ciente de todas as alterações apresentadas pelas macros, sugerimos o uso de conjuntos de alterações. Após revisar o conjunto de alterações, você poderá executá-lo para realmente aplicar as alterações.

Uma macro pode adicionar recursos do IAM a seu modelo. Para esses recursos, o CloudFormation requer que você [confirme suas funcionalidades](#). Como o CloudFormation não pode saber quais recursos são adicionados antes de processar o modelo, pode ser necessário confirmar as funcionalidades do IAM ao criar o conjunto de alterações, dependendo de as macros referenciadas incluírem ou não recursos do IAM. Dessa forma, quando você executar o conjunto de alterações, o CloudFormation terá os recursos necessários ao criar recursos do IAM.

Para criar ou atualizar uma pilha diretamente de um modelo processado sem primeiro analisar as alterações propostas em um conjunto de alterações, especifique o recurso `CAPABILITY_AUTO_EXPAND` durante uma solicitação de `CreateStack` ou `UpdateStack`. Você só deverá criar pilhas diretamente de um modelo de pilha que contém macros se souber o que o processamento da macro executa. Você não pode usar conjuntos de alterações com macros de conjunto de pilhas e deve atualizar seu conjunto de pilhas diretamente.

Para obter mais informações, consulte [CreateStack](#) ou [UpdateStack](#) na Referência de APIs do AWS CloudFormation.

 Important

Se o modelo do conjunto de pilhas fizer referência a uma ou mais macros, você deverá criar o conjunto de pilhas diretamente do modelo processado, sem primeiro revisar as alterações resultantes em um conjunto de alterações. O processamento de macros pode adicionar vários recursos dos quais você não tem conhecimento. Antes de criar ou atualizar um conjunto de pilhas de um modelo que faça referência a macros diretamente, certifique-se de saber qual processamento as macros executam.

Para reduzir o número de etapas para iniciar pilhas a partir de modelos que fazem referenciam macros, você pode usar os comandos `package` e `deploy` da AWS CLI. Para obter mais informações, consulte [Fazer upload de artefatos locais em um bucket do S3 com a AWS CLI](#) e [Criar uma pilha que inclua transformações](#).

Considerações

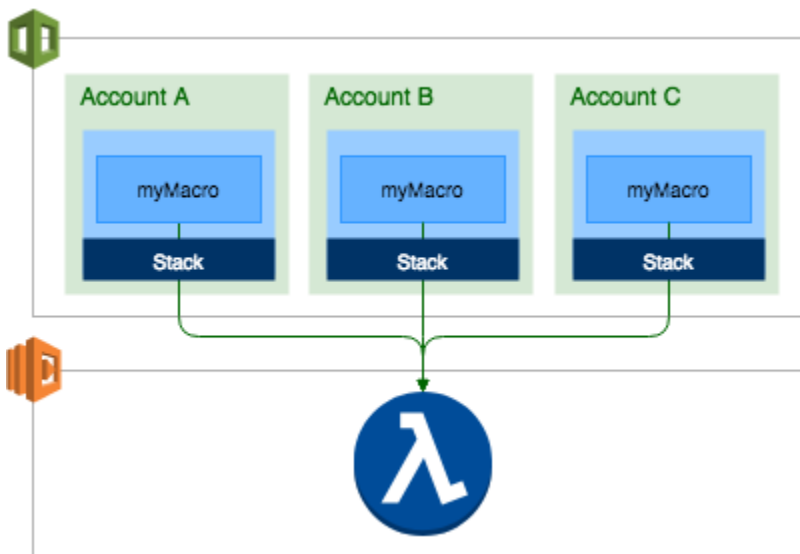
Ao trabalhar com macros, tenha em mente as seguintes observações e limitações:

- As macros são compatíveis apenas com as Regiões da AWS em que o Lambda está disponível. Para obter uma lista de regiões em que o Lambda está disponível, consulte [Endpoints e cotas do AWS Lambda](#).
- Todos os trechos do modelo processado deve ser um JSON válido.
- Quaisquer trechos de modelo processados devem passar por verificações de validação para uma operação de criação de pilha, atualização de pilha, criação de conjunto de pilhas ou atualização de conjunto de pilhas.
- O CloudFormation resolve as macros primeiro e, depois, processa o modelo. O modelo resultante deve ser um JSON válido e não deve exceder o limite de tamanho do modelo.
- Devido à ordem na qual o CloudFormation processa elementos em um modelo, uma macro não pode incluir módulos no conteúdo do modelo processado que ela retorna ao CloudFormation. Para obter mais informações, consulte [Ordem de avaliação de macros](#).
- Ao usar o recurso de reversão de atualização, o CloudFormation utiliza uma cópia do modelo original. Ele reverterá para o modelo original, mesmo se o trecho incluído tiver sido alterado.
- A inclusão de macros dentro de macros não funciona porque não processamos macros de modo recursivo.
- No momento, a função intrínseca `Fn::ImportValue` não é compatível em macros.

- As funções intrínsecas incluídas no modelo são avaliadas depois de quaisquer macros. Portanto, o conteúdo do modelo processado retornado pela macro pode incluir chamadas para funções intrínsecas, e estas são avaliadas como de costume.
- No momento, o StackSets não permite a criação nem a atualização de conjuntos de pilhas com permissões gerenciadas por serviço de modelos que fazem referência a macros do CloudFormation.

Escopo e permissões de conta das macros

Você pode usar macros apenas na conta em que elas foram criadas como um recurso. O nome da macro deve ser exclusivo dentro de uma conta especificada. No entanto, você pode disponibilizar a mesma funcionalidade em várias contas, ativando o acesso entre contas na função Lambda subjacente e, em seguida, criando definições de macro que fazem referência a essa função em várias contas. No exemplo abaixo, três contas contêm definições de macro, cada uma apontando para a mesma função Lambda.



Para criar uma definição de macro, o usuário deve ter permissões para criar uma pilha ou um conjunto de pilhas dentro da conta especificada.

Para o CloudFormation executar com êxito uma macro incluída em um modelo, o usuário deve ter permissões Invoke para a função do Lambda subjacente. Para evitar um possível dimensionamento de permissões, o CloudFormation personifica o usuário ao executar a macro.

Para mais informações, consulte [Gerenciando permissões no AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda e [Actions, resources, and condition keys for AWS Lambda](#) na Service Authorization Reference.

Criar uma definição de macro do CloudFormation

Quando você cria uma definição de macro, ela disponibiliza a função do Lambda subjacente na conta especificada para que o CloudFormation possa invocá-la ao processar os modelos.

Mapeamento de eventos

Quando o CloudFormation invoca a função do Lambda de uma macro, ele envia uma solicitação no formato JSON com a seguinte estrutura:

```
{
  "region" : "us-east-1",
  "accountId" : "$ACCOUNT_ID",
  "fragment" : { ... },
  "transformId" : "$TRANSFORM_ID",
  "params" : { ... },
  "requestId" : "$REQUEST_ID",
  "templateParameterValues" : { ... }
}
```

- **region**

A região na qual a macro reside.

- **accountId**

O ID de conta da conta da qual a macro está invocando a função Lambda.

- **fragment**

O conteúdo do modelo disponível para processamento personalizado, no formato JSON.

- Para macros incluídas no modelo Transform, essa seção é o modelo inteiro, exceto pela seção Transform.

- Para macros incluídas em uma chamada de função intrínseca Fn::Transform, isso inclui todos os nós irmãos (e os filhos deles) com base na localização da função intrínseca no modelo, exceto pela função Fn::Transform. Para obter mais informações, consulte [Escopo de modelo de macro](#).

- **transformId**

O nome da macro que invoca essa função.

- **params**

Para chamadas à função `Fn::Transform`, quaisquer parâmetros especificados para a função. O CloudFormation não avalia esses parâmetros antes de passá-los para a função.

Para macros incluídas na seção do modelo `Transform`, essa seção é vazia.

- `requestId`

O ID da solicitação que invoca essa função.

- `templateParameterValues`

Quaisquer parâmetros especificados na seção [Parameters](#) do modelo. O CloudFormation avalia esses parâmetros antes de passá-los para a função.

Formato de resposta

O CloudFormation espera que a função do Lambda subjacente retorne uma resposta no seguinte formato JSON:

```
{
  "requestId" : "$REQUEST_ID",
  "status" : "$STATUS",
  "fragment" : { ... },
  "errorMessage": "optional error message for failures"
}
```

- `requestId`

O ID da solicitação que invoca essa função. Isso deve corresponder ao ID de solicitação fornecido pelo CloudFormation ao invocar a função.

- `status`

O status da solicitação (não diferencia maiúsculas de minúsculas). Ele deve ser definido como `success`. O CloudFormation trata qualquer outra resposta como uma falha.

- `fragment`

O conteúdo do modelo processado para o CloudFormation incluir no modelo processado, incluindo irmãos. O CloudFormation substitui o conteúdo do modelo que é transmitido para a função do Lambda pelo fragmento de modelo que ele recebe na resposta do Lambda.

O conteúdo do modelo processado deve ser um JSON válido, e sua inclusão no modelo processado deve resultar em um modelo válido.

Se a função não chegar a alterar o conteúdo do modelo que o CloudFormation transmite a ela, mas você ainda precisar incluir esse conteúdo no modelo processado, a função precisará retornar conteúdo do modelo ao CloudFormation em sua resposta.

- `errorMessage`

A mensagem de erro que explica por que a transformação falhou. O CloudFormation exibe a mensagem de erro no painel Events (Eventos) da página Stack details (Detalhes da pilha) para sua nova pilha.

Por exemplo:

```
Error creating change set: Transform
      Conta da AWS account
      number::macro name failed with:
      error message string.
```

Criar uma definição de macro

Para criar uma definição de macro do CloudFormation

1. [Desenvolva uma função do Lambda](#) que processe o conteúdo de um modelo. Ela pode processar qualquer parte de um modelo, até o modelo inteiro.
2. Crie um modelo do CloudFormation contendo um tipo de recurso `AWS::CloudFormation::Macro` e especifique as propriedades `Name` e `FunctionName`. A propriedade `FunctionName` deve conter o ARN da função do Lambda a ser invocada quando o CloudFormation executar a macro.
3. (opcional) Para ajudar na depuração, você também pode especificar as propriedades `LogGroupName` e `LogRoleArn` ao criar o tipo de recurso `AWS::CloudFormation::Macro` para a macro. Essas propriedades permitem especificar o grupo de logs do CloudWatch para o qual o CloudFormation envia informações de registro de erros em log ao invocar a função do Lambda subjacente da macro e o perfil que o CloudFormation deve assumir ao enviar entradas para esses logs.

4. [Crie uma pilha](#) usando o modelo com a macro na conta em que você deseja usá-la. Ou [crie um conjunto de pilhas com permissões autogerenciadas](#) usando o modelo com a macro na conta de administrador, depois crie instâncias de pilhas nas contas de destino.
5. Depois que o CloudFormation criar com êxito as pilhas que contêm a definição da macro, ela estará disponível para uso nessas contas. Você usa uma macro fazendo referência a ela em um modelo, no local apropriado relevante para o conteúdo do modelo que você deseja processar.

Escopo de modelo de macro

Macros referenciadas na seção Transform de um modelo podem processar todo o conteúdo desse modelo.

Macros referenciadas em uma função `Fn::Transform` podem processar o conteúdo de qualquer um dos elementos irmãos (incluindo filhos) dessa função `Fn::Transform` no modelo.

Por exemplo, no modelo de exemplo abaixo, `AWS::Include` pode processar todas as propriedades `MyBucket` com base na localização da função `Fn::Transform` que o contém. `MyMacro` pode processar o conteúdo de todo o modelo devido à sua inclusão na seção Transform.

```
# Start of processable content for MyMacro
AWSTemplateFormatVersion: 2010-09-09
Transform: [MyMacro]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    # Start of processable content for AWS::Include
    Properties:
      BucketName: amzn-s3-demo-bucket1
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: 'AWS::Include'
          Parameters:
            Location: s3://amzn-s3-demo-bucket2/MyFileName.yaml
      CorsConfiguration: []
    # End of processable content for AWS::Include
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageID: ami-1234567890abcdef0
```

```
# End of processable content for MyMacro
```

Ordem de avaliação de macros

É possível fazer referência a várias macros em um determinado modelo, incluindo transformações hospedadas pelo CloudFormation, como [AWS::Include](#) e [AWS::Serverless](#).

As macros são avaliadas em ordem, com base na sua localização no modelo, da mais profundamente aninhada até a mais genérica. Macros no mesmo local do modelo são avaliadas em série com base na ordem em que estão listadas.

Transformações como `AWS::Include` e `AWS::Transform` são tratadas como qualquer outra macro em termos de ordem e escopo de ação.

Por exemplo, no modelo de exemplo abaixo, o CloudFormation avalia primeiro a macro `PolicyAdder`, pois ela é a mais profundamente aninhada no modelo. Depois, o CloudFormation avalia `MyMacro` antes de avaliar `AWS::Serverless` porque ela está listada antes de `AWS::Serverless` na seção `Transform`.

```
AWSTemplateFormatVersion: 2010-09-09
Transform: [MyMacro, AWS::Serverless]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: PolicyAdder
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageID: ami-1234567890abcdef0
```

Exemplo de macro simples de substituição de string

O exemplo a seguir mostra todo o processo de uso das macros, desde a definição de uma macro em um modelo até a criação de uma função do Lambda para a macro e o uso da macro em um modelo.

Neste exemplo, criamos uma macro simples que insere a sequência especificada no lugar do conteúdo de destino especificado no modelo processado. E, em seguida, vamos usá-la para inserir um espaço em branco `WaitHandleCondition` no local especificado do modelo processado.

Criar uma macro

Antes de usar uma macro, é necessário concluir duas tarefas: criar a função do Lambda que executa o processamento de modelo desejado e, em seguida, disponibilizar essa função do Lambda ao CloudFormation, criando uma definição de macro.

O modelo de exemplo a seguir contém a definição da nossa macro de exemplo. Para tornar a macro disponível em uma Conta da AWS específica, crie uma pilha a partir do modelo. A definição da macro especifica o nome da macro, uma breve descrição e faz referência ao ARN da função Lambda que o CloudFormation invoca quando essa macro é usada em um modelo. (Não incluímos uma propriedade `LogGroupName` ou `LogRoleARN` para o registro em log de erros.)

Neste exemplo, pressuponha que a pilha criada a partir desse modelo seja denominada `JavaMacroFunc`. Como a propriedade `Name` da macro é definida como o nome da pilha, a macro resultante também será denominada `JavaMacroFunc`.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Macro:
    Type: AWS::CloudFormation::Macro
    Properties:
      Name: !Sub '${AWS::StackName}'
      Description: Adds a blank WaitConditionHandle named WaitHandle
      FunctionName: 'arn:aws:lambda:us-east-1:012345678910:function:JavaMacroFunc'
```

Usar a macro

Para usar nossa macro, a incluímos em um modelo usando a função intrínseca `Fn::Transform`.

Quando você cria uma pilha usando o modelo a seguir, o CloudFormation chama nossa macro de exemplo. A função Lambda subjacente substitui uma sequência especificada por outra sequência especificada. Nesse caso, o resultado é que um `AWS::CloudFormation::WaitConditionHandle` em branco é inserido no modelo processado.

```
Parameters:
  ExampleParameter:
```

```
Type: String
Default: 'SampleMacro'
```

Resources:

```
2a:
  Fn::Transform:
    Name: "JavaMacroFunc"
    Parameters:
      replacement: 'AWS::CloudFormation::WaitConditionHandle'
      target: '$$REPLACEMENT$$'
    Type: '$$REPLACEMENT$$'
```

- A macro a ser invocada é especificada como `JavaMacroFunc`, do exemplo anterior de definição de macro.
- A macro recebe dois parâmetros, `target` e `replacement`, que representam a sequência de destino e seu valor de substituição desejado.
- A macro pode operar no conteúdo do nó `Type`, pois `Type` é um irmão da função `Fn::Transform` que referencia a macro.
- O `AWS::CloudFormation::WaitConditionHandle` resultante se chama `2a`.
- O modelo também contém um parâmetro de modelo, `ExampleParameter`, ao qual a macro também tem acesso (mas não usa neste caso).

Dados de entrada do Lambda

Quando o CloudFormation processa nosso exemplo de modelo durante a criação da pilha, ele transmite o mapeamento de eventos a seguir à função do Lambda referenciada na definição da macro `JavaMacroFunc`.

- `region : us-east-1`
- `accountId : 012345678910`
- `fragment :`

```
{
  "Type": "$$REPLACEMENT$$"
}
```

- `transformId : 012345678910::JavaMacroFunc`
- `params :`

```
{
  "replacement": "AWS::CloudFormation::WaitConditionHandle",
  "target": "$$REPLACEMENT$$"
}
```

- requestId : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- templateParameterValues :

```
{
  "ExampleParameter": "SampleMacro"
}
```

fragment contém um JSON que representa o fragmento de modelo que a macro pode processar. Esse fragmento consiste nos irmãos da chamada da função Fn::Transform, mas não a chamada de função propriamente dita. Além disso, params contém um JSON que representa os parâmetros da macro. Nesse caso, a substituição e o destino. Da mesma forma, templateParameterValues contém um JSON que representa os parâmetros especificados para o modelo como um todo.

Código da função do Lambda

O código a seguir é o código real para a função do Lambda subjacente ao exemplo da macro JavaMacroFunc. Ele itera sobre o fragmento de modelo incluído na resposta (seja em formato de sequência, lista ou mapa), procurando a sequência de destino especificada. Se ele encontrar a sequência de destino especificada, a função Lambda substituirá a sequência de destino pela sequência de substituição especificada. Caso contrário, a função deixará o fragmento de modelo inalterado. Em seguida, a função retorna um mapa das propriedades esperadas, discutido em detalhes abaixo, para o CloudFormation.

```
package com.macroexample.lambda.demo;

import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
```

```
public class LambdaFunctionHandler implements RequestHandler<Map<String, Object>,
    Map<String, Object>> {

    private static final String REPLACEMENT = "replacement";
    private static final String TARGET = "target";
    private static final String PARAMS = "params";
    private static final String FRAGMENT = "fragment";
    private static final String REQUESTID = "requestId";
    private static final String STATUS = "status";
    private static final String SUCCESS = "SUCCESS";
    private static final String FAILURE = "FAILURE";

    @Override
    public Map<String, Object> handleRequest(Map<String, Object> event, Context
context) {
        // TODO: implement your handler
        final Map<String, Object> responseMap = new HashMap<String, Object>();
        responseMap.put(REQUESTID, event.get(REQUESTID));
        responseMap.put(STATUS, FAILURE);
        try {
            if (!event.containsKey(PARAMS)) {
                throw new RuntimeException("Params are required");
            }

            final Map<String, Object> params = (Map<String, Object>) event.get(PARAMS);
            if (!params.containsKey(REPLACEMENT) || !params.containsKey(TARGET)) {
                throw new RuntimeException("replacement or target under Params are
required");
            }

            final String replacement = (String) params.get(REPLACEMENT);
            final String target = (String) params.get(TARGET);
            final Object fragment = event.getOrDefault(FRAGMENT, new HashMap<String,
Object>());
            final Object retFragment;
            if (fragment instanceof String) {
                retFragment = iterateAndReplace(replacement, target, (String) fragment);
            } else if (fragment instanceof List) {
                retFragment = iterateAndReplace(replacement, target, (List<Object>) fragment);
            } else if (fragment instanceof Map) {
                retFragment = iterateAndReplace(replacement, target, (Map<String, Object>)
fragment);
            } else {
                retFragment = fragment;
            }
        }
    }
}
```

```

        responseMap.put(STATUS, SUCCESS);
        responseMap.put(FRAGMENT, retFragment);
        return responseMap;
    } catch (Exception e) {
        e.printStackTrace();
        context.getLogger().log(e.getMessage());
        return responseMap;
    }
}

```

```

private Map<String, Object> iterateAndReplace(final String replacement, final
String target, final Map<String, Object> fragment) {
    final Map<String, Object> retFragment = new HashMap<String, Object>();
    final List<String> replacementKeys = new ArrayList<>();
    fragment.forEach((k, v) -> {
        if (v instanceof String) {
            retFragment.put(k, iterateAndReplace(replacement, target, (String)v));
        } else if (v instanceof List) {
            retFragment.put(k, iterateAndReplace(replacement, target, (List<Object>)v));
        } else if (v instanceof Map ) {
            retFragment.put(k, iterateAndReplace(replacement, target, (Map<String, Object>
v)));
        } else {
            retFragment.put(k, v);
        }
    });
    return retFragment;
}

```

```

private List<Object> iterateAndReplace(final String replacement, final String
target, final List<Object> fragment) {
    final List<Object> retFragment = new ArrayList<>();
    fragment.forEach(o -> {
        if (o instanceof String) {
            retFragment.add(iterateAndReplace(replacement, target, (String) o));
        } else if (o instanceof List) {
            retFragment.add(iterateAndReplace(replacement, target, (List<Object>) o));
        } else if (o instanceof Map) {
            retFragment.add(iterateAndReplace(replacement, target, (Map<String, Object>
o));
        } else {
            retFragment.add(o);
        }
    });
}

```

```
        return retFragment;
    }

    private String iterateAndReplace(final String replacement, final String target,
final String fragment) {
        System.out.println(replacement + " == " + target + " == " + fragment );
        if (fragment != null AND_AND fragment.equals(target))
            return replacement;
        return fragment;
    }
}
```

Resposta da função do Lambda

Veja a seguir o mapeamento que a função do Lambda retorna ao CloudFormation para processamento.

- `requestId` : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- `status` : SUCCESS
- `fragment` :

```
{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}
```

As correspondências de `requestId` enviadas do CloudFormation e um valor `status` de SUCCESS indica que a função Lambda processou com êxito o fragmento de modelo incluído na solicitação. Nessa resposta, `fragment` contém um JSON que representa o conteúdo a ser inserido no modelo processado no lugar do trecho de modelo original.

Modelo processado resultante

Depois que o CloudFormation recebe uma resposta bem-sucedida da função do Lambda, ele insere o fragmento de modelo retornado no modelo processado.

Veja a seguir o modelo processado resultante para o nosso exemplo. A chamada da função intrínseca `Fn::Transform` que referenciava a macro `JavaMacroFunc` não está mais incluída. O fragmento de modelo retornado pela função do Lambda está incluído no local apropriado, com o resultado de que o conteúdo `"Type": "$$REPLACEMENT$$"` foi substituído por `"Type": "AWS::CloudFormation::WaitConditionHandle"`.


```
{
  "Parameters": {
    "ExampleParameter": {
      "Default": "SampleMacro",
      "Type": "String"
    }
  },
  "Resources": {
    "2a": {
      "Type": "AWS::CloudFormation::WaitConditionHandle"
    }
  }
}
```

Solucionar problemas do modelo processado

Ao usar uma macro, o modelo processado pode ser encontrado no console do CloudFormation.

O estágio de um modelo indica seu status de processamento:

- **Original:** o modelo que o usuário enviou originalmente para criar ou atualizar a pilha ou o conjunto de pilhas.
- **Processed:** o modelo que o CloudFormation usou para criar ou atualizar a pilha ou o conjunto de pilhas após o processamento de todas as macros referenciadas. O modelo processado é formatado como JSON, mesmo que o modelo original tenha sido formatado como YAML.

Para solucionar problemas, use o modelo processado. Se um modelo não referenciar macros, os modelos original e processado serão idênticos.

Para obter mais informações, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Para usar a AWS CLI para obter o modelo processado, use o comando [get-template](#).

Limitação de tamanho

O tamanho máximo de um modelo de pilha processado é de 51.200 bytes quando transmitido diretamente em uma solicitação `CreateStack`, `UpdateStack` ou `ValidateTemplate`, ou 1 MB quando transmitido como um objeto do S3 usando um URL do modelo do Amazon S3. No entanto, durante o processamento, o CloudFormation atualiza o estado temporário do modelo enquanto ele

processa serialmente as macros contidas no modelo. Por causa disso, o tamanho do modelo durante o processamento pode temporariamente exceder o tamanho permitido de um modelo totalmente processado. O CloudFormation permite um buffer para esses modelos em processamento. No entanto, você deve projetar modelos e macros tendo em mente o tamanho máximo permitido para um modelo de pilha processado.

Se o CloudFormation retornar um erro `Transformation data limit exceeded` ao processar seu modelo, é porque ele excedeu o tamanho máximo de modelo que o CloudFormation permite durante o processamento.

Para resolver esse problema, considere o seguinte:

- Reestruture o modelo em vários modelos para evitar exceder o tamanho máximo de modelos em andamento. Por exemplo:
 - Use os modelos de pilha aninhada para encapsular partes do modelo. Para obter mais informações, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).
 - Crie várias pilhas e use referências entre pilhas para trocar informações entre elas. Para obter mais informações, consulte [Consultar saídas de recurso em outra pilha do CloudFormation](#).
- Reduza o tamanho do fragmento do modelo retornado por determinada macro. O CloudFormation não interfere no conteúdo de fragmentos retornados por macros.

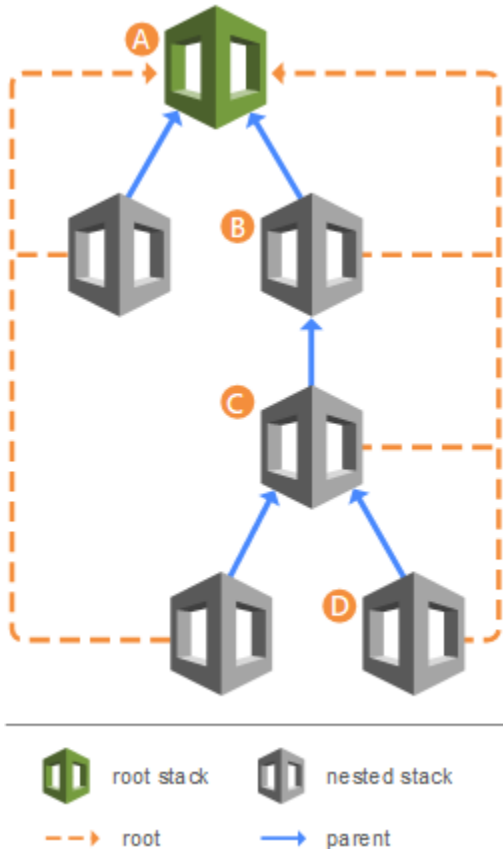
Dividir um modelo em partes reutilizáveis usando pilhas aninhadas

À medida que sua infraestrutura cresce, você poderá se encontrar criando repetidamente configurações de recursos idênticas em vários modelos. Para evitar essa redundância, é possível separar essas configurações comuns em modelos dedicados. Depois, use o recurso [AWS::CloudFormation::Stack](#) em seu modelo para fazer referência a esses modelos dedicados, criando pilhas aninhadas.

Por exemplo, suponha que você tem uma configuração de balanceador de carga para a maioria de suas pilhas. Em vez de copiar e colar as mesmas configurações em seus modelos, é possível criar um modelo dedicado para o load balancer. Em seguida, você poderá fazer referência a esse modelo a partir de outros modelos que exigem a mesma configuração do balanceador de carga.

As pilhas aninhadas podem conter outras pilhas aninhadas, o que resulta em uma hierarquia de pilhas, conforme mostrado no diagrama a seguir. A pilha raiz é a pilha de nível superior à qual todas as pilhas aninhadas pertencem. Cada pilha aninhada tem uma pilha pai imediata. Para o primeiro nível de pilhas aninhadas, a pilha raiz também é a pilha pai.

- Pilha A é a pilha raiz de todas as outras pilhas, aninhadas, na hierarquia.
- Para a pilha B, a pilha A é a pilha pai e a pilha raiz.
- Para a pilha D, a pilha C é a pilha pai; e, ao mesmo tempo, para a pilha C, a pilha B é a pilha pai.



Tópicos

- [Exemplo de antes e depois da divisão de um modelo](#)
- [Exemplo de uma arquitetura de pilha aninhada](#)
- [Executar operações de pilha em pilhas aninhadas](#)
- [Informações relacionadas](#)

Exemplo de antes e depois da divisão de um modelo

Este exemplo demonstra como é possível pegar um modelo único e grande do CloudFormation e reorganizá-lo em um design mais estruturado e reutilizável usando modelos aninhados. Inicialmente, o modelo “Antes de aninhar pilhas” mostra todos os recursos definidos em um arquivo. Isso pode se tornar confuso e difícil de gerenciar à medida que o número de recursos aumenta. O modelo

"Depois de aninhar pilhas" divide os recursos em modelos menores e separados, chamados de pilhas aninhadas. Cada pilha aninhada manipula um conjunto específico de recursos relacionados, tornando a estrutura geral mais organizada e fácil de manter.

Antes de aninhar pilhas	Depois de aninhar pilhas
<pre> AWSTemplateFormatVersion: 2010-09-09 Parameters: InstanceType: Type: String Default: t2.micro Description: The EC2 instance type Environment: Type: String Default: Production Description: The deployment environment Resources: MyEC2Instance: Type: AWS::EC2::Instance Properties: ImageId: ami-1234567890abcdef0 InstanceType: !Ref InstanceType MyS3Bucket: Type: AWS::S3::Bucket </pre>	<pre> AWSTemplateFormatVersion: 2010-09-09 Resources: MyFirstNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ <i>amzn-s3-demo-bucket</i> /first-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed InstanceType: t3.micro MySecondNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ <i>amzn-s3-demo-bucket</i> /second-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed Environment: Testing DependsOn: MyFirstNestedStack </pre>

Exemplo de uma arquitetura de pilha aninhada

Esta seção demonstra uma arquitetura de pilha aninhada que consiste em uma pilha de nível superior que faz referência a uma pilha aninhada. A pilha aninhada implanta uma função do Lambda em Node.js, recebe um valor de parâmetro da pilha de nível superior e retorna uma saída que é exposta por meio da pilha de nível superior.

Tópicos

- [Etapa 1: criar um modelo para a pilha aninhada em seu sistema local](#)
- [Etapa 2: criar um modelo para a pilha de nível superior em seu sistema local](#)
- [Etapa 3: empacotar e implantar os modelos](#)

Etapa 1: criar um modelo para a pilha aninhada em seu sistema local

O exemplo a seguir mostra o formato do modelo da pilha aninhada.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Nested stack template for Lambda function deployment
Parameters:
  MemorySize:
    Type: Number
    Default: 128
    MinValue: 128
    MaxValue: 10240
    Description: Lambda function memory allocation (128-10240 MB)
Resources:
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Sub "${AWS::StackName}-Function"
      Runtime: nodejs18.x
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          exports.handler = async (event) => {
            return {
              statusCode: 200,
              body: JSON.stringify('Hello from Lambda!')
            };
          };
      MemorySize: !Ref MemorySize
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
```

```
- Effect: Allow
  Principal:
    Service: lambda.amazonaws.com
  Action: sts:AssumeRole
ManagedPolicyArns:
  - 'arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
```

Outputs:

LambdaArn:

Description: ARN of the created Lambda function

Value: !GetAtt LambdaFunction.Arn

Etapa 2: criar um modelo para a pilha de nível superior em seu sistema local

O exemplo a seguir mostra o formato do modelo de pilha de nível superior e o recurso [AWS::CloudFormation::Stack](#) que faz referência à pilha que você criou na etapa anterior.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Top-level stack template that deploys a nested stack
Resources:
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: /path_to_template/nested-template.yaml
      Parameters:
        MemorySize: 256
Outputs:
  NestedStackLambdaArn:
    Description: ARN of the Lambda function from nested stack
    Value: !GetAtt NestedStack.Outputs.LambdaArn
```

Etapa 3: empacotar e implantar os modelos

Note

Ao trabalhar com modelos localmente, o comando `package` da AWS CLI pode ajudar você a preparar modelos para implantação. Ele gerencia automaticamente o upload de artefatos locais para o Amazon S3 (incluindo `TemplateURL`) e gera um novo arquivo de modelo com referências atualizadas a esses locais do S3. Para obter mais informações, consulte [Fazer upload de artefatos locais em um bucket do S3 com a AWS CLI](#).

Em seguida, você pode usar o comando [package](#) para carregar o modelo aninhado em um bucket do Amazon S3.

```
aws cloudformation package \  
  --s3-bucket amzn-s3-demo-bucket \  
  --template /path_to_template/top-level-template.yaml \  
  --output-template-file packaged-template.yaml \  
  --output json
```

O comando gera um novo modelo no caminho especificado por `--output-template-file`. Ele substitui a referência `TemplateURL` pelo local do Amazon S3, conforme mostrado abaixo.

Modelo resultante

```
AWSTemplateFormatVersion: 2010-09-09  
Description: Top-level stack template that deploys a nested stack  
Resources:  
  NestedStack:  
    Type: AWS::CloudFormation::Stack  
    Properties:  
      TemplateURL: https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket/8b3bb7aa7abfc6e37e2d06b869484bed.template  
      Parameters:  
        MemorySize: 256  
Outputs:  
  NestedStackLambdaArn:  
    Description: ARN of the Lambda function from nested stack  
    Value:  
      Fn::GetAtt:  
        - NestedStack  
        - Outputs.LambdaArn
```

Depois de executar o comando `package`, você pode implantar o modelo processado usando o comando [deploy](#). Para pilhas aninhadas que contêm recursos do IAM, é necessário reconhecer os recursos do IAM com a inclusão da opção `--capabilities`.

```
aws cloudformation deploy \  
  --template-file packaged-template.yaml \  
  --stack-name stack-name \  
  --capabilities CAPABILITY_NAMED_IAM
```

Executar operações de pilha em pilhas aninhadas

Ao trabalhar com pilhas aninhadas, você deverá manuseá-las cuidadosamente durante as operações. Algumas operações de pilha, como atualizações de pilha, devem ser iniciadas na pilha raiz em vez de serem realizadas diretamente em pilhas aninhadas. Quando você atualiza uma pilha raiz, somente as pilhas aninhadas com alterações de modelo serão atualizadas.

Além disso, a presença das pilhas aninhadas pode afetar as operações na pilha raiz. Por exemplo, se uma pilha aninhada ficar aprisionada no estado `UPDATE_ROLLBACK_IN_PROGRESS`, a pilha raiz aguardará até que a pilha aninhada conclua a reversão antes de continuar. Antes de prosseguir com as operações de atualização, certifique-se de ter as permissões do IAM necessárias para cancelar uma atualização da pilha caso ela seja revertida. Para obter mais informações, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Use os procedimentos a seguir para encontrar a pilha raiz e as pilhas aninhadas.

Para visualizar a pilha raiz de uma pilha aninhada

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, escolha o nome da pilha aninhada da qual deseja exibir a pilha raiz.

As pilhas aninhadas exibem ANINHADA ao lado do nome da pilha.

3. Na guia Informações da pilha, na seção Visão geral, selecione o nome da pilha listada como Pilha raiz.

Para visualizar as pilhas aninhadas que pertencem a uma pilha raiz

1. Escolha o nome da pilha raiz cujas pilhas aninhadas você deseja visualizar e, em seguida, escolha a guia Recursos.
2. Na coluna Tipo, procure recursos do tipo `AWS::CloudFormation::Stack`.

Informações relacionadas

- [Aninhar uma pilha existente](#)
- [Compreender atualização de comportamentos de recursos da pilha](#)
- [Continue revertendo de atualizações de pilhas aninhadas que falharam](#)

- [Falha na reversão de pilhas aninhadas](#)

Criar condições de espera em um modelo do CloudFormation

Este tópico explica como criar uma condição de espera em um modelo para coordenar a criação de recursos de pilha ou acompanhar o progresso de um processo de configuração. Por exemplo, você pode iniciar a criação de outro recurso depois que a configuração de um aplicativo é parcialmente concluída, ou você pode enviar sinais durante um processo de instalação e configuração para acompanhar seu progresso.

Quando o CloudFormation cria uma pilha que inclui uma condição de espera:

- Ele cria uma condição de espera como qualquer outro recurso e define o status da condição de espera como `CREATE_IN_PROGRESS`.
- O CloudFormation espera até receber o número necessário de sinais de sucesso ou até que o período de tempo limite da condição de espera seja atingido.
- Se ele receber o número necessário de sinais de sucesso antes que o período de tempo limite seja atingido:
 - Espere que o status da condição mude para `CREATE_COMPLETE`
 - A criação da pilha continua
- Se o tempo limite for atingido ou um sinal de falha for recebido:
 - Espere que o status da condição mude para `CREATE_FAILED`
 - Reverter a pilha

Important

Para recursos do Amazon EC2 e do Auto Scaling, recomendamos usar um atributo `CreationPolicy`, em vez de condições de espera. Adicione um atributo `CreationPolicy` a esses recursos e use o script auxiliar `cfn-signal` para sinalizar quando um processo de criação de instância foi concluído com êxito.

Para obter mais informações, consulte o atributo [CreationPolicy](#).

 Note

Se você usar o AWS PrivateLink, os recursos do VPC que respondem a condições de espera deverão ter acesso aos buckets do Amazon Simple Storage Service (Amazon S3) específicos do CloudFormation. Recursos devem enviar respostas à condição de espera para um URL pré-assinado do Amazon S3. Se não puderem enviar respostas para o Amazon S3, o CloudFormation não receberá uma resposta e há falha na operação de pilha. Para obter mais informações, consulte [Acessar o CloudFormation usando um endpoint de interface \(AWS PrivateLink\)](#) e [Controlar o acesso a partir de VPC endpoints com políticas de bucket](#).

Tópicos

- [Criar uma condição de espera no modelo](#)
- [Sintaxe de sinal da condição de espera](#)
- [Acessar dados de sinal](#)

Criar uma condição de espera no modelo

1. Processamento de condição de espera

Você começa definindo um recurso [AWS::CloudFormation::WaitConditionHandle](#) no modelo da pilha. Esse recurso gera o URL pré-assinado necessário para enviar sinais. Isso permite que você envie um sinal sem precisar fornecer as suas credenciais da AWS. Por exemplo:

Resources:

MyWaitHandle:

Type: AWS::CloudFormation::WaitConditionHandle

2. Condição de espera

Depois, você define um recurso [AWS::CloudFormation::WaitCondition](#) no modelo da pilha. A estrutura básica de um recurso AWS::CloudFormation::WaitCondition é assim:

MyWaitCondition:

Type: AWS::CloudFormation::WaitCondition

Properties:

```
Handle: String
Timeout: String
Count: Integer
```

O recurso `AWS::CloudFormation::WaitCondition` tem duas propriedades obrigatórias e uma propriedade opcional.

- **Handle** (obrigatório): uma referência a um `WaitConditionHandle` declarado no modelo.
- **Timeout** (obrigatório): o número de segundos que o CloudFormation deve esperar até o número necessário de sinais ser recebido. `Timeout` é uma propriedade de limite mínimo, o que significa que o tempo limite não ocorre antes do tempo especificado, mas pode ocorrer logo depois. O tempo máximo que você pode especificar é 43200 segundos (12 horas).
- **Count** (opcional): o número de sinais de sucesso que o CloudFormation deve receber antes de definir o status dessa condição de espera como `CREATE_COMPLETE` e continuar criando a pilha. Se não for especificado, o valor padrão é 1.

Normalmente, você quer que uma condição de espera comece imediatamente após a criação de um determinado recurso. Para isso, adicione o atributo `DependsOn` a uma condição de espera. Quando você adiciona um atributo `DependsOn` a uma condição de espera, o CloudFormation primeiro cria o recurso no atributo `DependsOn` e depois cria a condição de espera. Para obter mais informações, consulte o atributo [DependsOn](#).

O seguinte exemplo ilustra uma condição de espera que:

- Começa após a criação bem-sucedida do recurso `MyEC2Instance`
- Usa o recurso `MyWaitHandle` como `WaitConditionHandle`
- Tem um tempo limite de 4.500 segundos
- Tem a `Count` padrão de 1 (já que nenhuma propriedade `Count` está especificada)

```
MyWaitCondition:
  Type: AWS::CloudFormation::WaitCondition
  DependsOn: MyEC2Instance
  Properties:
    Handle: !Ref MyWaitHandle
    Timeout: '4500'
```

3. Enviar um sinal

Para sinalizar sucesso ou falha para o CloudFormation, você normalmente executa algum código ou script. Por exemplo, uma aplicação em execução em uma instância do EC2 pode realizar algumas tarefas de configuração adicionais e enviar um sinal para o CloudFormation para indicar a conclusão.

O sinal deve ser enviado para o URL pré-assinado gerado pelo identificador da condição de espera. Você usa o URL pré-assinado para sinalizar sucesso ou falha.

Para enviar um sinal

1. Para recuperar o URL pré-assinado no modelo, use a função intrínseca Ref com o nome lógico do identificador da condição de espera.

Conforme mostrado no exemplo a seguir, o modelo pode declarar uma instância do Amazon EC2 e passar o URL pré-assinado para instâncias EC2 usando a propriedade UserData do Amazon EC2. Isso permite que scripts ou aplicações executados nessas instâncias sinalizem sucesso ou falha para o CloudFormation.

```
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: t2.micro # Example instance type
    ImageId: ami-055e3d4f0bbeb5878 # Change this as needed (Amazon Linux 2023 in
us-west-2)
    UserData:
      Fn::Base64:
        Fn::Join:
          - ""
          - - "SignalURL="
            - { "Ref": "MyWaitHandle" }
```

Isso gera uma saída UserData assim:

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/...
```

Observação: no Console de gerenciamento da AWS e nas ferramentas de linha de comando, o URL pré-assinado é exibido como um ID físico do identificador de condição de espera.

2. (Opcional) Para detectar quando a pilha entra na condição de espera, você pode usar um dos seguintes métodos:

- Se você criar a pilha com notificações habilitadas, o CloudFormation publicará uma notificação para cada evento da pilha no tópico especificado. Se você ou seu aplicativo se inscrever nesse tópico, você poderá monitorar as notificações do evento de criação do identificador de condição de espera e recuperará o pre-signed URL na mensagem de notificação.
 - Você também pode monitorar os eventos da pilha usando o Console de gerenciamento da AWS, a AWS CLI ou um SDK.
3. Para enviar um sinal, envie uma mensagem de solicitação HTTP usando o pre-signed URL. O método de solicitação deve ser PUT, e o cabeçalho Content-Type deve ser omitido ou uma string vazia. A mensagem de solicitação deve ser uma estrutura JSON do formulário especificado em [Sintaxe de sinal da condição de espera](#).

É necessário enviar o número de sinais de sucesso especificados pela propriedade Count para que o CloudFormation continue a criar a pilha. Se a propriedade Count for maior do que 1, o valor do UniqueId de cada sinal deverá ser exclusivo em todos os sinais enviados a uma condição de espera específica. O UniqueId é uma string alfanumérica arbitrária.

Um comando `curl` é uma forma de enviar um sinal. O exemplo a seguir mostra uma linha do comando `curl` que sinaliza o êxito para uma condição de espera.

```
$ curl -T /tmp/a \  
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-  
west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-  
WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?  
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo  
%3D"
```

em que o arquivo `/tmp/a` contém a seguinte estrutura JSON:

```
{  
  "Status" : "SUCCESS",  
  "Reason" : "Configuration Complete",  
  "UniqueId" : "ID1234",  
  "Data" : "Application has completed configuration."  
}
```

Este exemplo mostra uma linha do comando `curl` que envia o mesmo sinal de êxito, exceto que ele envia a estrutura JSON como um parâmetro na linha de comando.

```
$ curl -X PUT \
  -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS", "Reason" : "Configuration Complete", "UniqueId" : "ID1234", "Data" : "Application has completed configuration."}' \
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo%3D"
```

Sintaxe de sinal da condição de espera

Ao enviar sinais para o URL gerado pela condição de espera, você deve usar o seguinte formato JSON:

```
{
  "Status" : "StatusValue",
  "UniqueId" : "Some UniqueId",
  "Data" : "Some Data",
  "Reason" : "Some Reason"
}
```

Propriedades

O campo Status pode ter um dos seguintes valores:

- SUCCESS
- FAILURE

O campo UniqueId identifica o sinal para o CloudFormation. Se a propriedade Count da condição de espera for maior que 1, o valor de UniqueId deverá ser exclusivo em todos os sinais enviados de uma determinada condição de espera; caso contrário, o CloudFormation considerará o sinal uma retransmissão dos sinais enviados anteriormente com o mesmo UniqueId e o ignorará.

O campo Data pode conter qualquer informação que você deseje enviar de volta com o sinal. Você pode acessar o valor Data chamando a função [Fn::GetAtt](#) no modelo.

O campo Reason é uma string sem nenhuma outra restrição em seu conteúdo além de conformidade com JSON.

Acessar dados de sinal

Para acessar os dados enviados por sinais válidos, você pode criar um valor de saída para a condição de espera no modelo do CloudFormation. Por exemplo:

Outputs:

WaitConditionData:

Description: *The data passed back as part of signalling the WaitCondition*

Value: !GetAtt *MyWaitCondition*.Data

Você pode visualizar esses dados usando o comando [describe-stacks](#) ou a guia Saídas do console do CloudFormation.

A função Fn::GetAtt retorna o UniqueId e os Data como um par de nome/valor em uma estrutura JSON. Por exemplo:

```
{"Signal1":"Application has completed configuration."}
```

Criar configurações de recursos reutilizáveis que podem ser incluídas em modelos com os módulos do CloudFormation

Os módulos são uma maneira de empacotar configurações de recursos para inclusão entre modelos de pilha, de modo transparente, gerenciável e repetível. Os módulos podem encapsular configurações comuns de serviço e práticas recomendadas como blocos de construção modulares e personalizáveis para você incluir em seus modelos de pilha. Os módulos permitem que você inclua configurações de recursos que incorporem práticas recomendadas, conhecimento especializado em áreas específicas e diretrizes aceitas (para áreas como segurança, conformidade, governança e regulamentações setoriais) aos seus modelos, sem precisar adquirir conhecimento profundo sobre as complexidades da implementação de recursos.

Por exemplo, um especialista na área de rede pode criar um módulo que contenha grupos de segurança integrados e regras de entrada/saída que sigam as diretrizes de segurança. Em seguida, você pode incluir esse módulo em seu modelo para provisionar infraestrutura de rede segura em sua pilha, sem precisar gastar tempo entendendo o funcionamento de VPCs, sub-redes, grupos de segurança e gateways. Além disso, como os módulos têm versionamento, se houver mudança nas diretrizes de segurança ao longo do tempo, o autor do módulo pode criar uma nova versão do módulo que incorpore essas alterações.

As características do uso de módulos em seus modelos incluem:

- **Previsibilidade:** um módulo precisa aderir ao esquema em que foi registrado no registro do CloudFormation para que você possa saber para quais recursos ele pode resolver após incluí-lo em seu modelo.
- **Reutilização:** você pode usar o mesmo módulo em vários modelos e contas.
- **Rastreabilidade:** o CloudFormation retém o conhecimento sobre quais recursos em uma pilha foram provisionados com base em um módulo, permitindo que você entenda facilmente a origem das alterações de recursos.
- **Capacidade de gerenciamento:** após registrar um módulo, você pode gerenciá-lo por meio do registro do CloudFormation, incluindo versionamento e disponibilidade de conta e regional.

Um módulo pode conter:

- Um ou mais recursos a serem provisionados com base no módulo, juntamente com quaisquer dados associados, como saídas ou condições.
- Qualquer parâmetro de módulo, permitindo que você especifique valores personalizados sempre que o módulo for usado.

Para obter informações sobre como desenvolver módulos, consulte [Desenvolver módulos](#) no Guia do usuário da CLI do CloudFormation.

Tópicos

- [Considerações sobre o uso de módulos](#)
- [Entender o versionamento de módulos](#)
- [Usar módulos do registro privado do CloudFormation](#)
- [Usar parâmetros para especificar valores de módulo](#)
- [Recursos do módulo de referência nos modelos do CloudFormation](#)

Considerações sobre o uso de módulos

- Não há cobrança adicional pelo uso de módulos. Você paga apenas pelos recursos para os quais os módulos resolvem em suas pilhas.

- As cotas do CloudFormation, como o número máximo de recursos permitidos em uma pilha ou o tamanho máximo do corpo do modelo, aplicam-se ao modelo processado independentemente de os recursos incluídos nesse modelo serem ou não provenientes de módulos. Para obter mais informações, consulte [Entender as cotas do CloudFormation](#).
- As tags especificadas para a pilha são atribuídas aos recursos individuais derivados do módulo.
- Scripts auxiliares especificados no módulo não se propagam para os recursos individuais contidos no módulo quando o CloudFormation processa o modelo.
- As saídas especificadas no módulo são propagadas para saídas no modelo.

Cada saída receberá um ID lógico que é uma concatenação do nome lógico do módulo e o nome de saída, conforme definido no módulo. Para obter mais informações, consulte [Obter resultados exportados de uma pilha do CloudFormation implantada](#).

- Os parâmetros especificados no módulo não são propagados para parâmetros no nível do modelo.

No entanto, você pode criar parâmetros no âmbito de modelos que fazem referência a parâmetros de módulo. Para obter mais informações, consulte [Usar parâmetros para especificar valores de módulo](#).

Entender o versionamento de módulos

O registro do CloudFormation atua como um repositório onde você pode registrar e gerenciar módulos para uso em sua região e Conta da AWS. Você pode registrar módulos de várias fontes, incluindo AWS, publicadores externos e suas próprias extensões personalizadas, dentro da sua conta e da sua região. Para obter mais informações, consulte [Gerenciar extensões com o registro do CloudFormation](#).

Os módulos podem ter versões diferentes para que você possa especificar qual versão do módulo deseja usar. Esse recurso de versionamento é particularmente útil quando você precisa atualizar ou modificar um módulo sem quebrar as pilhas existentes que dependem dele.

Tenha em mente as seguintes considerações ao usar várias versões de um módulo:

- Durante operações de pilha, o CloudFormation usa a versão do módulo que está registrada atualmente como padrão na região e na Conta da AWS em que a operação de pilha está sendo executada. Isso inclui módulos que estejam aninhados a outros módulos.

Portanto, lembre-se: se houver versões diferentes do mesmo módulo registradas como a versão padrão em diferentes contas ou regiões, o uso do mesmo modelo pode gerar resultados diferentes.

- Durante operações de pilha, o CloudFormation usa a versão do recurso que está registrada como a versão padrão na região e na Conta da AWS nas quais a operação de pilha está sendo executada. Isso inclui os recursos gerados pela inclusão de módulos.
- A alteração da versão padrão de um módulo não inicia nenhuma operação de atualização de pilha. No entanto, na próxima vez que você executar uma operação de pilha com qualquer modelo contendo o respectivo módulo, como uma atualização de pilha, o CloudFormation usará a nova versão padrão na operação.

A única exceção a isso é a execução de uma atualização de pilha com a opção usar modelo anterior especificada, conforme descrito abaixo.

- Para operações de atualização de pilha, se você especificar a opção usar modelo anterior, o CloudFormation usará o modelo processado anteriormente para a atualização de pilha e não reprocessará o módulo para nenhuma alteração que você possa ter feito nele.
- Para garantir resultados uniformes, se estiver incluindo módulos em um modelo de pilha para uso com conjuntos de pilha, certifique-se de que a mesma versão do módulo esteja definida como a versão padrão em todas as contas e regiões nas quais você está planejando implantar suas instâncias de pilha. Isso inclui módulos que estejam aninhados a outros módulos. Para obter mais informações, consulte [Gerenciar pilhas em todas as contas e regiões com StackSets](#).

Requisitos para ativar módulos públicos de terceiros

Para ativar com êxito um módulo público de terceiros em sua conta e região, o seguinte deve ser verdadeiro para cada extensão pública de terceiros (recurso ou módulo) incluída nesse módulo:

- Ativação da extensão: a extensão deve estar ativada na conta e na região em que você deseja usá-la. Para obter mais informações, consulte [Usar extensões públicas de terceiros do registro do CloudFormation](#).
- Registro do alias: se a extensão no módulo utilizar um alias de nome de tipo, ela deverá ser registrada na sua conta e região utilizando o mesmo alias de nome de tipo. Para obter mais informações, consulte [Usar aliases para fazer referência a extensões](#).
- Compatibilidade de versões: a versão da extensão ativada no momento deve ser uma das principais versões com suporte dessa extensão especificada no módulo.

Se você não tiver as extensões públicas de terceiros corretas e versões de extensão ativadas, o CloudFormation falhará na operação com um erro listando as extensões e versões que precisam ser ativadas antes que o módulo possa ser ativado com êxito.

Usar módulos do registro privado do CloudFormation

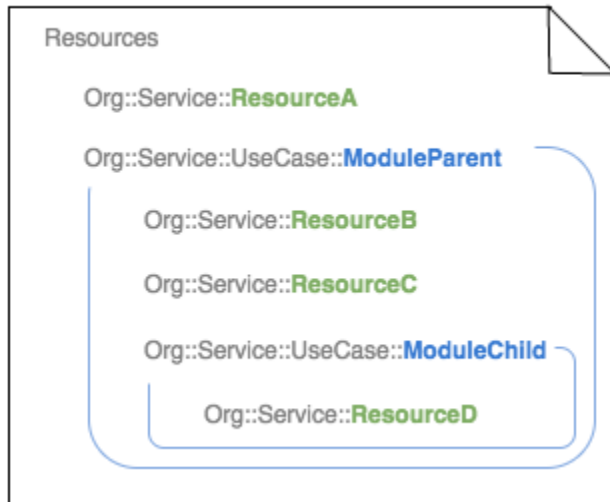
Este tópico explica como usar módulos em modelos do CloudFormation. Pense nos módulos como pacotes predefinidos de recursos que você pode adicionar aos seus modelos.

Para usar um módulo, as etapas são as seguintes:

- **Registre o módulo:** registre módulos no registro do CloudFormation como extensões privadas. Verifique se ele está registrado na região e na Conta da AWS em que você está trabalhando. Para obter mais informações, consulte [Conceitos do registro do CloudFormation](#).
- **Inclua-o em seu modelo:** adicione o módulo à seção [Resources](#) do seu modelo do CloudFormation, assim como você faria com outros recursos. Você também precisará fornecer todas as propriedades necessárias para o módulo.
- **Crie ou atualize a pilha:** quando você inicia uma operação de pilha, o CloudFormation gera um modelo processado que resolve todos os módulos incluídos nos recursos apropriados.
- **Pré-visualize as alterações:** antes de fazer alterações, você pode usar um conjunto de alterações para ver quais recursos serão adicionados ou alterados. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

Considere o exemplo a seguir: você tem um modelo que contém recursos e módulos. O modelo contém um recurso individual, `ResourceA`, e um módulo, `ModuleParent`. Esse módulo contém dois recursos, `ResourceB` e `ResourceC`, bem como um módulo aninhado, `ModuleChild`. `ModuleChild` contém um único recurso, `ResourceD`. Se você criar uma pilha com base nesse modelo, o CloudFormation o processará e resolverá os módulos para os recursos apropriados. A pilha resultante tem quatro recursos: `ResourceA`, `ResourceB`, `ResourceC` e `ResourceD`.

Template



Processed Template



O CloudFormation acompanha quais recursos em uma pilha foram criados com base em módulos. Você pode visualizar essas informações nas guias Events (Eventos), Resources (Recursos) e Drifts (Desvios) de uma determinada pilha. Elas também estão incluídas nas visualizações do conjunto de alterações.

É possível fazer a distinção entre módulos e recursos em um modelo porque eles aderem à seguinte convenção de nomenclatura com quatro partes, em contraste com a convenção habitual de três partes usada por recursos:

```
organization::service::use-case::MODULE
```

Usar parâmetros para especificar valores de módulo

No CloudFormation, é possível usar parâmetros de modelos para personalizar suas pilhas ao fornecer valores de entrada durante a criação ou atualização da pilha. Esses parâmetros permitem que você altere certos aspectos da pilha com base em suas necessidades. Para obter mais informações sobre a definição de parâmetros do modelo, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Da mesma forma, os módulos também podem ter parâmetros. Esses parâmetros de módulo permitem que você insira valores personalizados no módulo com base no modelo (ou outro módulo) que o está usando. Assim, o módulo pode usar esses valores personalizados para definir as propriedades dos recursos nele contidos.

Você também pode definir parâmetros de modelo que definem propriedades de módulo para poder inserir valores que são passados para o módulo no momento da operação de pilha.

Se um módulo contiver um módulo aninhado que possua seus próprios parâmetros do módulo, você poderá:

- Especificar os valores para os parâmetros do módulo aninhado diretamente no módulo pai.
- Definir os parâmetros correspondentes do módulo no módulo pai que permitem que os parâmetros do módulo aninhado sejam definidos pelo modelo (ou módulo) no qual o módulo pai está contido.

Como usar parâmetros de modelo para especificar valores de parâmetro de módulo

O exemplo a seguir mostra como definir parâmetros de modelo que repassam valores para um módulo.

Este modelo que contém `My::S3::SampleBucket::MODULE` define um parâmetro de modelo, `BucketName`, que permite que o usuário especifique um nome de bucket do S3 durante a operação de pilha.

```
# Template containing My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucket::MODULE'
    Properties:
      BucketName: !Ref BucketName
```

Como especificar propriedades em recursos em um módulo filho com base no módulo pai

O exemplo a seguir ilustra como especificar valores de parâmetro em um módulo que esteja aninhado dentro de outro módulo.

Este primeiro módulo, `My::S3::SampleBucketPrivate::MODULE`, será o módulo filho. Ele define dois parâmetros: `BucketName` e `AccessControl`. Os valores especificados para esses parâmetros são usados para especificar as propriedades `BucketName` e `AccessControl` do

recurso `AWS::S3::Bucket` que o módulo contém. Abaixo apresentamos o fragmento de modelo para `My::S3::SampleBucketPrivate::MODULE`.

```
# My::S3::SampleBucketPrivate::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket with Versioning and DeletionPolicy.
Parameters:
  BucketName:
    Description: Name for the bucket
    Type: String
  AccessControl:
    Description: AccessControl for the bucket
    Type: String
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref BucketName
      AccessControl: !Ref AccessControl
      DeletionPolicy: Retain
      VersioningConfiguration:
        Status: Enabled
```

Em seguida, o módulo anterior é aninhado em um módulo pai, `My::S3::SampleBucket::MODULE`. O módulo pai, `My::S3::SampleBucket::MODULE`, define os parâmetros do módulo filho das seguintes maneiras:

- Ele define o parâmetro `AccessControl` de `My::S3::SampleBucketPrivate::MODULE` para `Private`.
- Para `BucketName`, ele define um parâmetro de módulo, que permitirá que o nome do bucket seja especificado no modelo (ou módulo) que contém `My::S3::SampleBucket::MODULE`.

```
# My::S3::SampleBucket::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket. With Private AccessControl.
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
```

```
Type: 'My::S3::SampleBucketPrivate::MODULE'
Properties:
  BucketName: !Ref BucketName
  AccessControl: Private
```

Como especificar restrições para parâmetros de módulo

Os parâmetros de módulo não são compatíveis com a imposição de restrições. Para executar a verificação de restrição em um parâmetro de módulo, crie um parâmetro de modelo com as restrições desejadas. Em seguida, faça referência a esse parâmetro de modelo no seu parâmetro de módulo. Para obter mais informações sobre a definição de parâmetros do modelo, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Recursos do módulo de referência nos modelos do CloudFormation

Nos modelos do CloudFormation, muitas vezes é necessário definir as propriedades em um recurso com base no nome ou na propriedade de outro recurso. Para obter mais informações, consulte [Fazer referência a recursos](#).

Para fazer referência um recurso contido em um módulo em seu modelo do CloudFormation, é necessário combinar dois nomes lógicos:

- O nome lógico que você atribuiu ao próprio módulo ao incluí-lo em seu modelo.
- O nome lógico do recurso específico nesse módulo.

É possível combinar esses dois nomes lógicos usando ou não um ponto (.) entre eles. Por exemplo, se o nome lógico do módulo for `MyModule` e o nome lógico do recurso for `MyBucket`, você poderá se referir a esse recurso como `MyModule.MyBucket` ou `MyModuleMyBucket`.

Para encontrar os nomes lógicos dos recursos dentro de um módulo, é possível consultar o esquema do módulo, que está disponível no registro do CloudFormation, ou via operação [DescribeType](#). O esquema lista todos os recursos e seus nomes lógicos que fazem parte do módulo.

Depois de ter o nome lógico completo, você pode usar funções do CloudFormation, `GetAtt` como `Ref` e, para acessar valores de propriedade em recursos do módulo.

Por exemplo, você tem um módulo `My::S3::SampleBucket::MODULE` que contém um recurso `AWS::S3::Bucket` com o nome lógico `S3Bucket`. Para se referir ao nome desse bucket usando a função `Ref`, combine o nome do módulo em seu modelo (`MyBucket`) com o nome lógico do recurso no módulo (`S3Bucket`). O nome lógico completo é `MyBucket.S3Bucket` ou `MyBucketS3Bucket`.

Exemplo de modelo

O modelo de exemplo a seguir cria um bucket do S3 usando o módulo

`My::S3::SampleBucket::MODULE`. Ele também cria uma fila do Amazon SQS e define seu nome para ser igual ao nome do bucket do módulo. Além disso, o modelo produz o nome do recurso da Amazon (ARN) do bucket de S3 criado.

```
# Template that uses My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: My::S3::SampleBucket::MODULE
    Properties:
      BucketName: !Ref BucketName
  exampleQueue:
    Type: AWS::SQS::Queue
    Properties:
      QueueName: !Ref MyBucket.S3Bucket
Outputs:
  BucketArn:
    Value: !GetAtt MyBucket.S3Bucket.Arn
```


Gerenciando recursos da AWS como uma unidade única com pilhas do CloudFormation

Uma pilha é um conjunto de recursos da AWS que você pode gerenciar como uma unidade. Em outras palavras, você pode criar, atualizar e excluir um conjunto de recursos criando, atualizando e excluindo pilhas.

Criar uma pilha envolve implantar um modelo do CloudFormation que especifica os recursos e suas configurações, que o CloudFormation então provisiona e configura.

Atualizar uma pilha envolve fazer alterações no modelo ou nos parâmetros. O CloudFormation compara as alterações enviadas por você com o estado atual da pilha e atualiza apenas os recursos alterados. O CloudFormation pode interromper recursos ou substituir recursos atualizados, dependendo das propriedades que você atualizar. Para obter mais informações sobre os comportamentos de atualização de recursos, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

O CloudFormation fornece dois métodos para atualizar pilhas:

- **Conjuntos de alterações:** com conjuntos de alterações, você pode visualizar as alterações que o CloudFormation fará na pilha e decidir se aplica ou não essas alterações. Os conjuntos de alterações são documentos formatados por JSON que resumem as alterações que o CloudFormation fará em uma pilha. Use conjuntos de alterações quando quiser ter certeza de que o CloudFormation não fará alterações não intencionais ou quando quiser considerar várias opções. Por exemplo, é possível usar um conjunto de alterações para verificar se o CloudFormation não substituirá as instâncias de banco de dados da pilha durante uma atualização.
- **Atualização direta:** ao atualizar diretamente uma pilha, você envia alterações e o CloudFormation as implanta imediatamente. Use atualizações diretas quando você quiser implantar rapidamente as atualizações.

Excluir uma pilha exclui os recursos associados a ela. Uma pilha, por exemplo, pode incluir todos os recursos necessários para executar um aplicativo web, como um servidor web, um banco de dados e as regras de rede. Se não precisar mais do aplicativo web, você pode simplesmente excluir a pilha e todos os recursos relacionados serão excluídos.

 Note

Você será cobrado pelos recursos de pilha durante o tempo em que eles estiverem operando (mesmo que você tenha excluído a pilha imediatamente).

O CloudFormation garante que todos os recursos da pilha sejam criados ou excluídos conforme apropriado. Como o CloudFormation trata os recursos da pilha como uma unidade, é necessário que todos eles sejam criados ou excluídos com êxito para que a pilha seja criada ou excluída. Se um recurso não puder ser criado, o CloudFormation reverterá a pilha e excluirá automaticamente todos os recursos que foram criados. Se um recurso não pode ser excluído, todos os recursos restantes são mantidos até que a pilha possa ser excluída com êxito.

Tópicos

- [Interfaces para gerenciar pilhas](#)
- [Criar uma pilha no console do CloudFormation](#)
- [Visualizar informações da pilha no console do CloudFormation](#)
- [Atualizar seu modelo de pilha](#)
- [Compreender atualização de comportamentos de recursos da pilha](#)
- [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#)
- [Validar implantações de pilhas](#)
- [Atualizar pilhas diretamente](#)
- [Cancelar uma atualização de pilha](#)
- [Excluir uma pilha no console do CloudFormation](#)
- [Monitorar o progresso da pilha](#)
- [Reverta sua pilha do CloudFormation em caso de violação de alarme com acionadores de reversão](#)
- [Detectar alterações de configuração não gerenciadas em pilhas e recursos com detecção de desvios](#)
- [Importar recursos da AWS para uma pilha do CloudFormation](#)
- [Refatoração de pilhas](#)
- [Suporte a tipos de recursos](#)

- [Use links de criação rápida para criar pilhas do CloudFormation](#)
- [Exemplos de comandos de operação de pilha do CloudFormation para o AWS CLI e o PowerShell](#)

Interfaces para gerenciar pilhas

As pilhas do CloudFormation podem ser gerenciadas por meio de qualquer uma das seguintes interfaces:

- **Console do CloudFormation:** fornece uma interface web que você pode usar para acessar as pilhas. Para acessar o console do CloudFormation faça login no Console de gerenciamento da AWS, use a caixa de pesquisa na barra de navegação para pesquisar por CloudFormation e escolha o CloudFormation nos resultados da pesquisa.
- **AWS Command Line Interface:** fornece comandos para um amplo conjunto de serviços da AWS, incluindo o CloudFormation, e é compatível com Windows, Mac e Linux. Consulte informações sobre os comandos do CloudFormation em [cloudformation](#) na Referência de comandos da AWS CLI.
- **Ferramentas da AWS para PowerShell:** um conjunto de módulos do PowerShell criados com base na funcionalidade exposta pelo SDK para .NET. As ferramentas para PowerShell permitem criar scripts de operações nos recursos da AWS por meio da linha de comando do PowerShell. Você pode encontrar os cmdlets para o CloudFormation, em [Ferramentas da AWS para PowerShell Cmdlet Reference](#).
- **API de consulta:** fornece ações de API de baixo nível que são chamadas usando solicitações HTTPS. Se fizer chamadas de API na aplicação, você deverá escrever o código para tratar detalhes de baixo nível, como gerar o hash para assinar a solicitação. Consulte mais informações sobre as ações de API para o CloudFormation em [Actions](#) na Referência de API do AWS CloudFormation.
- **AWS SDKs:** fornecem APIs para linguagens de programação específicas e cuidam de muitos dos detalhes da conexão, como cálculo de assinaturas, tratamento de novas tentativas de solicitação e gerenciamento de erros. Consulte mais informações em [Ferramentas para criar com a AWS](#).
- **AWS Cloud Development Kit (AWS CDK):** o AWS CDK é um framework de desenvolvimento de software de código aberto que permite definir a infraestrutura da AWS usando linguagens de programação familiares, como TypeScript, Python, Java e .NET. Com o CDK, é possível modelar os recursos de aplicações e provisioná-los usando o CloudFormation diretamente pelo ambiente de desenvolvimento integrado (IDE). Para obter mais informações, consulte [AWS Cloud Development Kit \(AWS CDK\)](#).

Criar uma pilha no console do CloudFormation

Você pode criar um modelo de pilha e depois usá-lo para criar uma pilha usando o console do CloudFormation ou uma ferramenta de linha de comando. O console oferece uma interface guiada por assistente com opções predefinidas para agilizar o processo de criação de pilhas.

Tópicos

- [Criar uma pilha](#)
- [Configurar opções da pilha](#)
- [Visualizar a configuração da sua pilha](#)

Criar uma pilha

Siga as etapas desta seção para implantar seu modelo e criar uma pilha.

Pré-requisitos

- Você deve ter criado um modelo de pilha. Para obter mais informações, consulte [Trabalhar com modelos do CloudFormation](#).

Para criar uma pilha de (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual a pilha deve ser criada.
3. Na página Pilhas, escolha Criar pilha no canto superior direito e selecione Com novos recursos (padrão).

Se preferir, escolha a opção Com recursos existentes (importar recursos) para importar os recursos da AWS existentes descritos no modelo. Para obter mais informações sobre essa opção, consulte [Importar recursos da AWS para uma pilha do CloudFormation](#).

4. Na página Criar uma pilha, faça o seguinte:
 - Para usar um modelo existente, em Pré-requisito: preparar modelo, escolha Escolher um modelo existente. Em seguida, em Especificar modelo, escolha URL do Amazon S3 ou Carregar um arquivo de modelo com base na localização do modelo.

- Se você escolher URL do Amazon S3, forneça uma URL para o arquivo de modelo em um bucket do S3.

Se seu modelo incluir pilhas aninhadas (por exemplo, pilhas descritas em outros documentos de modelos localizados em subdiretórios), verifique se seu bucket do S3 contém os arquivos e diretórios necessários.

Se você tiver um modelo de um bucket habilitado para controle de versão, poderá especificar uma versão específica do modelo anexando `?versionId=version-id` à URL. Para obter mais informações sobre buckets habilitados por controle de versão, consulte [Trabalhar com objetos em um bucket habilitado com controle de versão](#), no Guia do usuário do Amazon Simple Storage Service.

O URL deve apontar para um modelo com um tamanho máximo de 1 MB que esteja armazenado em um bucket do S3 para o qual você tenha permissões de leitura. O URL pode ter no máximo 1.024 caracteres. Alguns recursos podem exigir que o bucket esteja na mesma região que a pilha.

- Se escolher Carregar um arquivo de modelo, escolha Escolher arquivo para escolher um arquivo de modelo no computador local. O arquivo de modelo deve ter 1 MB ou menos.

Após escolher o modelo, o CloudFormation faz upload do arquivo e exibe o URL do S3. O CloudFormation o carrega em um bucket do Amazon S3 na sua Conta da AWS. Se você já tiver um bucket do S3 que foi criado pelo CloudFormation em sua Conta da AWS, o CloudFormation adicionará o modelo a esse bucket. Se você ainda não tiver um bucket criado pelo CloudFormation, ele criará um bucket exclusivo para cada região em que você carregar um arquivo de modelo.

Veja a seguir algumas considerações ao usar buckets do S3 criados pelo CloudFormation:

- Os buckets são acessíveis a qualquer pessoa com permissões do Amazon S3 na sua conta Conta da AWS.
- O CloudFormation cria os buckets com a criptografia no lado do servidor habilitada por padrão, criptografando, assim, todos os objetos armazenados no bucket.

Gerencie diretamente as opções de criptografia para buckets criados pelo CloudFormation usando, por exemplo, o console do Amazon S3 em <https://console.aws.amazon.com/s3/>, ou a AWS CLI. Para obter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

- É possível utilizar seu próprio bucket e gerenciar suas permissões carregando modelos manualmente no Amazon S3. Ao criar ou atualizar uma pilha, especifique o URL de um arquivo de modelo do Amazon S3.
- Se tiver um modelo pronto, você poderá escolher Compilar do Infrastructure Composer para criar um modelo com o Infrastructure Composer. Para obter mais informações, consulte [Compositor de Infraestrutura da AWS](#).

5. Escolha Avançar para continuar e validar o modelo.

Antes de continuar, o CloudFormation valida seu modelo para verificar erros sintáticos e alguns semânticos, como dependências circulares. Durante a validação, o CloudFormation primeiro verifica se o modelo é JSON válido. Caso não seja, o CloudFormation verifica se o modelo é YAML válido. Se as duas verificações falharem, o CloudFormation retornará um erro de validação de modelo.

6. Na página Specify stack details (Especificar detalhes da pilha), digite o nome de uma pilha na caixa Stack name (Nome da pilha).

O nome da pilha é um identificador que ajuda a encontrar uma determinada pilha em uma lista de pilhas. Um nome de pilha pode conter apenas caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 128 caracteres.

7. Na seção Parâmetros, especificam valores para os parâmetros que foram definidos no modelo.

8. Escolha Avançar para continuar criando a pilha.

9. (Opcional) Na página Configurar opções de pilha, altere as opções de pilha padrão. Para obter mais informações, consulte [Configurar opções da pilha](#).

10. Se o seu modelo contém recursos do IAM, em Recursos, escolha Eu reconheço que este modelo pode criar recursos do IAM para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).

11. Escolha Próximo para continuar.

12. Na página Revisar e criar, revise os detalhes de sua pilha.

Para alterar qualquer um dos valores antes de iniciar a pilha, selecione Editar na seção que contém a configuração que deseja alterar.

13. (Opcional) É possível criar um conjunto de alterações para visualizar a configuração da pilha do antes de criá-la. Na página Revisar e criar, escolha Criar conjunto de alterações e siga as instruções. Para obter mais informações, consulte [Visualizar a configuração da sua pilha](#).
14. Escolha Enviar para lançar sua pilha.

O CloudFormation criará todos os recursos definidos no modelo.

Você pode monitorar o progresso e o status da criação da pilha na guia Eventos da nova pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Para criar uma pilha usando a linha de comando

Você pode usar um dos comandos a seguir:

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

Para obter exemplos de uso da linha de comando para criar uma pilha, consulte [Exemplos de comandos de operação de pilha do CloudFormation para o AWS CLI e o PowerShell](#).

Configurar opções da pilha

Na página Configurar opções de pilha, é possível configurar opções para as pilhas do CloudFormation, como etiquetas, notificação de eventos de pilha, ou uma política de pilha.

É possível definir as seguintes opções de pilha:

Tags

É possível adicionar até 50 pares de chave de etiquetas à sua pilha e a qualquer recurso para o qual o CloudFormation suporte a marcação. Uma etiqueta é uma chave e um valor definidos pelo cliente que podem ser atribuídos aos recursos da AWS para fins como controle de custos.

Uma Chave consiste em quaisquer caracteres alfanuméricos ou espaços. As chaves de tag podem ter até 127 caracteres.

Um Valor consiste em quaisquer caracteres alfanuméricos ou espaços. Os valores de tags podem ter até 255 caracteres.

Após a criação da pilha, adicionar, atualizar ou remover tags em nível de pilha iniciará uma atualização na pilha. Todos os recursos compatíveis com propagação de tags em nível de pilha também serão atualizados.

permissões do

Um perfil de serviço existente do IAM que o CloudFormation pode assumir. Em vez de usar as credenciais de sua conta, o CloudFormation usa as credenciais da função para criar a pilha. Para obter mais informações, consulte [Função de serviço do CloudFormation](#).

Stack failure options (Opções de falha da pilha)

Especifica as opções de falha de provisionamento de todas as implantações de pilha e operações de conjunto de alterações. Para obter mais informações, consulte [Escolha como lidar com falhas ao provisionar recursos](#).

A opção Roll back all stack resources (Reverter todos os recursos da pilha) reverterá todos os recursos especificados no modelo quando o status da pilha for CREATE_FAILED ou UPDATE_FAILED.

Para operações de criação, a opção Preserve successfully provisioned resources (Preservar recursos provisionados com sucesso) preserva o estado dos recursos bem-sucedidos, enquanto os recursos com falha permanecerão em um estado de falha até que a próxima operação de atualização seja realizada.

Para operações de atualização e conjunto de alterações, a opção Preserve successfully provisioned resources (Preservar recursos provisionados com sucesso) preserva o estado dos recursos bem-sucedidos e reverte os recursos com falha para o último estado estável conhecido. Os recursos com falha estarão em um estado de UPDATE_FAILED. Os recursos sem um último estado estável conhecido serão excluídos na próxima operação de pilha.

Também é possível definir as seguintes opções avançadas para a criação da pilha:

Política de pilha

Define os recursos que você deseja proteger contra atualizações não intencionais durante uma atualização da pilha. Por padrão, todos os recursos podem ser atualizados durante uma atualização da pilha.

É possível inserir a política de pilha diretamente como JSON ou fazer upload de um arquivo JSON que contém a política de pilha. Para obter mais informações, consulte [Impedir atualizações nos recursos de pilha](#).

Configuração de reversão

Você pode fazer com que o CloudFormation monitore o estado de sua pilha durante a criação e atualização de pilhas, e que você reverta essa operação se a pilha violar os limites de qualquer um dos alarmes especificados. Especifique os alarmes do CloudWatch que o CloudFormation deve monitorar. Se qualquer um dos alarmes entrar no estado ALARM durante a operação de pilha ou o período de monitoramento, o CloudFormation reverterá toda a operação de pilha. Para obter mais informações, consulte [Reverta sua pilha do CloudFormation em caso de violação de alarme com acionadores de reversão](#).

Opções de notificação

É possível especificar um tópico novo ou existente do Amazon Simple Notification Service para onde as notificações sobre eventos de pilha são enviadas.

Se você criar um tópico do Amazon SNS, é necessário especificar um nome e um endereço de e-mail para onde as notificações de eventos de pilha deverão ser enviadas.

Stack creation options (Opções de criação de pilha)

As opções a seguir estão inclusas para a criação da pilha, mas não estão disponíveis como parte das atualizações dela.

Timeout (Tempo limite)

Especifica a quantidade de tempo, em minutos, que o CloudFormation deve alocar antes de exceder o tempo limite das operações de criação de pilha. Se o CloudFormation não puder criar a pilha inteira dentro do tempo alocado, além de não ser criada, a pilha será revertida.

Por padrão, não há tempo limite para criação da pilha. No entanto, recursos individuais podem ter seus próprios limites de tempo com base na natureza do serviço que implementam. Por exemplo, se um recurso individual em sua pilha expirar, a criação dela também expirará, ainda que o tempo limite especificado para a criação não tenha sido atingido.

Termination protection

Impede que uma pilha seja excluída acidentalmente. Se um usuário tentar excluir uma pilha com proteção contra encerramento ativada, a exclusão falhará e a pilha, incluindo o status dela, permanecerá inalterada. Para obter mais informações, consulte [Proteger pilhas do CloudFormation contra exclusão](#).

Por padrão, a proteção contra encerramento está Disabled (Desabilitada).

Visualizar a configuração da sua pilha

Para visualizar como uma pilha do CloudFormation será configurada antes de criar a pilha, crie um conjunto de alterações. Essa funcionalidade permite que você examine várias configurações e faça correções e alterações na pilha antes de executar o conjunto de alterações. Para obter mais informações sobre conjuntos de alterações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

Criar um conjunto de alterações para uma pilha nova

Para criar um conjunto de alterações para uma pilha nova, selecione o modelo de pilha e especifique a configuração da pilha como você faria se estivesse criando uma nova pilha e escolha criar um novo conjunto de alterações em vez de uma nova pilha.

Para criar um conjunto de alterações para uma nova pilha

1. Na página Revisar e criar, escolha Criar conjunto de alterações.
2. Na caixa de diálogo Criar conjunto de alterações, insira um nome para o conjunto de alterações e uma descrição, se desejado. Escolha Criar conjunto de alterações.

Quando você cria um conjunto de alterações para uma nova pilha, o CloudFormation faz o seguinte:

- Inicia uma nova pilha com um status de REVIEW_IN_PROGRESS.
- Cria um conjunto de alterações para a nova pilha que reflete a configuração da pilha especificada nas etapas anteriores.

O CloudFormation exibe a página Change sets (Conjuntos de alterações) para a pilha proposta. Enquanto o CloudFormation cria o conjunto de alterações, o status é CREATE_IN_PROGRESS, e o status de execução é UNAVAILABLE. Quando o CloudFormation conclui com êxito a criação do conjunto de alterações, ele define o status do conjunto de alterações para CREATE_COMPLETE e o status de execução é AVAILABLE. O status da pilha é atualizado para REVIEW_IN_PROGRESS. Nesse ponto, você pode executar o conjunto de alterações para concluir a criação da nova pilha.

No painel Changes (Alterações), o CloudFormation exibe a configuração proposta da pilha.

Se o CloudFormation não conseguir criar o conjunto de alterações, ele definirá o status para `CREATE_FAILED`. Corrija o erro exibido no campo Status reason (Motivo do status) e crie um novo conjunto de alterações. Nesse estágio, você pode tentar várias configurações e fazer correções e alterações na pilha antes de executar o próximo conjunto de alterações.

3. Para concluir a criação de uma nova pilha com base no conjunto de alterações, escolha Execute (Executar), especifique sua configuração de reversão e depois escolha Execute change set (Executar conjunto de alterações).

Visualizar informações da pilha no console do CloudFormation

Depois de criar uma pilha do CloudFormation, você pode usar o Console de gerenciamento da AWS para visualizar seus dados e recursos.

Para visualizar informações sobre a pilha do CloudFormation

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha está localizada.
3. Na página Pilhas, escolha o nome da pilha. O CloudFormation exibe a página de detalhes da pilha para a pilha selecionada.

Note

Se a pilha tiver sido excluída, você poderá encontrá-la usando a opção Filtrar status. Para obter mais informações, consulte [Visualizar pilhas excluídas usando o console do CloudFormation](#).

Você pode visualizar as seguintes informações da pilha:

Informações da pilha

Exibe informações gerais sobre a pilha e sua configuração, incluindo:

Visão geral

Exibe nome da pilha, ID da pilha, pilha raiz e [perfil do IAM](#), juntamente com informações de status, como status da pilha, status de desvio e proteção contra encerramento.

Tags

Exibe todas as tags associadas à pilha.

Política de pilha

Descreve os recursos da pilha que são protegidos contra atualizações da pilha. Para poder atualizar esses recursos, eles devem ser permitidos explicitamente durante uma atualização de pilha. Para obter mais informações, consulte [Impedir atualizações nos recursos de pilha](#).

Configuração de reversão

Exibe todos os alarmes do CloudWatch especificados que o CloudFormation deve monitorar durante a operação da pilha ou o período de monitoramento especificado. Se qualquer um dos alarmes entrar no estado ALARM durante a operação de pilha ou o período de monitoramento, o CloudFormation reverterá toda a operação de pilha. Para obter mais informações, consulte [Reverta sua pilha do CloudFormation em caso de violação de alarme com acionadores de reversão](#).

Opções de notificação

Exibe o tópico do Amazon Simple Notification Service em que as notificações sobre eventos de pilha são enviadas, se especificado.

Eventos

Exibe as operações que são acompanhadas ao criar, atualizar ou excluir a pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Todos os eventos que são acionados por uma determinada operação de pilha são atribuídos ao mesmo token de solicitação do cliente, o qual você pode usar para rastrear as operações. Operações de pilha iniciadas a partir do console usam o formato de token Console-StackOperation-ID, que ajuda você a identificar facilmente a operação da pilha. Por exemplo, se você criar uma pilha usando o console, cada evento de pilha resultante receberia o mesmo token no seguinte formato: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Recursos

Exibe os recursos que fazem parte da pilha.

Saídas

Exibe as saídas que foram declaradas no modelo da pilha. Para obter mais informações, consulte [Obter resultados exportados de uma pilha do CloudFormation implantada](#).

Parâmetros

Exibe os parâmetros da pilha e seus valores.

Para pilhas que contêm parâmetros do Systems Manager, a coluna Valor resolvido exibe os valores usados na definição da pilha para os parâmetros do Systems Manager. Para obter mais informações, consulte [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#).

Modelo

Exibe o modelo da pilha.

Para pilhas que contêm macros, escolha View original template (Visualizar modelo original) para visualizar o modelo enviado pelo usuário ou View processed template (Visualizar modelo processado) para visualizar o modelo depois que o CloudFormation processou as macros referenciadas. O CloudFormation usa o modelo processado para criar ou atualizar sua pilha.

Conjuntos de alterações

Exibe os conjuntos de alterações da pilha.

Para obter mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#).

Git sync

Exibe o painel de Git sync da pilha.

Para obter mais informações, consulte [Painel de status do Git Sync](#).

Atualizar seu modelo de pilha

Para modificar os recursos ou propriedades em uma pilha do CloudFormation, você deve atualizar o modelo da pilha. Comece com o modelo existente para essa pilha e faça suas alterações nele. Se você tiver o modelo armazenado em um sistema de controle de origem, use uma cópia dele como ponto de partida. Ou então, você pode obter uma cópia de um modelo do CloudFormation.

Se você quiser apenas alterar os parâmetros ou as configurações da pilha (como o tópico Amazon SNS de uma pilha), poderá reutilizar o modelo existente sem obter uma cópia.

É possível atualizar um modelo de pilha do CloudFormation usando um editor de texto ou o [Compositor de Infraestrutura da AWS](#).

Para atualizar um modelo de pilha existente usando o Infrastructure Composer

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, escolha o nome da pilha para atualizar.
3. Escolha a guia Modelo e, em seguida, escolha Visualizar no Infrastructure Composer.

O CloudFormation abre o modelo no Infrastructure Composer.

4. Atualize seu modelo usando um dos seguintes métodos:
 - Interface Canvas: aqui, você pode arrastar e soltar da paleta Recursos. Configure os recursos clicando duas vezes em um cartão para abrir o painel Propriedades do recurso. Conecte os recursos conforme necessário. Para obter instruções detalhadas sobre como usar a interface Canvas, consulte [Como compor no AWS Infrastructure Composer](#).
 - Interface Modelo: alterne da interface Canvas para a interface Modelo. Faça atualizações em linha no código do modelo. Alterne entre os formatos JSON e YAML conforme necessário.
5. Escolha Validar para verificar erros de sintaxe no modelo.
6. Quando você estiver pronto para exportar as alterações para o CloudFormation, escolha Atualizar modelo.

Para atualizar um modelo de pilha existente usando a AWS CLI

1. Para obter o modelo para a pilha que você deseja atualizar, use o comando [get-template](#) da CLI.
2. Copie o modelo, cole-o em um arquivo de texto, modifique-o e salve-o. Copie apenas o modelo. O comando inclui o modelo entre aspas, mas não copie as aspas em torno do modelo. O modelo em si começa com uma chave de abertura e termina com a chave de fechamento final. Especifique as alterações nos recursos da pilha nesse arquivo.

Lembre-se dos seguintes pontos ao fazer alterações no modelo:

- Você não pode adicionar, modificar ou excluir um parâmetro que é usado por um recurso que não aceite atualizações.
- Para a maioria dos recursos, a alteração do nome lógico de um recurso é equivalente a excluir esse recurso e substituí-lo por um novo. Todos os outros recursos que dependem do recurso renomeado também precisam ser atualizados e isso pode fazer com que eles sejam substituídos. Outros recursos exigem que você atualize uma propriedade (não apenas o nome lógico) para iniciar uma atualização.
- Alguns recursos podem ter restrições sobre quais valores é possível definir para determinadas propriedades. Por exemplo, as alterações na propriedade `AllocatedStorage` de uma instância de banco de dados do RDS devem ser maiores que o valor atual. Se a atualização violar essas regras, essa parte falhará.
- A atualização de um recurso também pode afetar outros que fazem referência a ele. Se você usar funções como [A função Ref](#) ou [A função Fn::GetAtt](#) para definir uma propriedade com base em outro recurso, o CloudFormation também atualizará o recurso de referência quando o referenciado for alterado.
- Para obter informações sobre os efeitos da atualização de propriedades de recursos específicos, consulte a [Referência de tipos de propriedades e recursos da AWS](#). Para cada propriedade, os efeitos de uma atualização serão um dos seguintes:
 - Requisitos da atualização: [sem interrupção](#)
 - Atualização necessária: [Algumas interrupções](#)
 - Requisitos da atualização: [substituição](#)
- É possível verificar a sintaxe JSON ou YAML do seu modelo usando o comando [validate-template](#) da CLI ou especificando o modelo no console. O console realiza a validação automaticamente. No entanto, esses métodos apenas verificam a sintaxe do seu modelo e não validam se os valores de propriedade especificados para um recurso são válidos para esse recurso. Para validações mais complexas ou para verificar a aplicação das práticas recomendadas, você também pode usar ferramentas adicionais, como o [CloudFormation Linter \(cfn-lint\)](#) e o [CloudFormation Rain \(rain fmt\)](#).

Note

Às vezes, o CloudFormation não permite certas alterações que você tenta fazer e diz que a alteração não é permitida. Porém, essa mensagem pode ocorrer de modo assíncrono, porque, por padrão, os recursos são criados e atualizados pelo CloudFormation em uma ordem não determinística.

Compreender atualização de comportamentos de recursos da pilha

Quando você envia uma atualização, o CloudFormation atualiza os recursos com base nas diferenças entre o que você envia e o modelo atual da pilha. Os recursos que não foram alterados funcionam sem interrupção durante o processo de atualização. Para recursos atualizados, o CloudFormation usa um dos seguintes comportamentos de atualização:

Atualizações sem interrupções

CloudFormationO atualiza o recurso sem interromper a operação do recurso e sem alterar o ID físico do recurso. Por exemplo, se você atualizar determinadas propriedade de um recurso [AWS::CloudTrail::Trail](#), o CloudFormation atualiza a trilha sem interrupções.

Atualizações com alguma interrupção

CloudFormationO atualiza o recurso com interrupções. Por exemplo, se você atualizar determinadas propriedades de um recurso [AWS::EC2::Instance](#), a instância poderá ter alguma interrupção enquanto o CloudFormation e o Amazon EC2 reconfiguram a instância.

Substituição

O CloudFormation recria o recurso durante uma atualização, o que também gera um novo ID físico. Normalmente o CloudFormation cria o recurso de substituição primeiro, altera as referências de outros recursos dependentes para apontar para o recurso substituto e, em seguida, exclui o recurso antigo. Por exemplo, se você atualizar a propriedade `AvailabilityZone` de um tipo de recurso [AWS::EC2::Instance](#), o CloudFormation cria um novo recurso e substitui o recurso de instância do EC2 atual pela nova.

Se você estiver adicionando ou removendo uma propriedade que requer substituição, isso também acionará uma atualização. Essa atualização acontecerá mesmo se o valor real da propriedade não mudar.

O método que o CloudFormation usa depende da propriedade que você atualiza para um determinado tipo de recurso. O comportamento da atualização de cada propriedade é descrito na [Referência de tipos de propriedades e recursos da AWS](#).

Dependendo do comportamento da atualização, você pode decidir quando modificar os recursos para reduzir o impacto dessas alterações em seu aplicativo. Especificamente, você pode planejar quando os recursos devem ser substituídos durante uma atualização. Por exemplo, se você atualizar a propriedade `Port` de um tipo de recurso [AWS::RDS::DBInstance](#), o CloudFormation substituirá a

instância de banco de dados criando uma nova instância de banco de dados com a configuração de porta atualizada e excluirá a instância de banco de dados antiga. Antes da atualização, você pode planejar executar as seguintes atividades para se preparar para a substituição do banco de dados:

- Faça um snapshot dos bancos de dados atuais.
- Prepare uma estratégia de como os aplicativos que usam essa instância de banco de dados irão lidar com uma interrupção enquanto a instância de banco de dados está sendo substituída.
- Verifique se os aplicativos que usam a instância de banco de dados consideram a configuração da porta atualizada e todas as outras atualizações que você tiver feito.
- Use o DB snapshot para restaurar os bancos de dados na nova instância de banco de dados.

Este exemplo não abrange todos os casos, tendo a intenção de dar uma ideia do que é necessário planejar quando um recurso for substituído durante uma atualização.

Note

Se o modelo incluir uma ou mais [pilhas aninhadas](#), o CloudFormation também iniciará uma atualização de cada pilha aninhada. Isso é necessário para determinar se as pilhas aninhadas foram modificadas. O CloudFormation atualiza apenas os recursos nas pilhas aninhadas que têm alterações especificadas nos modelos correspondentes.

Atualizar pilhas do CloudFormation usando conjuntos de alterações

Quando você precisa atualizar uma pilha, entender como as alterações afetarão os recursos em execução antes de implementá-las pode ajudar você a atualizar pilhas com confiança. Os conjuntos de alterações permitem que você visualize como as alterações propostas para uma pilha poderiam afetar os recursos em execução, incluindo seu impacto nas propriedades e atributos dos recursos. Não importa se as alterações vão excluir ou substituir recursos críticos, o CloudFormation só fará as alterações na pilha quando você decidir executar o conjunto de alterações, o que permite a você decidir se deseja continuar com as alterações propostas ou explorar outras alterações criando outro conjunto de alterações. Você pode criar e gerenciar conjuntos de alterações usando o console do CloudFormation, a AWS CLI ou a API do CloudFormation.

Tópicos

- [Criar um conjunto de alterações para uma pilha do CloudFormation](#)

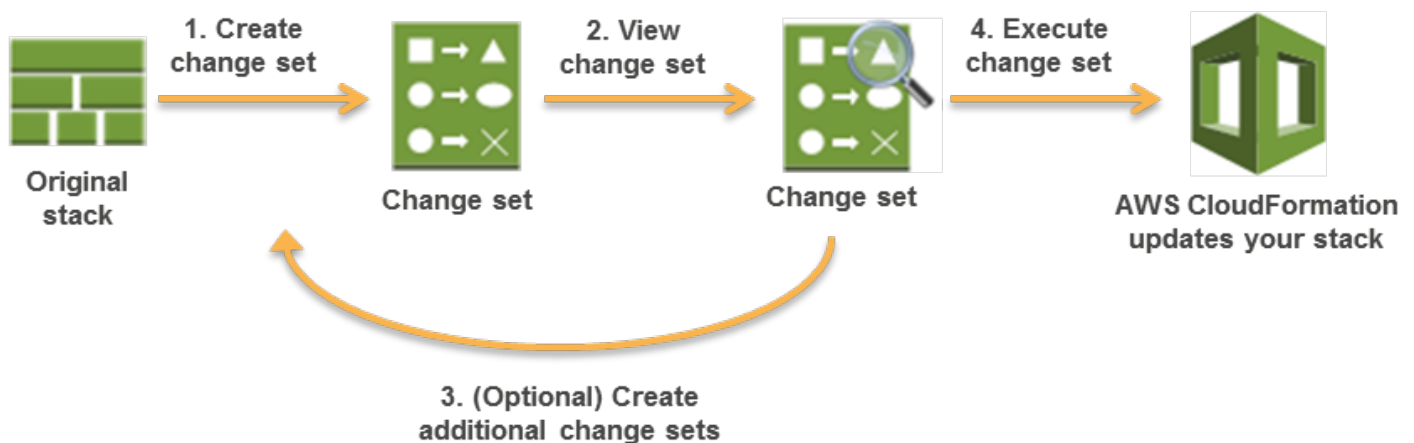
- [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#)
- [Uso de conjuntos de alterações sensíveis a desvios](#)
- [Executar um conjunto de alterações para uma pilha do CloudFormation](#)
- [Excluir um conjunto de alterações para uma pilha do CloudFormation](#)
- [Exemplos de conjuntos de alterações para pilhas do CloudFormation](#)
- [Conjuntos de alteração para pilhas aninhadas](#)

Important

Os conjuntos de alterações não indicam se o CloudFormation atualizará uma pilha com êxito. Por exemplo, um conjunto de alterações não verifica se você excederá a cota de uma conta, se você está atualizando um recurso que não oferece suporte a alterações ou se você tem permissões suficientes para modificar um recurso, e tudo isso pode provocar uma falha na atualização da pilha. Se uma atualização falhar, o CloudFormation tentará reverter seus recursos para o estado original.

Visão geral do conjunto de alterações

O diagrama a seguir resume como usar conjuntos de alterações para atualizar uma pilha:



1. Crie um conjunto de alterações enviando as alterações para a pilha que você deseja atualizar. Você pode enviar um modelo de pilha modificado ou valores de parâmetro de entrada modificados. O CloudFormation compara a pilha com as alterações que você enviou para gerar o conjunto de alterações. Ele não faz alterações em sua pilha neste momento.

2. Visualize o conjunto de alterações para conferir quais recursos e configurações da pilha serão alterados. Por exemplo, você pode ver quais recursos o CloudFormation vai adicionar, modificar ou excluir. Além disso, você pode ver uma comparação entre o antes e o depois das propriedades e atributos dos recursos, como as tags, que o CloudFormation modificará.
3. Opcional: se deseja considerar outras alterações antes de decidir quais irá fazer, crie outros conjuntos de alterações. Criar vários conjuntos de alterações ajuda você a compreender e avaliar como as diferentes alterações afetarão os recursos e as propriedades. Você poderá criar quantos conjuntos de alterações forem necessários.
4. Execute o conjunto com as alterações que deseja aplicar à pilha. O CloudFormation atualiza a pilha com essas alterações.

 Note

Depois de executar uma alteração, o CloudFormation remove todos os conjuntos de alterações que estão associados à pilha, já que não são aplicáveis à pilha atualizada.

Também é possível excluir conjuntos de alterações para impedir a execução de um que não deva ser aplicado.

Criar um conjunto de alterações para uma pilha do CloudFormation

Para criar um conjunto de alterações para uma pilha em execução, envie as alterações que deseja fazer fornecendo um modelo modificado, novos valores de parâmetros de entrada, ou ambos. O CloudFormation gera um conjunto de alterações comparando sua pilha com as alterações enviadas.

É possível modificar um modelo antes de criar o conjunto de alterações ou durante a criação do conjunto de alterações.

Create a change set (console)

Para criar um conjunto de alterações

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, escolha a pilha em execução para a qual você deseja criar um conjunto de alterações.

4. No painel de detalhes da pilha, escolha Ações de pilha e selecione Criar um conjunto de alterações.
5. Na página Create change set for **stack-name** (Criar conjunto de alterações para nome-pilha), execute um dos seguintes procedimentos para modificar os valores do parâmetro de entrada, especificar a localização de um modelo de atualização ou modificar o modelo:

Tarefa	Ação
Como modificar os valores do parâmetro de entrada	Escolha Usar modelo existente e depois Avançar para prosseguir e inserir ou modificar os valores dos parâmetros de entrada.
Como especificar a localização de um modelo atualizado	Se você modificou o modelo, escolha Substituir modelo existente e faça uma das seguintes alternativas: • Para um modelo armazenado no bucket do Amazon S3, escolha Amazon S3 URL (URL do Amazon S3). Digite ou cole o URL do modelo e escolha Next (Próximo). Se você tiver um modelo em um bucket com versionamento habilitado, poderá especificar uma versão específica do modelo anexando <code>?versionId= version-id</code> ao URL. Para obter mais informações, consulte Trabalhar com objetos em um bucket com versionamento habilitado no Guia do usuário do Amazon Simple Storage Service. • Para um modelo armazenado localmente em seu computador, escolha Upload a template file (Fazer upload de um arquivo de modelo). Escolha Choose File (Escolher arquivo) para navegar até o arquivo e selecioná-lo, depois escolha Next (Próximo).
Como modificar o modelo	Se não tiver modificado o modelo, escolha Editar modelo no Infrastructure Composer e, em seguida, escolha Editar no Infrastructure Composer. Você será redirecionado para o AWS Infrastructure Composer. Após modificar o modelo, escolha Criar conjunto de alterações e depois Confirmar e continuar no

Tarefa	Ação
	CloudFormation para voltar à página Criar conjunto de alterações para a <i>stack-name</i> , em seguida, escolha Avançar.

- Na página Especificar detalhes da pilha, especifique um nome para o conjunto de alterações e, opcionalmente, especifique uma descrição do conjunto de alterações para identificar sua finalidade na seção Visão geral. Se o seu modelo contém parâmetros, na página Specify stack details (Especificar detalhes da pilha), insira ou modifique os valores do parâmetro de entrada aplicáveis e escolha Next (Próximo).

Se você estiver reutilizando o modelo da pilha, o CloudFormation preencherá cada parâmetro com o valor atual nela, com exceção dos parâmetros declarados com o atributo NoEcho. Para usar os valores existentes para esses parâmetros, selecione Use existing value (Usar valor existente).

Para obter mais informações sobre o uso de NoEcho para ocultar informações confidenciais e sobre o uso de parâmetros dinâmicos para gerenciar segredos, consulte a prática recomendada [Não incorporar credenciais em seus modelos](#).

- Na página Configurar opções da pilha, atualize as tags da pilha, o perfil de serviço do IAM, a política da pilha, a configuração de reversão, o tópico de notificação do Amazon SNS (se aplicável) ou conjunto de alterações.

Note

Os conjuntos de alterações para pilhas aninhadas são habilitados por padrão, o que criará conjuntos de alterações para todas as pilhas aninhadas especificadas em seu modelo. Para criar um conjunto de alterações apenas para a pilha atual, escolha Disabled (Desabilitado). Para obter mais informações sobre os conjuntos de alterações para pilhas aninhadas, consulte [Conjuntos de alteração para pilhas aninhadas](#).

- Se o modelo incluir recursos do IAM, escolha Reconheço que o CloudFormation pode criar recursos do IAM em Recursos. Os recursos do IAM podem modificar as permissões na sua conta da AWS. Analise esses recursos para garantir que você está permitindo apenas as ações que deseja. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
- Escolha Próximo para continuar.

10. Na página Review **stack-name** (Revisar nome-pilha), revise as alterações para esse conjunto de alterações.
11. Selecione Enviar.

Você será redirecionado para a guia Changes (Alterações) da página de detalhes do conjunto de alterações. Enquanto o CloudFormation está gerando o conjunto de alterações, o status do conjunto de alterações é CREATE_PENDING. Após criar o conjunto de alterações, o CloudFormation define o status como CREATE_COMPLETE. Na seção Alterações, o CloudFormation relaciona todas as alterações que serão feitas em sua pilha. Para obter mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#).

Escolha Visualizar detalhes na coluna Alterações no nível da propriedade para visualizar as alterações feitas no nível da propriedade.

Se o CloudFormation falhar na criação do conjunto de alterações (exibindo o status FAILED), corrija o erro exibido no campo Status e recrie o conjunto de alterações.

12. Depois de confirmar que as alterações parecem estar corretas, escolha Executar conjunto de alterações

Create a change set for nested stacks (console)

Para criar um conjunto de alterações para pilhas aninhadas

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, selecione a pilha em execução para a qual você deseja criar um conjunto de alterações.
4. No painel de detalhes da pilha, escolha Ações de pilha e selecione Criar um conjunto de alterações.
5. Na página Create change set for **stack-name** (Criar conjunto de alterações para nome-pilha), execute um dos seguintes procedimentos para modificar os valores do parâmetro de entrada, especificar a localização de um modelo de atualização ou modificar o modelo:

Tarefa	Ação
Como modificar os valores do parâmetro de entrada	Escolha Usar modelo existente e depois Avançar para prosseguir e inserir ou modificar os valores dos parâmetros de entrada.
Como especificar a localização de um modelo atualizado	Se você modificou o modelo, escolha Substituir modelo existente e faça uma das seguintes alternativas: - Para um modelo armazenado no bucket do Amazon S3, escolha Amazon S3 URL (URL do Amazon S3). Digite ou cole o URL do modelo e escolha Next (Próximo). Se você tiver um modelo em um bucket com versionamento habilitado, poderá especificar uma versão específica do modelo anexando <code>?versionId= <i>version-id</i></code> ao URL. Para obter mais informações, consulte Trabalhar com objetos em um bucket com versionamento habilitado no Guia do usuário do Amazon Simple Storage Service. - Para um modelo armazenado localmente em seu computador, escolha Upload a template file (Fazer upload de um arquivo de modelo). Escolha Choose File (Escolher arquivo) para navegar até o arquivo e selecioná-lo, depois escolha Next (Próximo).
Como modificar o modelo	Se não tiver modificado o modelo, escolha Editar modelo no Infrastructure Composer e, em seguida, escolha Editar no Infrastructure Composer. Você será redirecionado para o AWS Infrastructure Composer. Após modificar o modelo, escolha Criar conjunto de alterações e depois Confirmar e continuar no CloudFormation para voltar à página Criar conjunto de alterações para a <i>stack-name</i> , em seguida, escolha Avançar.

- Na página Especificar detalhes da pilha, especifique um nome para o conjunto de alterações e, opcionalmente, especifique uma descrição do conjunto de alterações para identificar sua finalidade na seção Visão geral. Se o seu modelo contém parâmetros, na página Specify

stack details (Especificar detalhes da pilha), insira ou modifique os valores do parâmetro de entrada aplicáveis e escolha Next (Próximo).

Se você estiver reutilizando o modelo da pilha, o CloudFormation preencherá cada parâmetro com o valor atual nela, com exceção dos parâmetros declarados com o atributo NoEcho. Para usar os valores existentes para esses parâmetros, selecione Use existing value (Usar valor existente).

Para obter mais informações sobre o uso de NoEcho para ocultar informações confidenciais e sobre o uso de parâmetros dinâmicos para gerenciar segredos, consulte a melhor prática [Não incorporar credenciais em seus modelos](#).

7. Na página Configurar opções da pilha, atualize as tags da pilha, o perfil de serviço do IAM, a política da pilha, a configuração de reversão, o tópico de notificação do Amazon SNS (se aplicável) ou conjunto de alterações. Para obter mais informações, consulte [Configurar opções da pilha](#).

 Note

Os conjuntos de alterações para pilhas aninhadas são habilitados por padrão, o que criará conjuntos de alterações para todas as pilhas aninhadas especificadas em seu modelo. Para obter mais informações sobre os conjuntos de alterações para pilhas aninhadas, consulte [Conjuntos de alteração para pilhas aninhadas](#).

8. Se o modelo incluir recursos do IAM, escolha Reconheço que o CloudFormation pode criar recursos do IAM em Recursos. Os recursos do IAM podem modificar as permissões na sua conta da AWS. Analise esses recursos para garantir que você está permitindo apenas as ações que deseja. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
9. Escolha Próximo para continuar.
10. Na página Review **stack-name** (Revisar nome-pilha), revise as alterações para esse conjunto de alterações.
11. Selecione Enviar.

 Note

Os conjuntos de alterações no nível da propriedade do CloudFormation não resolvem referências entre pilhas quando você cria conjuntos de alterações para pilhas

aninhadas. Os conjuntos de alterações poderão marcar recursos em uma pilha subordinada para substituição condicional se eles referenciarem a saída de uma pilha superior e ela tiver sido modificada

Você será redirecionado para a guia Changes (Alterações) da página de detalhes do conjunto de alterações. Enquanto o CloudFormation está gerando o conjunto de alterações, o status do conjunto de alterações é CREATE_PENDING. Após criar o conjunto de alterações, o CloudFormation define o status como CREATE_COMPLETE. Na seção Alterações, o CloudFormation relaciona todas as alterações que serão feitas em sua pilha. Para obter mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#).

Se o CloudFormation falhar na criação do conjunto de alterações (exibindo o status FAILED), corrija o erro exibido no campo Status e recrie o conjunto de alterações.

12. Depois de confirmar que as alterações parecem estar corretas, escolha Executar conjunto de alterações

Para criar um conjunto de alterações (AWS CLI)

- Use o comando [create-change-set](#).

Você envia suas alterações como opções de comando. Você pode especificar novos valores de parâmetros, um modelo modificado, ou ambos. Por exemplo, o comando a seguir cria um conjunto de alterações chamado SampleChangeSet para a pilha MyStack. O conjunto de alterações usa o modelo da pilha atual, mas com um valor diferente para o parâmetro Purpose:

```
aws cloudformation create-change-set --stack-name MyStack \  
  --change-set-name SampleChangeSet --use-previous-template \  
  --parameters \  
    ParameterKey="InstanceType",UsePreviousValue=true  
    ParameterKey="KeyPairName",UsePreviousValue=true  
    ParameterKey="Purpose",ParameterValue="production"
```

Visualizar um conjunto de alterações para uma pilha do CloudFormation

Depois de criar um conjunto de alterações, você pode visualizar as alterações propostas antes da execução. Você pode usar o console do CloudFormation, a AWS CLI ou a API do CloudFormation para visualizar conjuntos de alterações. O console do CloudFormation fornece um resumo das alterações e uma lista detalhada delas no formato JSON. A AWS CLI e a API do AWS CloudFormation retornam uma lista detalhada das alterações em formato JSON.

View a change set (console)

Para visualizar um conjunto de alterações

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, escolha o nome da pilha que contém o conjunto de alterações que você deseja visualizar.
4. No painel de navegação, escolha Change Sets (Conjuntos de alterações) para visualizar uma lista dos conjuntos de alterações da pilha.
5. Escolha o nome do conjunto de alterações que deseja visualizar.

O console do CloudFormation direcionará você para a página de detalhes do conjunto de alterações, onde poderá ver o horário em que o conjunto de alterações foi criado, o status, a entrada usada para gerá-lo e um resumo das alterações.

Na seção Alterações, cada linha representa um recurso que o CloudFormation vai adicionar, modificar ou remover.

- Adicionar – O CloudFormation cria um recurso quando você adicionar um recurso ao modelo da pilha.
- Modificar – O CloudFormation modifica um recurso quando você altera as propriedades de um recurso no modelo da pilha.
- Remover – O CloudFormation exclui um recurso quando você exclui um recurso ao modelo da pilha.

 Note

Uma modificação pode fazer com que o recurso seja interrompido ou substituído (recriado). Para obter mais informações sobre os comportamentos de atualização de recursos, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

Para se concentrar em alterações específicas, use a visualização com filtro. Por exemplo, filtre por um determinado tipo de recurso, como `AWS::EC2::Instance`. Para filtrar por um recurso específico, especifique o ID lógico ou físico, como `myWebServer` ou `i-123abcd4`.

6. Na seção Alterações, escolha Visualizar detalhes na coluna Alterações no nível da propriedade para visualizar as alterações feitas nos valores das propriedades do recurso.
7. O console do CloudFormation direciona você para a página das alterações no nível de propriedade de um recurso, onde é possível ver a configuração do modelo do recurso antes da execução de um conjunto de alterações e como ficará a configuração do modelo depois que o conjunto de alterações for executado.

A tabela da seção Alterações no nível da propriedade mostra o Caminho, o Tipo de alteração, o Valor antes e o Valor depois das propriedades afetadas. Na tabela, escolha a caixa de seleção de cada alteração que você deseja realçar nas visualizações Antes e Depois do seu modelo para ver quais alterações serão feitas no nível da propriedade.

- Adicionar: as propriedades adicionadas são realçadas em verde.
- Modificar: as propriedades modificadas são realçadas em azul.
- Remover: as propriedades removidas são realçadas em vermelho.

View a change set for nested stack (console)

Para visualizar um conjunto de alterações para pilhas aninhadas (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, escolha o nome da pilha que contém o conjunto de alterações que você deseja visualizar.

4. No painel de navegação, escolha Conjuntos de alterações para visualizar uma lista dos conjuntos de alterações da pilha.
5. Escolha o nome do conjunto de alterações que deseja visualizar.

O console do CloudFormation direcionará você para a página de detalhes do conjunto de alterações, onde poderá ver o horário em que o conjunto de alterações foi criado, o status, a entrada usada para gerá-lo e um resumo das alterações.

Na seção Changes (Alterações), cada linha representa um recurso para o qual o CloudFormation vai adicionar, modificar, remover ou mostrar o status de dinâmico.

- Adicionar – O CloudFormation cria um recurso quando você adicionar um recurso ao modelo da pilha.
- Modificar – O CloudFormation modifica um recurso quando você altera as propriedades de um recurso no modelo da pilha.
- Remover – O CloudFormation exclui um recurso quando você exclui um recurso ao modelo da pilha.
- Dinâmico – O CloudFormation não pode determinar a ação exata de alteração de recurso com base no modelo da pilha aninhada.

 Note

Uma modificação pode fazer com que o recurso seja interrompido ou substituído (recriado). Para obter mais informações sobre os comportamentos de atualização de recursos, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

Para se concentrar em alterações específicas, use a visualização com filtro. Por exemplo, filtre por um determinado tipo de recurso, como **AWS::CloudFormation::Stack**. Para filtrar por um recurso específico, especifique o ID lógico ou físico, como **DeadLetterQueue** ou **NestedStack**.

6. Na seção Changes (Alterações), escolha View nested change set (Exibir conjunto de alterações aninhadas) do conjunto de alterações aninhadas que deseja visualizar.

O console do CloudFormation direciona você para a página de detalhes do conjunto de alterações aninhadas. Você pode escolher Go to root change set (Acessar o conjunto raiz de alterações) para visualizar o conjunto raiz de alterações ou escolher View parent change set (Exibir conjunto pai de alterações) para visualizar o conjunto pai de alterações. Para obter mais informações, consulte [Conjuntos de alteração para pilhas aninhadas](#).

 Note

Os conjuntos de alterações no nível da propriedade do CloudFormation não resolvem referências entre pilhas quando você cria conjuntos de alterações para pilhas aninhadas. Os conjuntos de alterações poderão marcar recursos em uma pilha subordinada para substituição condicional se eles referenciarem a saída de uma pilha superior e ela tiver sido modificada

Para visualizar um conjunto de alterações (AWS CLI)

1. Para obter o ID do conjunto de alterações, execute o comando [change-sets](#).

Especifique o nome da pilha que contém o conjunto de alterações que você deseja visualizar, como mostrado no seguinte exemplo:

```
aws cloudformation list-change-sets --stack-name MyStack
```

O CloudFormation retorna uma lista de conjuntos de alterações, semelhante a esta:

```
{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
      "Status": "CREATE_COMPLETE",
      "ChangeSetName": "SampleChangeSet",
      "CreationTime": "2020-11-18T20:44:05.889Z",
      "StackName": "MyStack",
      "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
    },
    {
```

```

    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-conditional",
    "CreationTime": "2020-11-18T21:15:56.398Z",
    "StackName": "MyStack",
    "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-conditional/1a2345b6-0000-00a0-
a123-00abc0abc000"
  },
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-replacement",
    "CreationTime": "2020-11-18T21:03:37.706Z",
    "StackName": "MyStack",
    "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-replacement/1a2345b6-0000-00a0-
a123-00abc0abc000"
  }
]
}

```

2. Execute o comando [describe-change-set](#), especificando o ID do conjunto de alterações que você deseja visualizar. Por exemplo:

```

aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000

```

O CloudFormation retorna informações sobre o conjunto de alterações especificado.

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    }
  ],

```

```

    {
      "ParameterValue": "ellioty-useast1",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Static",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          }
        ],
        "Action": "Modify",
        "Scope": [
          "Tags"
        ],
        "LogicalResourceId": "MyEC2Instance",
        "Replacement": "False"
      },
      "Type": "Resource"
    }
  ],
  "CreationTime": "2020-11-18T23:35:25.813Z",
  "Capabilities": [],
  "StackName": "MyStack",
  "NotificationARNs": [],
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet-direct/9edde307-960d-4e6e-ad66-b09ea2f20255"
}

```

Use `--include-property-values` com `describe-change-set` para listar as alterações no nível da propriedade.

A chave `Changes` relaciona as alterações nos recursos. Se você executar esse conjunto de alterações, o CloudFormation poderá atualizar as tags da instância do EC2 `i-1abc23d4`. Para obter uma descrição de cada campo, consulte o tipo de dados [Change](#), na Referência de APIs do AWS CloudFormation.

Para obter mais exemplos de conjuntos de alterações, consulte [Exemplos de conjuntos de alterações para pilhas do CloudFormation](#).

Para visualizar as alterações no nível da propriedade em um conjunto de alterações (AWS CLI)

- O comando a seguir lista as alterações no nível da propriedade relacionadas a um conjunto de alterações de um recurso `AWS::EC2::NetworkInterface` que removerá a propriedade `Ipv4Prefixes`, modificará a `Description` do recurso e adicionará uma `Tag`:

```
aws cloudformation describe-change-set --include-property-values \  
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841
```

O seguinte é um exemplo de saída.

```
"ChangeSetName": "ExampleChangeSet",  
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",  
  "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/  
ab664180-f686-11ee-9e29-12cd92393671",  
  "StackName": "ExampleStack",  
  "Description": null,  
  "Parameters": null,  
  "CreationTime": "2024-04-09T18:04:59.935000+00:00",  
  "ExecutionStatus": "AVAILABLE",  
  "Status": "CREATE_COMPLETE",  
  "StatusReason": null,  
  "NotificationARNs": [],  
  "RollbackConfiguration": {  
    "RollbackTriggers": []  
  },  
  "Capabilities": [],
```



```

    "Tags": null,
    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
  {
    "Changes": [
      {
        "Type": "Resource",
        "ResourceChange": {
          "Action": "Modify",
          "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
          "PhysicalResourceId": "eni-067fd35b649a05b71",
          "ResourceType": "AWS::EC2::NetworkInterface",
          "Replacement": "False",
          "Scope": [
            "Properties",
            "Tags"
          ],
          "Details": [
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Ipv4Prefixes",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Ipv4Prefixes",
                "BeforeValue": "[]",
                "AttributeChangeType": "Remove"
              },
              "Evaluation": "Static",
              "ChangeSource": "DirectModification"
            },
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Description",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Description",
                "BeforeValue": "",
                "AfterValue": "Description",
                "AttributeChangeType": "Modify"
              },
              "Evaluation": "Static",

```

```

        "ChangeSource": "DirectModification"
    },
    {
        "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never",
            "Path": "/Properties/Tags/0",
            "AfterValue": "{ \"Key\": \"Test\", \"Value\": \"Test\" }",
            "AttributeChangeType": "Add"
        },
        "Evaluation": "Static",
        "ChangeSource": "DirectModification"
    }
],
    "BeforeContext": "{ \"Properties\": { \"Description\": \"\",
    \"PrivateIpAddress\": \"172.31.76.2\", \"PrivateIpAddresses\": [ { \"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\" } ], \"SecondaryPrivateIpAddressCount\": \"0\",
    \"Ipv6PrefixCount\": \"0\", \"Ipv4Prefixes\": [], \"Ipv4PrefixCount\": \"0\", \"GroupSet
    \": [ \"sg-05a45689b1059e82d\" ], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": [] },
    \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\": \"Retain\" }",
    "AfterContext": "{ \"Properties\": { \"Description\": \"Description\",
    \"PrivateIpAddress\": \"172.31.76.2\", \"PrivateIpAddresses\": [ { \"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\" } ], \"SecondaryPrivateIpAddressCount
    \": \"0\", \"Ipv6PrefixCount\": \"0\", \"Ipv4PrefixCount\": \"0\", \"GroupSet\":
    [ \"sg-05a45689b1059e82d\" ], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": [ { \"Value\":
    \"Test\", \"Key\": \"Test\" } ] }, \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\":
    \"Retain\" }"
    }
    ],
    "ChangeSetName": "ExampleChangeSet",
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
    ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/
    ab664180-f686-11ee-9e29-12cd92393671",

```

Uso de conjuntos de alterações sensíveis a desvios

Os conjuntos de alterações sensíveis a desvios são conjuntos de alterações aprimorados do CloudFormation que permitem identificar e gerenciar o desvio de pilha de maneira segura. As

pilhas se desviam da configuração de modelo esperada quando os recursos da pilha são alterados fora do CloudFormation, usando o Console de gerenciamento da AWS, a AWS CLI ou o SDK dos serviços subjacentes. Os conjuntos de alterações sensíveis a desvios comparam os modelos com o estado real dos recursos da pilha e alinham os recursos desviados com suas definições de modelo. Quando a definição do modelo de um recurso é atualizada para corresponder ao seu estado real, os conjuntos de alterações sensíveis a desvios redefinem o status de desvio do recurso sem modificar o recurso.

Os benefícios de usar conjuntos de alterações sensíveis ao desvio incluem:

- Pré-visualizar substituições de desvio: detecte se uma implantação reverterá as alterações críticas fora de banda feitas durante a resposta a incidentes.
- Reconciliação sistemática de desvios: alinhe com segurança os recursos desviados com as definições de modelos compatíveis, seja substituindo as propriedades desviadas pelos valores do modelo ou atualizando o modelo para corresponder ao estado real do recurso.
- Recursos aprimorados de reversão: se uma implantação falhar, restaure os recursos para seu estado real de pré-implantação, não para o estado anterior do modelo.
- Visibilidade completa: veja exatamente como sua implantação afetará as configurações reais dos recursos antes de fazer alterações.

Os conjuntos de alterações tradicionais fornecem uma comparação do seu novo modelo com o modelo anterior para uma pilha, mas não levam em conta o desvio da pilha. Os conjuntos de alterações sensíveis a desvios resolvem esse problema fornecendo uma comparação tripla entre:

- Estado real: a configuração ativa de seus recursos. O CloudFormation lerá a configuração dos recursos em sua conta no momento da criação do conjunto de alterações por meio das APIs de serviço subjacentes.
- Estado de implantação anterior: a configuração definida no modelo da sua última implantação do CloudFormation.
- Estado desejado: a configuração definida em seu novo modelo.

Os conjuntos de alterações sensíveis ao desvio atualizarão o estado real de todos os recursos da pilha para corresponder ao estado desejado, mesmo que um recurso não tenha sido alterado explicitamente no modelo.

Considerações

- Suporte ao tipo de recurso: os conjuntos de alterações sensíveis a desvios são compatíveis com a comparação do estado desejado com o estado real de centenas de tipos de recursos. Para tipos de recursos não compatíveis, os conjuntos de alterações sensíveis a desvios recorrem à comparação do estado de implantação anterior com o estado desejado. Para obter detalhes, consulte a seção Tipos de recursos compatíveis.
- Propriedades somente para gravação: para propriedades que contêm dados sensíveis (senhas, segredos), os conjuntos de alterações sensíveis a desvios se comparam aos valores de implantação anteriores, em vez dos valores reais.
- Propriedades gerenciadas pela AWS: os conjuntos de alterações sensíveis a desvios preservam as alterações fora de banda feitas pelos serviços da AWS nas propriedades gerenciadas, como a capacidade desejada de um grupo do Auto Scaling. Para obter detalhes, consulte a seção de propriedades gerenciadas pela AWS.
- Chaves de tags externas: conjuntos de alterações sensíveis a desvios não removem nem modificam chaves de tags que não foram especificadas no modelo, evitando conflitos com sistemas de controle de acesso por atributo (ABAC).
- Substituição de recursos desviados: conjuntos de alterações sensíveis a desvios não são compatíveis com a reconciliação de desvios para propriedades imutáveis.
- Anexos de pilha cruzada: alguns tipos de recursos, como `AWS::IAM::Policy`, podem influenciar o estado real de outros tipos de recursos, como `AWS::IAM::Role`. Os conjuntos de alterações sensíveis a desvios lidam com a anexação de recursos em uma pilha. Se um recurso for modificado por meio da anexação de um recurso secundário de uma pilha diferente, os conjuntos de alterações sensíveis a desvios detectarão a modificação como desvio e poderão reverter o anexo. Os recursos anexáveis conhecidos incluem `AWS::IAM::Policy`, `AWS::IAM::ManagedPolicy`, `AWS::EC2::SecurityGroupIngress` e `AWS::EC2::SecurityGroupEgress`.

Propriedades gerenciadas pela AWS

Você pode configurar propriedades específicas de recursos para o gerenciamento ativo da AWS. Por exemplo, é possível permitir que o Amazon Relational Database Service (Amazon RDS) atualize automaticamente a versão secundária do mecanismo de uma tabela do Amazon RDS. Essas alterações podem aparecer como um desvio de pilha no CloudFormation. Os conjuntos de alterações sensíveis a desvios reconhecem que o desvio é esperado para propriedades gerenciadas pela AWS

e deixam seu valor real inalterado se você não tiver modificado a propriedade em seu modelo. Os principais exemplos de propriedades gerenciadas pela AWS são:

- Habilitar a propriedade `AutoMinorVersionUpgrade` de uma tabela do Amazon RDS para permitir atualizações automáticas da versão do mecanismo.
- Uso do recurso `AWS::ApplicationAutoScaling::ScalableTarget` para habilitar o ajuste de escala automático de propriedades como as unidades de capacidade de leitura/gravação de uma tabela do Amazon DynamoDB e a contagem desejada de um cluster do Amazon Elastic Container Service.
- Uso da `AWS::AutoScaling::ScalingPolicy` para grupos do Auto Scaling.

Os conjuntos de alterações com sensíveis a desvios esclarecem as propriedades que foram identificadas como gerenciadas pela AWS. Para obter detalhes, consulte a seção da AWS CLI.

Uso de conjuntos de alterações sensíveis a desvios (console)

Você pode criar e gerenciar conjuntos de alterações sensíveis a desvios por meio do console do CloudFormation usando o mesmo fluxo de trabalho dos conjuntos de alterações tradicionais, com opções adicionais para modos de implantação.

Criação de conjuntos de alterações sensíveis a desvios

Para criar um conjunto de alterações sensíveis a desvios (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, selecione a pilha que você deseja atualizar.
3. Escolha Ações de pilhas e depois Criar conjunto de alterações para a pilha atual.
4. Na página Criar conjunto de alterações, em Tipo de conjunto de alterações, escolha Conjunto de alterações sensível a desvios.
5. Escolha Criar conjunto de alterações.
6. Na página de detalhes do conjunto de alterações, revise a comparação tripla que mostra a implantação atual, anterior e o estado desejado para cada recurso afetado.
7. Se as alterações estiverem adequadas para você, escolha Executar conjunto de alterações.

Análise de conjuntos de alterações sensíveis a desvios

Ao visualizar um conjunto de alterações sensível a desvios no console, você verá informações aprimoradas em comparação com os conjuntos de alterações tradicionais:

- Status de desvio da pilha: indica se a pilha foi desviada de sua última implantação.
- Comparação de propriedades: mostra uma diferença JSON entre o estado real e o estado desejado de um recurso afetado.
- Indicadores de desvio: marca claramente as propriedades do diferencial JSON que foram desviadas. Clique em Visualizar desvio para ver o valor da implantação anterior de uma propriedade.
- Indicadores da fonte de valor: mostra se os valores anteriores de uma propriedade vêm do estado real ou do estado de implantação anterior.

Uso de conjuntos de alterações sensíveis a desvios (AWS CLI)

Você pode criar e gerenciar conjuntos de alterações sensíveis a desvios usando a AWS CLI, adicionando o parâmetro `--deployment-mode REVERT_DRIFT` ao comando `create-change-set`.

Criação de conjuntos de alterações sensíveis a desvios

Para criar um conjunto de alterações sensível a desvios

Use o comando [create-change-set](#) com o parâmetro `--deployment-mode REVERT_DRIFT` para criar um conjunto de alterações sensível ao desvio.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name my-drift-aware-changeset \  
  --template-body file://updated-template.yaml \  
  --deployment-mode REVERT_DRIFT \  
  --capabilities CAPABILITY_IAM
```

Análise de conjuntos de alterações sensíveis a desvios

Para revisar os detalhes de um conjunto de alterações sensível a desvios

Use o comando [describe-change-set](#) para revisar os detalhes de um conjunto de alterações sensível a desvios:

```
aws cloudformation describe-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

A resposta inclui informações aprimoradas para conjuntos de alterações sensíveis a desvios:

- **StackDriftStatus**: mostra se a pilha foi desviada (DRIFTED, IN_SYNC, NOT_CHECKED ou UNKNOWN).
- **ResourceDriftStatus**: mostra o status de desvio de cada recurso (DELETED, MODIFIED, IN_SYNC ou NOT_CHECKED).
- **BeforeValueFrom**: indica se o valor anterior de uma propriedade vem de ACTUAL_STATE ou PREVIOUS_DEPLOYMENT_STATE.
- **Drift**: contém os detalhes do desvio de uma propriedade, incluindo PreviousValue, ActualValue e DriftDetectionTimestamp.
- **ResourceDriftIgnoredProperties**: contém as propriedades de um recurso para o qual o conjunto de alterações não reverterá o desvio e os motivos para ignorar o desvio.

Para obter mais informações, consulte [DescribeChangeSet](#) na Referência de APIs do AWS CloudFormation.

Execução de conjuntos de alterações sensíveis a desvios

Para executar um conjunto de alterações sensível a desvios

Depois de analisar o conjunto de alterações, use o comando [execute-change-set](#) para aplicar as alterações:

```
aws cloudformation execute-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

Se a implantação falhar, o CloudFormation reverterá os recursos para seu estado real de pré-implantação e preservará as alterações fora de banda feitas desde a última implantação.

Tipos de recursos compatíveis

Os conjuntos de alterações sensíveis são compatíveis com uma comparação tripla do estado real, do estado anterior da implantação e do estado desejado para todos os [tipos de recursos](#) compatíveis com a detecção de desvios, exceto os seguintes tipos de recursos:

- `AWS::SageMaker::MlflowTrackingServer`
- `AWS::Route53Resolver::FirewallRuleGroup`
- `AWS::MediaLive::Multiplexprogram`
- `AWS::VpcLattice::ResourceConfiguration`
- `AWS::S3::StorageLensGroup`
- `AWS::Bedrock::AgentAlias`
- `AWS::MSK::Cluster`
- `AWS::RDS::DBProxy`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::QBusiness::Index`
- `AWS::NetworkManager::CoreNetwork`
- `AWS::IAM::OIDCProvider`
- `AWS::Organizations::ResourcePolicy`
- `AWS::SNS::TopicInlinePolicy`
- `AWS::Route53::KeySigningKey`
- `AWS::DataZone::PolicyGrant`
- `AWS::Transfer::Certificate`
- `AWS::SageMaker::ImageVersion`
- `AWS::Neptune::DBParameterGroup`
- `AWS::ODB::CloudVmCluster`
- `AWS::RolesAnywhere::TrustAnchor`
- `AWS::Detective::Graph`
- `AWS::Maester::DocumentType`
- `AWS::SageMaker::ModelPackageGroup`

- `AWS::S3Express::BucketPolicy`
- `AWS::Panorama::PackageVersion`
- `AWS::S3Tables::TableBucketPolicy`

Os conjuntos de alterações sensíveis a desvios recorrem a uma comparação do estado de implantação anterior e do estado desejado para recursos não compatíveis com a comparação tripla.

Executar um conjunto de alterações para uma pilha do CloudFormation

Para fazer as modificações descritas em um conjunto de alterações em sua pilha, execute o conjunto de alterações.

Important

Depois de executar um conjunto de alterações, o CloudFormation exclui qualquer conjunto de alterações que esteja associado à pilha, já que não são mais válidos para a pilha atualizada. Se uma atualização falhar, você precisará criar um novo conjunto de alterações.

Políticas de pilha e como executar um conjunto de alterações

Se você executar um conjunto de alterações em uma pilha que tenha uma política de pilha associada a ela, o CloudFormation impõe a política ao atualizá-la. Não é possível especificar uma política de pilha temporária que substitua a existente ao executar um conjunto de alterações. Para atualizar um recurso protegido, é necessário atualizar a política de pilha ou usar o método de atualização direta. Para obter mais informações, consulte [Atualizar pilhas diretamente](#).

Execute a change set (console)

Para executar um conjunto de alterações

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, escolha o nome da pilha em execução que você deseja atualizar.
4. No painel de navegação, escolha Conjuntos de alterações para visualizar uma lista dos conjuntos de alterações da pilha.
5. Escolha o nome do conjunto alterações que deseja executar.

6. Na página de detalhes do conjunto de alterações, escolha Executar conjunto de alterações.

O CloudFormation inicia a atualização da pilha imediatamente. O console do CloudFormation direciona você para a guia Eventos, em que é possível monitorar o progresso da atualização da pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Execute a change set for nested stacks (console)

Para executar um conjunto de alterações para pilhas aninhadas

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, escolha o nome da pilha em execução que você deseja atualizar. Você deve escolher o nome da pilha associado ao conjunto raiz de alterações.
4. No painel de navegação, escolha Conjuntos de alterações para visualizar uma lista dos conjuntos de alterações da pilha.
5. Escolha o nome do conjunto raiz de alterações que deseja executar.
6. Na página de detalhes do conjunto de alterações, escolha Executar conjunto de alterações.

Note

O CloudFormation executa as alterações descritas no seu conjunto raiz de alterações e conjuntos de alterações aninhadas, se a opção Habilitado para conjuntos de alterações para pilhas aninhadas tiver sido selecionada durante o processo [Criar um conjunto de alterações para uma pilha do CloudFormation](#).

O CloudFormation inicia a atualização da pilha imediatamente. O console do CloudFormation direciona você para a guia Eventos, em que é possível monitorar o progresso da atualização da pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Para executar um conjunto de alterações (AWS CLI)

- Execute o comando [execute-change-set](#).

Especifique o ID do conjunto de alterações que deseja executar, como mostrado no exemplo a seguir:

```
aws cloudformation execute-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

O comando no exemplo executa um conjunto de alterações com o ID `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000`.

Depois de executar o comando, o CloudFormation inicia a atualização da pilha. Para visualizar o progresso da pilha, use o comando [describe-stacks](#).

Excluir um conjunto de alterações para uma pilha do CloudFormation

A exclusão de um conjunto de alterações o remove da lista de conjuntos de alterações da pilha. A exclusão de um conjunto de alterações impede que você ou outro usuário execute acidentalmente um conjunto de alterações que não deva ser aplicado. A menos que você os exclua, o CloudFormation reterá todos os conjuntos de alterações até você atualizar a pilha.

Delete a change set

Para excluir um conjunto de alterações (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, selecione o nome da pilha que contém o conjunto de alterações que deseja excluir.
4. No painel de navegação, escolha Conjuntos de alterações para visualizar uma lista dos conjuntos de alterações da pilha.
5. Selecione o nome do conjunto alterações que deseja excluir.
6. Na página de detalhes do conjunto de alterações, escolha Excluir conjunto de alterações.

O CloudFormation começa a excluir o conjunto de alterações imediatamente da lista de conjuntos de alterações da pilha e você é redirecionado para a página Pilhas.

Delete a change set for nested stacks (console)

Para excluir um conjunto de alterações para pilhas aninhadas

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, selecione o nome da pilha associada ao conjunto de alterações raiz.
4. No painel de navegação, escolha Conjuntos de alterações para visualizar uma lista dos conjuntos de alterações da pilha.
5. Selecione o nome do conjunto alterações que deseja excluir.
6. Na página de detalhes do conjunto de alterações, escolha Delete (Excluir). Escolhendo Excluir, você excluirá toda a hierarquia de conjuntos de alterações aninhados.

Note

A operação de exclusão para conjuntos de alterações para pilhas aninhadas é assíncrona e mostrará um status DELETE_PENDING, seguido de um status DELETE_IN_PROGRESS. Após a conclusão da operação de exclusão do conjunto de alterações, os conjuntos de alterações serão removidos da lista. As pilhas aninhadas com status REVIEW_IN_PROGRESS também serão excluídas se tiverem sido criadas durante a criação do conjunto de alterações.

O CloudFormation começa imediatamente a excluir o conjunto de alterações da lista de conjuntos de alterações da pilha.

Note

Se você tiver pilhas aninhadas que estão presas em uma operação em andamento, consulte Solução de problemas de erros [Falha na reversão de pilhas aninhadas](#).

Para excluir um conjunto de alterações (AWS CLI)

- Execute o comando [delete-change-set](#), especificando o ID do conjunto de alterações que você deseja excluir, conforme mostrado no seguinte exemplo:

```
aws cloudformation delete-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Exemplos de conjuntos de alterações para pilhas do CloudFormation

Esta seção fornece exemplos de conjuntos de alterações que o CloudFormation poderia criar para alterações de pilha comuns. Eles mostram como editar um modelo diretamente; modificar um único parâmetro de entrada; planejar a recriação de recursos (substituições) (o que evita a perda de dados que não foram colocados em backup ou a interrupção de aplicativos que estão em execução em sua pilha) e adicionar e remover recursos. Para ilustrar como os conjuntos de alterações funcionam, vamos acompanhar as alterações que foram enviadas e discutir o conjunto de alterações resultante. Como cada exemplo tem como fundamento o pressuposto de que você entendeu o exemplo anterior, recomendamos que você leia-os em sequência. Para obter uma descrição de cada campo em um conjunto de alterações, consulte o tipo de dados [Change](#), na Referência de APIs do AWS CloudFormation.

Você pode usar o [console](#), a AWS CLI ou a operação da API [DescribeChangeSet](#) do CloudFormation para visualizar detalhes de conjuntos de alterações.

Geramos cada um dos seguintes conjuntos de alterações de uma pilha com o seguinte [modelo de exemplo](#):

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Description" : "A sample EC2 instance template for testing change sets.",  
  "Parameters" : {  
    "Purpose" : {  
      "Type" : "String",  
      "Default" : "testing",  
      "AllowedValues" : ["testing", "production"],  
      "Description" : "The purpose of this instance."  
    },  
    "KeyPairName" : {  
      "Type": "AWS::EC2::KeyPair::KeyName",  
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the  
instance"  
    },  
  },  
}
```

```

    "InstanceType" : {
      "Type" : "String",
      "Default" : "t2.micro",
      "AllowedValues" : ["t2.micro", "t2.small", "t2.medium"],
      "Description" : "The EC2 instance type."
    }
  },
  "Resources" : {
    "MyEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyPairName" },
        "InstanceType" : { "Ref" : "InstanceType" },
        "ImageId" : "ami-8fcee4e5",
        "Tags" : [
          {
            "Key" : "Purpose",
            "Value" : { "Ref" : "Purpose" }
          }
        ]
      }
    }
  }
}

```

Editar um modelo diretamente

Quando você modifica diretamente os recursos no modelo de pilha para gerar um conjunto de alterações, o CloudFormation classifica a alteração como uma modificação direta, em oposição a alterações iniciadas por um valor de parâmetro atualizado. O conjunto de alterações a seguir, que adicionou uma nova tag à instância `i-1abc23d4`, é um exemplo de uma modificação direta. Todos os outros valores de entrada, como os valores de parâmetro e as capacidades, permanecem inalterados, assim podemos nos concentrar na estrutura de `Changes`.

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {
      "ParameterValue": "testing",

```

```

        "ParameterKey": "Purpose"
    },
    {
        "ParameterValue": "MyKeyName",
        "ParameterKey": "KeyPairName"
    },
    {
        "ParameterValue": "t2.micro",
        "ParameterKey": "InstanceType"
    }
],
"Changes": [
    {
        "ResourceChange": {
            "ResourceType": "AWS::EC2::Instance",
            "PhysicalResourceId": "i-1abc23d4",
            "Details": [
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Static",
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never"
                    }
                }
            ],
            "Action": "Modify",
            "Scope": [
                "Tags"
            ],
            "LogicalResourceId": "MyEC2Instance",
            "Replacement": "False"
        },
        "Type": "Resource"
    }
],
"CreationTime": "2020-11-18T23:35:25.813Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Na estrutura de `Changes`, há apenas uma estrutura `ResourceChange`. Essa estrutura descreve informações como o tipo de recurso que o CloudFormation alterará, a ação que o CloudFormation realizará, o ID do recurso, o escopo da alteração e se ela requer uma substituição (em que o CloudFormation criará um novo recurso e, em seguida, excluirá o antigo). No exemplo, o conjunto de alterações indica que o CloudFormation modificará o atributo `Tags` da instância do EC2 `i-1abc23d4` e não exigirá que a instância seja substituída.

Na estrutura `Details`, o CloudFormation rotula essa mudança como uma modificação direta que nunca exigirá que a instância seja recriada (substituída). Você pode executar essa mudança com confiança, sabendo que o CloudFormation não substituirá a instância.

O CloudFormation mostra essa mudança como uma avaliação `Static`. Uma avaliação estática significa que o CloudFormation pode determinar o valor da tag antes de executar o conjunto de alterações. Em alguns casos, o CloudFormation pode determinar um valor somente depois que você executar um conjunto de alterações. O CloudFormation rotula essas alterações como avaliações `Dynamic`. Por exemplo, se você fizer referência a um recurso atualizado que é substituído condicionalmente, o CloudFormation não poderá determinar se ela será alterada.

Modificar o valor de um parâmetro de entrada

Ao modificar o valor de um parâmetro de entrada, o CloudFormation gera duas alterações para cada recurso que usa o valor de parâmetro atualizado. Neste exemplo, queremos destacar como são essas alterações e em quais informações você deve se concentrar. O exemplo a seguir foi gerado alterando-se apenas o valor do parâmetro de entrada `Purpose`.

O parâmetro `Purpose` especifica um valor de chave de tag para a instância EC2. No exemplo, o valor do parâmetro foi alterado de `testing` para `production`. O novo valor é mostrado na estrutura `Parameters`.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet",
  "Parameters": [
    {
      "ParameterValue": "production",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
```



```

        "ParameterKey": "KeyPairName"
    },
    {
        "ParameterValue": "t2.micro",
        "ParameterKey": "InstanceType"
    }
],
"Changes": [
    {
        "ResourceChange": {
            "ResourceType": "AWS::EC2::Instance",
            "PhysicalResourceId": "i-1abc23d4",
            "Details": [
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never"
                    }
                },
                {
                    "CausingEntity": "Purpose",
                    "ChangeSource": "ParameterReference",
                    "Evaluation": "Static",
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never"
                    }
                }
            ],
            "Action": "Modify",
            "Scope": [
                "Tags"
            ],
            "LogicalResourceId": "MyEC2Instance",
            "Replacement": "False"
        },
        "Type": "Resource"
    }
],
"CreationTime": "2020-11-18T23:59:18.447Z",
"Capabilities": [],
"StackName": "MyStack",

```

```
"NotificationARNs": [],  
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"  
}
```

A estrutura `Changes` funciona de maneira semelhante ao exemplo em [Editar um modelo diretamente](#). Existe apenas uma estrutura `ResourceChange`. Ela descreve uma alteração no atributo `Tags` da instância EC2 `i-1abc23d4`.

No entanto, na estrutura `Details`, o conjunto de alterações mostra duas alterações para o atributo `Tags`, mesmo que apenas um único valor do parâmetro tenha sido alterado. Os recursos que fazem referência a um valor de parâmetro alterado (usando a função intrínseca `Ref`) sempre resultam em duas alterações: uma com uma avaliação `Dynamic` e outra com uma avaliação `Static`. Você pode ver esses tipos de alterações visualizando os seguintes campos:

- Para alterar a avaliação `Static`, visualize o campo `ChangeSource`. Neste exemplo, o campo `ChangeSource` será igual a `ParameterReference`, o que significa que essa alteração é resultado de um valor de referência de parâmetro atualizado. O conjunto de alterações deve conter uma alteração similar na avaliação `Dynamic`.
- Você pode encontrar a alteração da avaliação `Dynamic` correspondente comparando a estrutura `Target` de ambas as alterações, que conterão as mesmas informações. Neste exemplo, as estruturas `Target` das duas alterações contêm os mesmos valores para os campos `Attribute` e `RequireRecreation`.

Para esses tipos de alterações, o foco na avaliação estático, que oferece mais informações detalhadas sobre a alteração. Neste exemplo, a avaliação estática mostra que a alteração é o resultado de uma modificação no valor de referência de um parâmetro (`ParameterReference`). O parâmetro exato que foi alterado é indicado pelo campo `CauseEntity` (o parâmetro `Purpose`).

Determinar o valor do campo `Replacement`

O campo `Replacement` em uma estrutura `ResourceChange` indica se o CloudFormation recriará o recurso. O planejamento da recriação de recursos (substituições) impede a perda de dados que não foram colocados em backup ou a interrupção de aplicativos que estão em execução em sua pilha.

O valor no campo `Replacement` depende de uma alteração exigir uma substituição ou não, o que é indicado pelo campo `RequiresRecreation` em uma estrutura `Target` da alteração. Por exemplo, se o campo `RequiresRecreation` for `Never`, o campo `Replacement` será `False`. No

entanto, se houver várias alterações em um único recurso e cada uma tiver um valor diferente para o campo `RequiresRecreation`, o CloudFormation atualizará o recurso usando o comportamento mais intrusivo. Em outras palavras, se apenas uma das muitas alterações exigir uma substituição, o CloudFormation terá que substituir o recurso e, portanto, definir o campo `Replacement` como `True`.

O conjunto de alterações a seguir foi gerado pela alteração dos valores de cada parâmetro (`Purpose`, `InstanceType`, e `KeyPairName`), que são usados pela instância EC2. Com essas alterações, o CloudFormation terá que substituir a instância porque o campo `Replacement` é igual a `True`.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-multiple",
  "Parameters": [
    {
      "ParameterValue": "production",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyNewKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.small",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-7bef86f8",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Dynamic",
            "Target": {
              "Attribute": "Properties",
              "Name": "KeyName",
              "RequiresRecreation": "Always"
            }
          }
        ]
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeSource": "DirectModification",
    "Evaluation": "Dynamic",
    "Target": {
      "Attribute": "Properties",
      "Name": "InstanceType",
      "RequiresRecreation": "Conditionally"
    }
  },
  {
    "ChangeSource": "DirectModification",
    "Evaluation": "Dynamic",
    "Target": {
      "Attribute": "Tags",
      "RequiresRecreation": "Never"
    }
  },
  {
    "CausingEntity": "KeyPairName",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {
      "Attribute": "Properties",
      "Name": "KeyName",
      "RequiresRecreation": "Always"
    }
  },
  {
    "CausingEntity": "InstanceType",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {
      "Attribute": "Properties",
      "Name": "InstanceType",
      "RequiresRecreation": "Conditionally"
    }
  },
  {
    "CausingEntity": "Purpose",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {

```

```

        "Attribute": "Tags",
        "RequiresRecreation": "Never"
      }
    ],
    "Action": "Modify",
    "Scope": [
      "Tags",
      "Properties"
    ],
    "LogicalResourceId": "MyEC2Instance",
    "Replacement": "True"
  },
  "Type": "Resource"
}
],
"CreationTime": "2020-11-18T00:39:35.974Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet-multiple/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Identifique a alteração que requer a substituição do recurso, visualizando cada alteração (as avaliações estáticas na estrutura `Details`). Neste exemplo, cada alteração tem um valor diferente para o campo `RequireRecreation`, mas a alteração na propriedade `KeyName` tem o comportamento de atualização mais intrusivo, sempre exigindo uma recriação. O CloudFormation substituirá a instância porque o nome da chave foi alterado.

Se o nome da chave permanecesse inalterado, a alteração na propriedade `InstanceType` teria o comportamento de atualização mais intrusivo (`Conditionally`), assim o campo `Replacement` seria `Conditionally`. Para encontrar as condições em que o CloudFormation substitui a instância, veja o comportamento de atualização da propriedade `InstanceType` do tipo de recurso [AWS::EC2::Instance](#).

Adicionar e remover recursos

O exemplo a seguir foi gerado enviando um modelo modificado que remove a instância EC2 e adiciona um grupo Auto Scaling e uma configuração de execução.

```
{
```

```
"StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
"Status": "CREATE_COMPLETE",
"ChangeSetName": "SampleChangeSet-addremove",
"Parameters": [
  {
    "ParameterValue": "testing",
    "ParameterKey": "Purpose"
  },
  {
    "ParameterValue": "MyKeyName",
    "ParameterKey": "KeyPairName"
  },
  {
    "ParameterValue": "t2.micro",
    "ParameterKey": "InstanceType"
  }
],
"Changes": [
  {
    "ResourceChange": {
      "Action": "Add",
      "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
      "Scope": [],
      "Details": [],
      "LogicalResourceId": "AutoScalingGroup"
    },
    "Type": "Resource"
  },
  {
    "ResourceChange": {
      "Action": "Add",
      "ResourceType": "AWS::AutoScaling::LaunchConfiguration",
      "Scope": [],
      "Details": [],
      "LogicalResourceId": "LaunchConfig"
    },
    "Type": "Resource"
  },
  {
    "ResourceChange": {
      "ResourceType": "AWS::EC2::Instance",
      "PhysicalResourceId": "i-1abc23d4",
      "Details": [],
```

```

        "Action": "Remove",
        "Scope": [],
        "LogicalResourceId": "MyEC2Instance"
    },
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T01:44:08.444Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-addremove/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Na estrutura `Changes`, há três estruturas `ResourceChange`, uma para cada recurso. Para cada recurso, o campo `Action` indica se o CloudFormation adiciona ou remove o recurso. Os campos `Scope` e `Details` estão vazios porque se aplicam somente a recursos modificados.

Para novos recursos, o CloudFormation não pode determinar o valor de alguns campos até que você execute o conjunto de alterações. Por exemplo, o CloudFormation não fornece os IDs físicos do grupo do Auto Scaling e configuração de execução, pois eles não existem ainda. O CloudFormation cria os novos recursos quando você executa o conjunto de alterações.

Visualizar alterações no nível da propriedade

O exemplo a seguir mostra as alterações no nível da propriedade feitas na propriedade `Tag` de uma instância do Amazon EC2. A tag `Value` e `Key` mudará para `Test`.

```

"ChangeSetName": "SampleChangeSet",
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/38d91d27-798d-4736-9bf1-fb7c46207807",
"StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
SampleEc2Template/68edcdc0-f6b6-11ee-966c-126d572cdd11",
"StackName": "SampleEc2Template",
"Description": "A sample EC2 instance template for testing change sets.",
"Parameters": [
    {
        "ParameterKey": "KeyPairName",
        "ParameterValue": "BatchTest"
    },
    {

```

```

        "ParameterKey": "Purpose",
        "ParameterValue": "testing"
    },
    {
        "ParameterKey": "InstanceType",
        "ParameterValue": "t2.micro"
    }
],
"CreationTime": "2024-04-09T21:29:10.759000+00:00",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [],
"RollbackConfiguration": {
:....skipping...
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "MyEC2Instance",
                "PhysicalResourceId": "i-0cc7856a36315e62b",
                "ResourceType": "AWS::EC2::Instance",
                "Replacement": "False",
                "Scope": [
                    "Tags"
                ],
            },
            "Details": [
                {
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never",
                        "Path": "/Properties/Tags/0/Value",
                        "BeforeValue": "testing",
                        "AfterValue": "Test",
                        "AttributeChangeType": "Modify"
                    },
                    "Evaluation": "Static",
                    "ChangeSource": "DirectModification"
                },
                {
                    "Target": {
                        "Attribute": "Tags",

```



```

        "RequiresRecreation": "Never",
        "Path": "/Properties/Tags/0/Key",
        "BeforeValue": "Purpose",
        "AfterValue": "Test",
        "AttributeChangeType": "Modify"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
  }
],
  "BeforeContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":"
  "\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\": [{\"Value\":\"testing\",
  \"Key\":\"Purpose\"}]}}",
  "AfterContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":"
  "\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\": [{\"Value\":\"Test\", \"Key\":"
  "\":\"Test\"}]}}"
}
}
]

```

A estrutura `Details` mostra os valores de `Key` e `Value` antes da execução do conjunto de alterações e quais serão os valores após a execução do conjunto de alterações.

Conjuntos de alteração para pilhas aninhadas

Com os conjuntos de alterações para pilhas aninhadas, você pode visualizar as alterações nos recursos de suas aplicações e infraestrutura em toda a hierarquia de pilha aninhada, dando continuidade às atualizações após confirmar que todas as alterações estejam conforme pretendido.

Consulte as seguintes seções para obter mais detalhes sobre conjuntos de alterações para pilhas aninhadas:

Tópicos

- [Visão geral de conjuntos de alterações e pilhas aninhadas](#)
- [Como trabalhar com conjuntos de alterações para pilhas aninhadas \(console\)](#)
- [Como trabalhar com conjuntos de alterações para pilhas aninhadas \(AWS CLI\)](#)

Visão geral de conjuntos de alterações e pilhas aninhadas

Os conjuntos de alterações para pilhas aninhadas combinam os seguintes recursos para expandir o escopo da visualização de alterações para toda a hierarquia de pilha:

- Um conjunto de alterações é um recurso do CloudFormation que oferece uma visualização de como as alterações propostas em uma pilha afetarão os recursos existentes ou recém-criados. Ao criar um conjunto de alterações, o CloudFormation fornece uma lista de alterações propostas, fazendo uma comparação entre sua pilha e as alterações nos recursos que você enviou. Para obter mais informações sobre conjuntos de alterações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).
- Uma pilha aninhada é uma pilha criada como parte de outra pilha. Por exemplo, talvez você tenha recursos relacionados a redes e segurança em uma pilha aninhada e recursos de aplicação em outra. Fazer o particionamento de modelos de aplicação dessa forma ajuda na capacidade de manutenção e reutilização de código. Para saber mais sobre pilhas aninhadas, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).

Como trabalhar com conjuntos de alterações para pilhas aninhadas (console)

- Criar um conjunto de alterações – Cria um conjunto de alterações mediante o envio de alterações com base em qualquer nível da hierarquia de pilha. Você pode enviar um modelo modificado de pilha ou valores modificados de parâmetro de entrada e o CloudFormation compara sua pilha aninhada às alterações que você enviou para gerar um conjunto de alterações. Os conjuntos de alterações para pilhas aninhadas são habilitados por padrão no console do CloudFormation. Para obter mais informações, consulte [Criar um conjunto de alterações para uma pilha do CloudFormation](#).

▼ Change sets

Change sets for nested stacks

Creates change sets for all nested stacks specified in your template. [Learn more](#) 

☒ Enabled

Creates a change set for all nested stacks specified in the template.

☐ Disabled

Create a change set only on the target stack.

Note

Um conjunto raiz de alterações é o conjunto de alterações associado à pilha com base na qual toda a hierarquia de conjuntos de alterações é criada. Você deve executar ou excluir

conjuntos de alterações para pilhas aninhadas com base no conjunto raiz de alterações. Para obter mais informações, consulte [Executar operações de pilha em pilhas aninhadas](#).

- Visualizar o conjunto de alterações: visualize as alterações nos recursos dentro de pilhas aninhadas antes de executá-las. Você pode ver as alterações propostas na seção Alterações do seu conjunto de alterações navegando pela pilha atual e pelos respectivos conjuntos de alterações aninhados. Para obter mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#).
- Executar o conjunto de alterações – Execute as alterações descritas no conjunto de alterações que pertencem à pilha atual e aos seus descendentes. A operação de execução deve ser feita a partir do conjunto raiz de alterações. Para obter mais informações, consulte [Executar um conjunto de alterações para uma pilha do CloudFormation](#).
- Excluir o conjunto de alterações – Remove os conjuntos de alterações da pilha atual. A exclusão de um conjunto de alterações ajuda a impedir que você ou outro usuário inicie acidentalmente um conjunto de alterações que não deva ser aplicado. A operação de exclusão deve ser executada com base no conjunto raiz de alterações. Para obter mais informações, consulte [Excluir um conjunto de alterações para uma pilha do CloudFormation](#).

Como trabalhar com conjuntos de alterações para pilhas aninhadas (AWS CLI)

create-change-set

- [create-change-set](#) – conjuntos de alterações para pilhas aninhadas não são habilitados por padrão para o AWS CLI. Para criar um conjunto de alterações para toda a hierarquia de pilha, especifique a opção `--include-nested-stacks`. Para obter mais informações, consulte [Criar um conjunto de alterações para uma pilha do CloudFormation](#).

O exemplo de AWS CLI a seguir cria um conjunto de alterações para a pilha raiz especificada.

```
aws cloudformation create-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM \  
  --include-nested-stacks
```

O seguinte é um exemplo de saída.

```
{
  "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:Stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

describe-change-set

- [describe-change-set](#) – Retorna uma lista de alterações que o CloudFormation fará se você executar o conjunto de alterações. Se o conjunto de alterações especificado contiver conjuntos filhos de alterações que pertençam a pilhas aninhadas, ChangeSetId retornará informações sobre esse conjunto de alterações. Para obter mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation](#).

O exemplo de AWS CLI a seguir descreve o conjunto de alterações para a pilha raiz especificada.

```
aws cloudformation describe-change-set \
  --change-set-name my-root-stack-change-set \
  --stack-name my-root-stack
```

O seguinte é um exemplo de saída.

```
{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "ChildStack",
        "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
        "ResourceType": "AWS::CloudFormation::Stack",
        "Replacement": "False",
        "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
        "Scope": [
          "Properties"
        ],
        "Details": [
```

```

        {
            "Target": {
                "Attribute": "Properties",
                "RequiresRecreation": "Never"
            },
            "Evaluation": "Dynamic",
            "ChangeSource": "Automatic"
        }
    ]
}

],
"ChangeSetName": "my-root-stack-change-set",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
"StackName": "my-root-stack",
"IncludeNestedStacks": true,
"ParentChangeSetId": null,
"RootChangeSetId": null,
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
    "CAPABILITY_IAM"
],
"Tags": null
}

```

O exemplo de AWS CLI a seguir descreve o conjunto de alterações para a pilha aninhada especificada.

```
aws cloudformation describe-change-set \
```

```
--change-set-name my-nested-stack-change-set \
--stack-name my-nested-stack
```

O seguinte é um exemplo de saída.

```
{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "function",
        "PhysicalResourceId": "my-function",
        "ResourceType": "AWS::Lambda::Function",
        "Replacement": "False",
        "ChangeSetId": null,
        "Scope": [
          "Properties"
        ],
        "Details": [
          {
            "Target": {
              "Attribute": "Properties",
              "Name": "Timeout",
              "RequiresRecreation": "Never"
            },
            "Evaluation": "Static",
            "ChangeSource": "DirectModification"
          }
        ]
      }
    }
  ],
  "ChangeSetName": "my-nested-stack-change-set",
  "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
  "ParentChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "RootChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "IncludeNestedStacks": true,
```

```
"StackName": "my-nested-stack",
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "UNAVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": "Executable from root change set",
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
    "CAPABILITY_IAM"
],
"Tags": null
}
```

execute-change-set

- [execute-change-set](#) – Cria ou atualiza uma pilha usando as informações de entrada fornecidas quando o conjunto de alterações especificado foi criado. Para criar um conjunto de alterações para toda a hierarquia de pilha, é necessário especificar a opção `--include-nested-stacks` durante a operação `create-change-set`. Para obter mais informações, consulte [Executar um conjunto de alterações para uma pilha do CloudFormation](#).

Note

`execute-change-set` deve ser executado com base no conjunto raiz de alterações e aplicará o conjunto de alterações em toda a hierarquia de pilhas.

O exemplo de AWS CLI a seguir executa um conjunto de alterações para a pilha raiz especificada.

```
aws cloudformation execute-change-set \
  --stack-name my-root-stack \
  --change-set-name my-root-stack-change-set
```

delete-change-set

- [delete-change-set](#) – Exclui o conjunto de alterações especificado. A exclusão de conjuntos de alterações garante que ninguém use o conjunto de alterações errado. A exclusão de conjuntos de alterações é assíncrona para conjuntos de alterações criados com a opção `--include-nested-stacks`. Para obter mais informações, consulte [Excluir um conjunto de alterações para uma pilha do CloudFormation](#).

Note

`delete-change-set` deve ser executado com base no conjunto raiz de alterações e excluirá toda a hierarquia de conjuntos de alterações. As pilhas aninhadas com status `REVIEW_IN_PROGRESS` também serão excluídas se tiverem sido criadas durante a operação `create-change-set`.

O exemplo de AWS CLI a seguir exclui o conjunto de alterações para a pilha raiz especificada.

```
aws cloudformation delete-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set
```

Validar implantações de pilhas

Com a validação de pré-implantação, você pode identificar e resolver possíveis problemas de implantação antes de executar seus conjuntos de alterações do CloudFormation. Esse recurso valida seus modelos em relação a cenários de falha comuns, ajudando você a detectar problemas no início do ciclo de desenvolvimento.

Tópicos

- [Como funciona a validação pré-implantação](#)
- [Considerações](#)
- [Pré-requisitos](#)
- [Validar a implantação de uma pilha \(console\)](#)
- [Validar uma implantação de pilha \(AWS CLI\)](#)
- [Tipos de validação](#)

- [Limitações do recurso](#)

Como funciona a validação pré-implantação

A validação pré-implantação envolve as seguintes fases:

1. Criar seu conjunto de alterações: gere um conjunto de alterações como você faria normalmente para as atualizações da pilha do CloudFormation. A validação pré-implantação é habilitada por padrão ao criar seu conjunto de alterações.
2. Execução da validação: o CloudFormation executa várias verificações de validação em relação ao seu modelo e ambiente de destino. Atualmente, há suporte para três tipos de validação: validação de sintaxe de propriedade em relação a esquemas de recursos, detecção de conflito de nome de recurso com recursos existentes e validação de esvaziamento do bucket do S3 para operações de exclusão.
3. Analisar os resultados da validação: o CloudFormation fornece feedback detalhado sobre quaisquer problemas encontrados, incluindo um caminho preciso que identifica a localização do problema no modelo, eliminando a depuração manual do modelo.
4. Resolver problemas: resolva os problemas identificados atualizando seus modelos ou resolvendo conflitos antes de prosseguir com a implantação.
5. Executar com confiança: implante seu conjunto de alterações sabendo que cenários de falha comuns foram validados antecipadamente.

Considerações

Ao usar a validação de pré-implantação, tenha em mente o seguinte:

- A validação pré-implantação foca os três cenários comuns de falha na implantação. Isso não garante que sua implantação será bem-sucedida, mas reduz a probabilidade de falhas comuns.
- Os modos de validação têm um comportamento diferente:
 - O modo FAIL impede a execução do conjunto de alterações quando a validação detecta erros, garantindo que modelos problemáticos não possam prosseguir com a implantação. Isso se aplica a erros de sintaxe de propriedades e conflitos de nomenclatura de recursos.
 - O modo WARN permite que a criação do conjunto de alterações seja bem-sucedida apesar das falhas de validação, fornecendo avisos que os desenvolvedores podem revisar e resolver antes

da execução. Isso se aplica a violações de restrições, como o esvaziamento do bucket do S3, que pode ser resolvido por meio de intervenção manual.

- Os resultados da validação estão vinculados ao conjunto de alterações específico. Se você modificar seu modelo, precisará criar um novo conjunto de alterações para obter resultados de validação atualizados.
- A validação do bucket do S3 verifica apenas a presença do objeto, não as políticas de bucket ou outras restrições que possam impedir a exclusão.

Pré-requisitos

Para usar a validação de pré-implantação, você deve ter:

- As permissões necessárias do IAM para criar conjuntos de alterações e ler recursos em sua conta. Para verificar o esvaziamento do bucket do S3, você precisa da permissão `s3:ListBucket`.
- Acesso às Regiões da AWS onde suas pilhas estão implantadas.
- Modelos do CloudFormation que você deseja validar antes da implantação.

Validar a implantação de uma pilha (console)

Use o procedimento a seguir para validar sua implantação usando o console.

Para validar um modelo antes da implantação

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha está localizada.
3. Na página Pilhas, escolha a pilha em execução para a qual você deseja criar um conjunto de alterações.
4. No painel de detalhes da pilha, escolha Atualizar pilha e selecione Criar um conjunto de alterações.
5. Na página Criar conjunto de alterações para **stack-name**, carregue seu modelo atualizado ou especifique a origem do modelo.
6. Escolha Avançar para prosseguir com as etapas restantes de configuração do conjunto de alterações.

7. Se o modelo incluir recursos do IAM, escolha Reconheço que o CloudFormation pode criar recursos do IAM em Recursos. Os recursos do IAM podem modificar as permissões na sua conta da AWS. Analise esses recursos para garantir que você está permitindo apenas as ações que deseja. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
8. Na página Revisar, escolha Criar conjunto de alterações.
9. O CloudFormation criará o conjunto de alterações e executará as verificações de validação. Analise os resultados da validação na guia Validação de implantação.
10. Se a validação for aprovada ou os avisos forem atendidos, escolha Executar conjunto de alterações para implantar suas alterações.
11. Se a validação falhar, corrija os problemas e crie um novo conjunto de alterações para revalidar sua implantação.

Validar uma implantação de pilha (AWS CLI)

Os comandos da AWS CLI para a validação de pré-implantação incluem:

- [create-change-set](#) que valida automaticamente durante a criação do conjunto de alterações.
- [describe-change-set](#) para verificar o status do conjunto de alterações
- [describe-events](#) para revisar os resultados da validação.

Siga o procedimento a seguir para validar a implantação de pilha usando a AWS CLI.

Para validar um modelo antes da implantação

1. Use o comando [create-change-set](#):

```
aws cloudformation create-change-set \  
  --stack-name MyStack \  
  --change-set-name MyChangeSet \  
  --change-set-type "CREATE" \  
  --template-body file://updated-template.yaml
```

O comando retorna o ARN do conjunto de alterações e o ARN da pilha.

2. Use o comando [describe-events](#) com o ARN do conjunto de alterações ou o nome do conjunto de alterações para revisar o status e os resultados da validação.

```
aws cloudformation describe-events \
  --change-set-name "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
  MyChangeSet/94498df5-1afb-43b1-9869-9f82b2d877ac"
```

Exemplo de saída de um erro de validação:

```
{
  "OperationEvents":[
    {
      "EventId":"9b5c9a29-4704-4ad0-8082-afb49418d55b",
      "StackId":"arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/
c3908380-b357-11f0-a97f-0ad08f35df65",
      "OperationId":"f558b823-e1e3-4de3-a222-e6b930ddcad4",
      "OperationType":"CREATE_CHANGESET",
      "OperationStatus":"FAILED",
      "EventType":"STACK_EVENT",
      "Timestamp":"2025-10-27T17:10:02.923Z",
      "StartTime":"2025-10-27T17:09:57.537Z",
      "EndTime":"2025-10-27T17:10:02.923Z"
    },
    {
      "EventId":"2d8c3262-3468-4283-82fb-6e780e9e4f1d",
      "StackId":"arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/
c3908380-b357-11f0-a97f-0ad08f35df65",
      "OperationId":"f558b823-e1e3-4de3-a222-e6b930ddcad4",
      "OperationType":"CREATE_CHANGESET",
      "EventType":"VALIDATION_ERROR",
      "LogicalResourceId":"NotificationBucket",
      "PhysicalResourceId":"",
      "ResourceType":"AWS::S3::Bucket",
      "Timestamp":"2025-10-27T17:10:02.461Z",
      "ValidationFailureMode":"FAIL",
      "ValidationName":"PROPERTY_VALIDATION",
      "ValidationStatus":"FAILED",
      "ValidationStatusReason":"#/NotificationConfiguration/
QueueConfigurations/0: required key [Event] not found",
      "ValidationPath":"/Resources/NotificationBucket/Properties/
NotificationConfiguration/QueueConfigurations/0"
    }
  ]
}
```

3. Solucione quaisquer erros de validação atualizando seu modelo e crie um novo conjunto de alterações.
4. Depois que a validação for aprovada, execute o conjunto de alterações:

```
aws cloudformation execute-change-set \  
  --change-set-name MyChangeSet \  
  --stack-name MyStack
```

Tipos de validação

A validação pré-implantação inclui os seguintes tipos de verificações:

- Validação da sintaxe da propriedade: valida as propriedades dos recursos em relação aos esquemas de recursos da AWS. Ela verifica as propriedades obrigatórias e os valores válidos das propriedades e identifica combinações de propriedades obsoletas ou não compatíveis.
- Detecção de conflitos de nomes de recursos: verifica se há conflitos de nomenclatura com recursos existentes da AWS. Ela valida se os nomes dos recursos atendem aos requisitos de nomenclatura da AWS e identifica possíveis conflitos antes das tentativas de implantação.
- Validação do esvaziamento do bucket do S3: avisa ao tentar excluir buckets do S3 que contêm objetos. Ela fornece contagens de objetos para ajudar a avaliar o impacto da exclusão e ajuda a evitar falhas comuns de exclusão do S3.

Cada tipo de validação fornece mensagens de erro específicas e a localização do erro no modelo para ajudá-lo a resolver problemas rapidamente.

Limitações do recurso

Os seguintes tipos de recurso não são compatíveis com a validação pré-implantação:

- `AWS::ApiGatewayV2::ApiGatewayManagedOverrides`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AppMesh::GatewayRoute`
- `AWS::AppMesh::Mesh`
- `AWS::AppMesh::Route`
- `AWS::AppMesh::VirtualGateway`

- `AWS::AppMesh::VirtualNode`
- `AWS::AppMesh::VirtualRouter`
- `AWS::AppMesh::VirtualService`
- `AWS::AppStream::Fleet`
- `AWS::AppStream::Stack`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::AppSync::ApiCache`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AutoScalingPlans::ScalingPlan`
- `AWS::Budgets::Budget`
- `AWS::CertificateManager::Certificate`
- `AWS::Cloud9::EnvironmentEC2`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeBuild::ReportGroup`
- `AWS::CodeBuild::SourceCredential`
- `AWS::CodeCommit::Repository`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodeStar::GitHubRepository`
- `AWS::Config::ConfigurationRecorder`

- `AWS::Config::DeliveryChannel`
- `AWS::Config::OrganizationConfigRule`
- `AWS::Config::RemediationConfiguration`
- `AWS::DAX::Cluster`
- `AWS::DAX::ParameterGroup`
- `AWS::DAX::SubnetGroup`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DLM::LifecyclePolicy`
- `AWS::DMS::Certificate`
- `AWS::DMS::Endpoint`
- `AWS::DMS::EventSubscription`
- `AWS::DMS::ReplicationInstance`
- `AWS::DMS::ReplicationSubnetGroup`
- `AWS::DMS::ReplicationTask`
- `AWS::DocDB::DBCluster`
- `AWS::DocDB::DBClusterParameterGroup`
- `AWS::DocDB::DBInstance`
- `AWS::DocDB::DBSubnetGroup`
- `AWS::DocDB::EventSubscription`
- `AWS::EC2::ClientVpnAuthorizationRule`
- `AWS::EC2::ClientVpnEndpoint`
- `AWS::EC2::ClientVpnRoute`
- `AWS::EC2::ClientVpnTargetNetworkAssociation`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticLoadBalancing::LoadBalancer`

- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::FSx::FileSystem`
- `AWS::FSx::Snapshot`
- `AWS::FSx::StorageVirtualMachine`
- `AWS::FSx::Volume`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::CustomEntityType`
- `AWS::Glue::DataCatalogEncryptionSettings`
- `AWS::Glue::DataQualityRuleset`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::MLTransform`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::TableOptimizer`
- `AWS::Glue::Workflow`
- `AWS::Greengrass::ConnectorDefinition`
- `AWS::Greengrass::ConnectorDefinitionVersion`
- `AWS::Greengrass::CoreDefinition`
- `AWS::Greengrass::CoreDefinitionVersion`
- `AWS::Greengrass::DeviceDefinition`
- `AWS::Greengrass::DeviceDefinitionVersion`
- `AWS::Greengrass::FunctionDefinition`
- `AWS::Greengrass::FunctionDefinitionVersion`

- `AWS::Greengrass::Group`
- `AWS::Greengrass::GroupVersion`
- `AWS::Greengrass::LoggerDefinition`
- `AWS::Greengrass::LoggerDefinitionVersion`
- `AWS::Greengrass::ResourceDefinition`
- `AWS::Greengrass::ResourceDefinitionVersion`
- `AWS::Greengrass::SubscriptionDefinition`
- `AWS::Greengrass::SubscriptionDefinitionVersion`
- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTThingsGraph::FlowTemplate`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisAnalytics::ApplicationReferenceDataSource`
- `AWS::KinesisAnalyticsV2::ApplicationCloudWatchLoggingOption`
- `AWS::KinesisAnalyticsV2::ApplicationOutput`
- `AWS::KinesisAnalyticsV2::ApplicationReferenceDataSource`
- `AWS::LakeFormation::DataLakeSettings`
- `AWS::LakeFormation::Permissions`
- `AWS::LakeFormation::Resource`
- `AWS::ManagedBlockchain::Member`
- `AWS::ManagedBlockchain::Node`
- `AWS::MediaConvert::JobTemplate`
- `AWS::MediaConvert::Preset`
- `AWS::MediaConvert::Queue`
- `AWS::MediaLive::Channel`
- `AWS::MediaLive::Input`

- `AWS::MediaLive::InputSecurityGroup`
- `AWS::MediaStore::Container`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::Pinpoint::ADMChannel`
- `AWS::Pinpoint::APNSChannel`
- `AWS::Pinpoint::APNSSandboxChannel`
- `AWS::Pinpoint::APNSVoipChannel`
- `AWS::Pinpoint::APNSVoipSandboxChannel`
- `AWS::Pinpoint::App`
- `AWS::Pinpoint::ApplicationSettings`
- `AWS::Pinpoint::BaiduChannel`
- `AWS::Pinpoint::Campaign`
- `AWS::Pinpoint::EmailChannel`
- `AWS::Pinpoint::EmailTemplate`
- `AWS::Pinpoint::EventStream`
- `AWS::Pinpoint::GCMChannel`
- `AWS::Pinpoint::PushTemplate`
- `AWS::Pinpoint::Segment`
- `AWS::Pinpoint::SMSChannel`
- `AWS::Pinpoint::SmsTemplate`
- `AWS::Pinpoint::VoiceChannel`
- `AWS::PinpointEmail::ConfigurationSet`
- `AWS::PinpointEmail::ConfigurationSetEventDestination`
- `AWS::PinpointEmail::DedicatedIpPool`

- `AWS::PinpointEmail::Identity`
- `AWS::QLDB::Ledger`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SageMaker::CodeRepository`
- `AWS::SageMaker::EndpointConfig`
- `AWS::SageMaker::Model`
- `AWS::SageMaker::NotebookInstance`
- `AWS::SageMaker::NotebookInstanceLifecycleConfig`
- `AWS::SageMaker::Workteam`
- `AWS::SDB::Domain`
- `AWS::ServiceCatalog::AcceptedPortfolioShare`
- `AWS::ServiceCatalog::LaunchRoleConstraint`
- `AWS::ServiceCatalog::Portfolio`
- `AWS::ServiceCatalog::StackSetConstraint`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptFilter`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::SSM::MaintenanceWindow`
- `AWS::SSM::MaintenanceWindowTarget`
- `AWS::SSM::MaintenanceWindowTask`

- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFRegional::ByteMatchSet`
- `AWS::WAFRegional::GeoMatchSet`
- `AWS::WAFRegional::IPSet`
- `AWS::WAFRegional::RateBasedRule`
- `AWS::WAFRegional::RegexPatternSet`
- `AWS::WAFRegional::Rule`
- `AWS::WAFRegional::SizeConstraintSet`
- `AWS::WAFRegional::SqlInjectionMatchSet`
- `AWS::WAFRegional::WebACL`
- `AWS::WAFRegional::WebACLAssociation`
- `AWS::WAFRegional::XssMatchSet`
- `AWS::WorkSpaces::Workspace`
- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::DomainNameAccessAssociation`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::ApplicationEntitlementAssociation`
- `AWS::AppStream::ApplicationFleetAssociation`
- `AWS::AppSync::DomainNameApiAssociation`
- `AWS::AppSync::SourceApiAssociation`
- `AWS::CleanRooms::ConfiguredTableAssociation`
- `AWS::CleanRooms::IdNamespaceAssociation`
- `AWS::CodeGuruReviewer::RepositoryAssociation`
- `AWS::Cognito::IdentityPoolRoleAttachment`

- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Connect::IntegrationAssociation`
- `AWS::Deadline::QueueFleetAssociation`
- `AWS::Deadline::QueueLimitAssociation`
- `AWS::EC2::EIPAssociation`
- `AWS::EC2::EnclaveCertificateIamRoleAssociation`
- `AWS::EC2::GatewayRouteTableAssociation`
- `AWS::EC2::IPAMResourceDiscoveryAssociation`
- `AWS::EC2::IpPoolRouteTableAssociation`
- `AWS::EC2::LocalGatewayRouteTableVPCAssociation`
- `AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation`
- `AWS::EC2::NetworkInterfaceAttachment`
- `AWS::EC2::RouteServerAssociation`
- `AWS::EC2::SecurityGroupVpcAssociation`
- `AWS::EC2::SubnetNetworkAclAssociation`
- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::TransitGatewayMulticastDomainAssociation`
- `AWS::EC2::TransitGatewayPeeringAttachment`
- `AWS::EC2::TransitGatewayRouteTableAssociation`
- `AWS::EC2::TransitGatewayVpcAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VolumeAttachment`
- `AWS::ECS::ClusterCapacityProviderAssociations`
- `AWS::EKS::PodIdentityAssociation`
- `AWS::FSx::DataRepositoryAssociation`
- `AWS::FSx::S3AccessPointAttachment`

- `AWS::GlobalAccelerator::CrossAccountAttachment`
- `AWS::LakeFormation::TagAssociation`
- `AWS::NetworkFirewall::VpcEndpointAssociation`
- `AWS::NetworkManager::ConnectAttachment`
- `AWS::NetworkManager::CustomerGatewayAssociation`
- `AWS::NetworkManager::DirectConnectGatewayAttachment`
- `AWS::NetworkManager::LinkAssociation`
- `AWS::NetworkManager::SiteToSiteVpnAttachment`
- `AWS::NetworkManager::TransitGatewayRouteTableAttachment`
- `AWS::NetworkManager::VpcAttachment`
- `AWS::Notifications::ChannelAssociation`
- `AWS::Notifications::ManagedNotificationAccountContactAssociation`
- `AWS::Notifications::ManagedNotificationAdditionalChannelAssociation`
- `AWS::Notifications::OrganizationalUnitAssociation`
- `AWS::ResourceExplorer2::DefaultViewAssociation`
- `AWS::Route53Profiles::ProfileAssociation`
- `AWS::Route53Profiles::ProfileResourceAssociation`
- `AWS::Route53Resolver::FirewallRuleGroupAssociation`
- `AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation`
- `AWS::Route53Resolver::ResolverRuleAssociation`
- `AWS::SSM::Association`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecurityHub::PolicyAssociation`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::ServiceActionAssociation`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkResourceAssociation`

- `AWS::VpcLattice::ServiceNetworkServiceAssociation`
- `AWS::VpcLattice::ServiceNetworkVpcAssociation`
- `AWS::WAFv2::WebACLAssociation`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::WorkspacesInstances::VolumeAssociation`
- `AWS::IAM::Policy`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::QueuePolicy`
- `AWS::EC2::NetworkAclEntry`
- `AWS::EC2::VPNGatewayRoutePropagation`
- `AWS::CloudFormation::Stack`
- `AWS::CloudWatch::MetricStream`
- `AWS::WorkSpaces::ConnectionAlias`
- `AWS::IoT::ProvisioningTemplate`
- `AWS::MediaPackage::Channel`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::Route53Resolver::ResolverQueryLoggingConfig`
- `AWS::NetworkManager::TransitGatewayRegistration`
- `AWS::ImageBuilder::Image`
- `AWS::Config::ConformancePack`
- `AWS::S3::AccessPoint`
- `AWS::CodeStarConnections::Connection`
- `AWS::CloudFront::CachePolicy`
- `AWS::FMS::NotificationChannel`
- `AWS::ImageBuilder::InfrastructureConfiguration`
- `AWS::Detective::Graph`
- `AWS::EC2::CarrierGateway`
- `AWS::CloudWatch::CompositeAlarm`
- `AWS::CodeArtifact::Repository`
- `AWS::GroundStation::DataflowEndpointGroup`

- `AWS::ElasticLoadBalancingV2::Listener`
- `AWS::ImageBuilder::ImageRecipe`
- `AWS::NetworkManager::Device`
- `AWS::Kendra::DataSource`
- `AWS::Timestream::Database`
- `AWS::CodeGuruProfiler::ProfilingGroup`
- `AWS::Lambda::EventSourceMapping`
- `AWS::ECR::Repository`
- `AWS::WAFv2::IPSet`
- `AWS::GameLift::Alias`
- `AWS::IoTSiteWise::Asset`
- `AWS::OpsWorksCM::Server`
- `AWS::IoT::Authorizer`
- `AWS::WAFv2::RuleGroup`
- `AWS::NetworkManager::Site`
- `AWS::ResourceGroups::Group`
- `AWS::MediaPackage::PackagingConfiguration`
- `AWS::ImageBuilder::ImagePipeline`
- `AWS::ECS::TaskDefinition`
- `AWS::Macie::CustomDataIdentifier`
- `AWS::MediaPackage::OriginEndpoint`
- `AWS::Logs::LogGroup`
- `AWS::CodeArtifact::Domain`
- `AWS::Kendra::Faq`
- `AWS::ECS::TaskSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::ECS::Cluster`
- `AWS::SSO::Assignment`
- `AWS::GlobalAccelerator::Listener`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`

- `AWS::RDS::DBProxy`
- `AWS::EC2::FlowLog`
- `AWS::ImageBuilder::Component`
- `AWS::CloudFront::RealtimeLogConfig`
- `AWS::NetworkManager::GlobalNetwork`
- `AWS::RDS::DBProxyTargetGroup`
- `AWS::WAFv2::WebACL`
- `AWS::IVS::StreamKey`
- `AWS::IVS::PlaybackKeyPair`
- `AWS::Macie::Session`
- `AWS::Route53::HealthCheck`
- `AWS::Synthetics::Canary`
- `AWS::Lambda::CodeSigningConfig`
- `AWS::EFS::AccessPoint`
- `AWS::Timestream::Table`
- `AWS::MediaPackage::PackagingGroup`
- `AWS::ECS::PrimaryTaskSet`
- `AWS::Config::ConfigurationAggregator`
- `AWS::GroundStation::Config`
- `AWS::IoTSiteWise::AssetModel`
- `AWS::SES::ConfigurationSet`
- `AWS::ImageBuilder::DistributionConfiguration`
- `AWS::Config::OrganizationConformancePack`
- `AWS::EC2::LocalGatewayRoute`
- `AWS::KMS::Key`
- `AWS::Detective::MemberInvitation`
- `AWS::EKS::FargateProfile`
- `AWS::MediaPackage::Asset`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::Macie::FindingsFilter`

- `AWS::IoT::Certificate`
- `AWS::SageMaker::MonitoringSchedule`
- `AWS::IVS::Channel`
- `AWS::Kendra::Index`
- `AWS::EventSchemas::RegistryPolicy`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::GlobalAccelerator::Accelerator`
- `AWS::EC2::PrefixList`
- `AWS::GameLift::GameServerGroup`
- `AWS::NetworkManager::Link`
- `AWS::EFS::FileSystem`
- `AWS::Route53::HostedZone`
- `AWS::GroundStation::MissionProfile`
- `AWS::KMS::Alias`
- `AWS::FMS::Policy`
- `AWS::SSO::PermissionSet`
- `AWS::StepFunctions::StateMachine`
- `AWS::QLDB::Stream`
- `AWS::IoTSiteWise::Gateway`
- `AWS::ECS::Service`
- `AWS::ECS::CapacityProvider`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::EC2Fleet`
- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::S3::BucketPolicy`

- `AWS::SNS::Topic`
- `AWS::SNS::Subscription`
- `AWS::RDS::DBInstance`
- `AWS::RDS::DBParameterGroup`
- `AWS::RDS::DBCluster`
- `AWS::RDS::DBClusterParameterGroup`
- `AWS::RDS::DBSubnetGroup`
- `AWS::RDS::EventSubscription`
- `AWS::RDS::GlobalCluster`
- `AWS::RDS::OptionGroup`
- `AWS::Neptune::DBInstance`
- `AWS::Neptune::DBParameterGroup`
- `AWS::Neptune::DBCluster`
- `AWS::Neptune::DBClusterParameterGroup`
- `AWS::Neptune::DBSubnetGroup`
- `AWS::Redshift::Cluster`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::Redshift::ClusterSubnetGroup`
- `AWS::Redshift::EndpointAccess`
- `AWS::Redshift::EndpointAuthorization`
- `AWS::Redshift::EventSubscription`
- `AWS::Redshift::ScheduledAction`
- `AWS::ElastiCache::SubnetGroup`

Atualizar pilhas diretamente

Quando você deseja implementar rapidamente atualizações em sua pilha, execute uma atualização direta. Com uma atualização direta, você envia um modelo ou parâmetros de entrada que especificam as atualizações dos recursos na pilha e o CloudFormation as implanta imediatamente. Se você quiser usar um modelo para fazer as atualizações, é possível modificar o modelo atual e armazená-lo localmente ou em um bucket do Amazon S3.

Para propriedades de recursos que não suportam atualizações, você deve manter os valores atuais. Para visualizar as alterações que o CloudFormation fará em sua pilha antes de atualizá-la, use conjuntos de alterações. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

Ao atualizar uma pilha, o CloudFormation pode interromper os recursos ou substituir os recursos atualizados, dependendo de quais propriedades você atualiza. Para obter mais informações sobre os comportamentos de atualização de recursos, consulte [Compreender atualização de comportamentos de recursos da pilha](#).

Para atualizar uma pilha de (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Na página Pilhas, selecione a pilha em execução que você deseja atualizar.
4. No painel de detalhes da pilha, escolha Update (Atualizar).
5. Se você não modificou o modelo de pilha, selecione Usar modelo existente e escolha Avançar.

Se você modificou o modelo, selecione Substituir o modelo existente e especifique a localização do modelo atualizado na seção Especificar modelo:

- Para um modelo armazenado localmente em seu computador, selecione Upload a template file (Fazer upload de um arquivo de modelo). Selecione Escolher arquivo para navegar até o arquivo e selecioná-lo, depois escolha Próximo.

Note

Se você carregar um arquivo de modelo local, o CloudFormation o carregará dele em um bucket do Amazon Simple Storage Service (Amazon S3) na sua Conta da AWS. Se você ainda não tiver um bucket do S3 que foi criado pelo CloudFormation, ele criará um bucket exclusivo para cada Região na qual você faz upload de um arquivo de modelo. Se você já tiver um bucket do S3 que foi criado pelo CloudFormation em sua Conta da AWS, o CloudFormation adicionará o modelo a esse bucket. Considerações para se ter em mente sobre buckets do S3 criados pelo CloudFormation

- Os buckets são acessíveis a qualquer pessoa com permissões do Amazon S3 na sua conta Conta da AWS.

- O CloudFormation cria os buckets com a criptografia no lado do servidor habilitada por padrão, criptografando, assim, todos os objetos armazenados no bucket.

Gerencie diretamente as opções de criptografia para buckets criados pelo CloudFormation usando, por exemplo, o console do Amazon S3 em <https://console.aws.amazon.com/s3/>, ou a AWS CLI. Para obter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

- É possível utilizar seu próprio bucket e gerenciar suas permissões carregando modelos manualmente no Amazon S3. Ao criar ou atualizar uma pilha, especifique o URL de um arquivo de modelo do Amazon S3.

- Para um modelo armazenado no bucket do Amazon S3, escolha Amazon S3 URL (URL do Amazon S3). Digite ou cole o URL do modelo e escolha Next (Próximo).

Se você tiver um modelo em um bucket com versionamento habilitado, poderá especificar uma versão específica do modelo anexando `?versionId=version-id` ao URL. Para obter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado](#) no Guia do usuário do Amazon Simple Storage Service.

Se algum problema de sintaxe for detectado, o console fornecerá mensagens de erro que ajudarão você a corrigir o modelo.

6. Se o seu modelo contém parâmetros, na página Especificar detalhes da pilha, é possível inserir ou modificar os valores dos parâmetros, depois escolha Próximo.

O CloudFormation preenche cada parâmetro com o valor que está definido atualmente na pilha, com exceção de parâmetros declarados com o atributo NoEcho. No entanto, ainda é possível usar os valores atuais ao marcar Usar valor existente.

Para obter mais informações sobre o uso de NoEcho para mascarar informações confidenciais, além do uso de parâmetros dinâmicos para gerenciar segredos, consulte a prática recomendada [Não incorporar credenciais em seus modelos](#).

7. Na página Configure stack options (Configurar opções da pilha), é possível atualizar as tags e as permissões aplicadas à pilha e modificar opções avançadas como política de pilha, configurar a reversão ou atualizar o tópico de notificação do Amazon SNS. Para saber mais sobre essas opções, consulte [Configurar opções da pilha](#).

8. Se o seu modelo contém recursos do IAM, em Recursos, escolha Eu reconheço que este modelo pode criar recursos do IAM para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
9. Escolha Próximo para continuar.
10. Analise as informações e as alterações da pilha que você enviou.

Verifique se você enviou as informações corretas, como os valores de parâmetro ou modelo de URL correto.

Na seção Visualização do conjunto de alterações, verifique se o CloudFormation fará todas as alterações esperadas. Por exemplo, é possível verificar se o CloudFormation adiciona, remove e modifica os recursos que você pretende adicionar, remover ou modificar. O CloudFormation gera esta pré-visualização criando um conjunto de alterações para a pilha. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

11. Quando estiver satisfeito com as suas alterações, clique em Update stack (Atualizar pilha).

 Note

Nesse ponto, você também tem a opção de visualizar o conjunto de alterações para rever as atualizações propostas mais cuidadosamente. Para isso, escolha View change set (Visualizar conjunto de alterações) em vez de Update stack (Atualizar pilha). O CloudFormation mostra o conjunto de alterações gerado com base nas suas atualizações. Quando estiver pronto para executar a atualização da pilha, clique em Executar.

O CloudFormation exibe a página de detalhes para a sua pilha, com o painel Events (Eventos) selecionado. Sua pilha agora tem um status de UPDATE_IN_PROGRESS. Depois que o CloudFormation tiver concluído com êxito a atualização da pilha, ele definirá o status da pilha como UPDATE_COMPLETE.

Caso a atualização da pilha falhe, o CloudFormation reverterá automaticamente as alterações e definirá o status como UPDATE_ROLLBACK_COMPLETE.

 Note

É possível cancelar uma atualização enquanto ela está no estado UPDATE_IN_PROGRESS. Para obter mais informações, consulte [Cancelar uma atualização de pilha](#).

Para atualizar uma pilha usando a linha de comando

Você pode usar um dos comandos a seguir:

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

Para obter exemplos de uso de linha de comando para atualizar uma pilha, consulte [Exemplos de comandos de operação de pilha do CloudFormation para o AWS CLI e o PowerShell](#).

Cancelar uma atualização de pilha

Após o início de uma atualização de pilha, é possível cancelar a atualização da pilha se ela ainda estiver no estado UPDATE_IN_PROGRESS. Após a conclusão de uma atualização, não será possível cancelá-la. No entanto, uma pilha pode ser atualizada novamente com as definições anteriores.

Se você cancelar a atualização de uma pilha, ela será revertida para a configuração de pilha existente antes de iniciar a atualização da pilha.

Tópicos

- [Para cancelar uma atualização de pilha \(console\)](#)
- [Para cancelar uma atualização de pilha \(AWS CLI\)](#)

Para cancelar uma atualização de pilha (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha está localizada.

3. Na página Pilhas escolha a pilha que está sendo atualizada. Seu status deve ser `UPDATE_IN_PROGRESS`.
4. Escolha Stack actions (Ações da pilha) e Cancel update stack (Cancelar atualização da pilha).
5. Para continuar com o cancelamento da atualização, clique em Cancelar atualização. Caso contrário, clique em Cancelar para retomar a atualização.

A pilha continua para o estado `UPDATE_ROLLBACK_IN_PROGRESS`. Após a conclusão do cancelamento da atualização, a pilha é definida para `UPDATE_ROLLBACK_COMPLETE`.

Para cancelar uma atualização de pilha (AWS CLI)

Use o comando [cancel-update-stack](#) para cancelar uma atualização. Para obter mais informações, consulte [Cancelar uma atualização de pilha](#).

Excluir uma pilha no console do CloudFormation

Se não precisar mais dos recursos em uma pilha, você poderá excluir a pilha inteira.

Ao excluir uma pilha, o CloudFormation exclui todos os recursos dessa pilha, a menos que você use uma política de exclusão para reter recursos específicos. Para obter mais informações, consulte o atributo [DeletionPolicy](#).

Para excluir um conjunto de pilhas (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha está localizada.
3. Na página Pilhas, escolha a pilha que deseja excluir. A pilha deve estar em execução no momento.
4. Escolha Excluir.
5. Quando a confirmação for solicitada, escolha Delete (Excluir).

Note

O operação de exclusão da pilha não pode ser interrompida uma vez que a ação já tenha começado. A pilha continua para o estado `DELETE_IN_PROGRESS`.

Quando a exclusão da pilha for concluída, a pilha estará no estado DELETE_COMPLETE. Por padrão, as pilhas no estado DELETE_COMPLETE não são exibidas no console do CloudFormation. Para exibir as pilhas excluídas, é necessário alterar o filtro de visualização de pilhas, conforme descrito em [Visualizar pilhas excluídas usando o console do CloudFormation](#).

Para forçar a exclusão de uma pilha (console)

Uma exclusão de pilha pode falhar devido à falha na exclusão de um recurso da pilha. Por exemplo, o CloudFormation não excluirá um recurso do qual outra pilha também dependa. Todos os recursos que não foram excluídos permanecerão até que você possa excluir a pilha com êxito. Se a exclusão falhar e retornar um estado DELETE_FAILED, você pode tentar novamente usando um de dois métodos.

1. Na página Pilhas no console do CloudFormation, escolha a pilha que você deseja forçar a exclusão.
2. No painel de detalhes da pilha, escolha Tentar excluir novamente.
3. Escolha entre as seguintes opções:
 - Excluir essa pilha, mas reter recursos: essa opção permite selecionar os recursos específicos que não puderam ser excluídos originalmente, mas que você deseja reter durante a exclusão forçada da pilha.
 - Forçar a exclusão dessa pilha inteira: essa opção retém todos os recursos que não puderam ser excluídos e retém as dependências desses recursos.
4. Escolha Excluir para iniciar o processo de exclusão forçada com base em suas seleções.

Para revisar os recursos retidos (console)

Depois que excluir a pilha, você pode visualizar no console os recursos que foram retidos.

1. Na lista de pilhas, escolha Filtrar status e selecione Excluído.
2. Escolha a pilha excluída.
3. Escolha a guia Recursos.
4. Todos os recursos retidos mostram o Status DELETE_SKIPPED.
5. Escolha o recurso retido que você deseja revisar.

Para excluir uma pilha usando a linha de comando

Você pode usar um dos comandos a seguir:

- [delete-stack](#) (AWS CLI)
- [Remove-CFNStack](#) (AWS Tools for Windows PowerShell)

Para obter exemplos de uso de linha de comando para criar uma pilha, consulte [Exemplos de comandos de operação de pilha do CloudFormation para o AWS CLI e o PowerShell](#).

Recursos relacionados

Para obter ajuda na solução problemas de erro de exclusão de pilhas, consulte o tópico [Falha na exclusão da pilha](#) da solução de problemas.

Para obter informações sobre proteção contra a exclusão acidental de pilhas, consulte [Proteger pilhas do CloudFormation contra exclusão](#).

Visualizar pilhas excluídas usando o console do CloudFormation

Por padrão, o console do CloudFormation não exibe pilhas com o status DELETE_COMPLETE. Para exibir informações sobre pilhas excluídas, você deve alterar a visualização da pilha.

Para visualizar pilhas excluídas

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha excluída está localizada.
3. Na página Pilhas, escolha Excluídas no menu suspenso Status do filtro.

O CloudFormation lista todas as pilhas excluídas com o status DELETE_COMPLETE.

Consulte também

- [Excluir uma pilha no console do CloudFormation](#)
- [Visualizar informações da pilha no console do CloudFormation](#)

Monitorar o progresso da pilha

Esta seção descreve como monitorar uma implantação de pilha que está em andamento. O CloudFormation fornece uma lista detalhada e cronológica dos eventos de implantação, mostrando o progresso e quaisquer problemas encontrados durante a implantação.

Tópicos

- [Visualizar eventos de pilha do CloudFormation](#)
- [Visualizar eventos das pilhas por operação](#)
- [Visualizar um cronograma de implantação da pilha do CloudFormation](#)
- [Entender os eventos de criação de pilhas do CloudFormation](#)
- [Monitorar o progresso da atualização de uma pilha](#)
- [Continuar revertendo uma atualização](#)
- [Determinar a causa de uma falha na pilha](#)
- [Escolha como lidar com falhas ao provisionar recursos](#)

Visualizar eventos de pilha do CloudFormation

Você pode visualizar eventos de pilha para monitorar o progresso e o status de sua pilha e os recursos na pilha. Os eventos de pilha ajudam a entender quando os recursos estão sendo criados, atualizados ou excluídos e se a implantação de pilha está transcorrendo como esperado.

Tópicos

- [Visualizar eventos da pilha \(console\)](#)
- [Visualizar eventos da pilha \(AWS CLI\)](#)
- [Códigos de status da pilha](#)

Visualizar eventos da pilha (console)

Para visualizar os eventos da pilha

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual você criou a pilha.

3. Na página Stacks (Pilhas) do console do CloudFormation, selecione o nome da pilha. O CloudFormation exibe a página de detalhes da pilha para a pilha selecionada.
4. Escolha a guia Eventos para visualizar os eventos de pilha que o CloudFormation gerou para sua pilha.

O CloudFormation atualiza automaticamente os eventos da pilha a cada minuto. Além disso, o CloudFormation exibe o emblema Novos eventos disponíveis quando ocorrem novos eventos de pilha. Escolha o ícone de atualização para carregar esses eventos na lista. Ao visualizar eventos de criação da pilha, você pode entender a sequência de eventos que levam à criação da sua pilha (ou falha, se você está depurando a pilha).

Durante a criação da pilha, ela estará listada na página Pilhas com um status de `CREATE_IN_PROGRESS`. Após a pilha ter sido criada com êxito, seu status muda para `CREATE_COMPLETE`.

Para obter mais informações, consulte [Entender os eventos de criação de pilhas do CloudFormation](#) e [Monitorar o progresso da atualização de uma pilha](#).

Visualizar eventos da pilha (AWS CLI)

Como alternativa, você pode usar o comando [describe-stack-events](#) enquanto a pilha está sendo criada para visualizar os eventos à medida que eles são relatados.

O comando `describe-stack-events` a seguir descreve os eventos da pilha *my-stack*.

```
aws cloudformation describe-stack-events --stack-name my-stack
```

O seguinte é um exemplo de resposta.

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "7b755820-7edf-11f0-ab15-0673b09f3847",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
    }
  ]
}
```

```

    "Timestamp": "2025-08-21T22:37:56.243000+00:00",
    "ResourceStatus": "CREATE_COMPLETE",
    "ClientRequestToken": "token"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
    "EventId": "WebServer-CREATE_COMPLETE-2025-08-21T22:37:54.356Z",
    "StackName": "my-stack",
    "LogicalResourceId": "WebServer",
    "PhysicalResourceId": "i-099df76cb31b866a9",
    "ResourceType": "AWS::EC2::Instance",
    "Timestamp": "2025-08-21T22:37:54.356000+00:00",
    "ResourceStatus": "CREATE_COMPLETE",
    "ResourceProperties": "{\"UserData\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"]}]\",
    \"ClientRequestToken\": \"token\"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
    "EventId": "WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:31.226Z",
    "StackName": "my-stack",
    "LogicalResourceId": "WebServer",
    "PhysicalResourceId": "i-099df76cb31b866a9",
    "ResourceType": "AWS::EC2::Instance",
    "Timestamp": "2025-08-21T22:37:31.226000+00:00",
    "ResourceStatus": "CREATE_IN_PROGRESS",
    "ResourceStatusReason": "Resource creation Initiated",
    "ResourceProperties": "{\"UserData\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"]}]\",
    \"ClientRequestToken\": \"token\"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
    "EventId": "WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:29.210Z",
    "StackName": "my-stack",

```

```

        "LogicalResourceId": "WebServer",
        "PhysicalResourceId": "",
        "ResourceType": "AWS::EC2::Instance",
        "Timestamp": "2025-08-21T22:37:29.210000+00:00",
        "ResourceStatus": "CREATE_IN_PROGRESS",
        "ResourceProperties": "{\"UserData\": \"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXZzLWNmbi1ib290c3RyYXAKL29wdC9hdMvYmluL2Nmbi1pbml0IC12I\", \"ImageId\": \"ami-0bbc328167dee8f3c\", \"InstanceType\": \"t2.micro\", \"SecurityGroupIds\": [\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"], \"Tags\": [{\"Value\": \"Bootstrap Tutorial Web Server\", \"Key\": \"Name\"}]}\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServerSecurityGroup-CREATE_COMPLETE-2025-08-21T22:37:28.803Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServerSecurityGroup\",
        \"PhysicalResourceId\": \"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\",
        \"ResourceType\": \"AWS::EC2::SecurityGroup\",
        \"Timestamp\": \"2025-08-21T22:37:28.803000+00:00\",
        \"ResourceStatus\": \"CREATE_COMPLETE\",
        \"ResourceProperties\": \"{\\\"GroupDescription\\\":\\\"Allow HTTP access from my IP address\\\",\\\"SecurityGroupIngress\\\":[{\\\"CidrIp\\\":\\\"0.0.0.0/0\\\",\\\"Description\\\":\\\"HTTP\\\",\\\"FromPort\\\":\\\"80\\\",\\\"ToPort\\\":\\\"80\\\",\\\"IpProtocol\\\":\\\"tcp\\\"}]}\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServerSecurityGroup-CREATE_IN_PROGRESS-2025-08-21T22:37:22.626Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServerSecurityGroup\",
        \"PhysicalResourceId\": \"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\",
        \"ResourceType\": \"AWS::EC2::SecurityGroup\",
        \"Timestamp\": \"2025-08-21T22:37:22.626000+00:00\",
        \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
        \"ResourceStatusReason\": \"Resource creation Initiated\",
        \"ResourceProperties\": \"{\\\"GroupDescription\\\":\\\"Allow HTTP access from my IP address\\\",\\\"SecurityGroupIngress\\\":[{\\\"CidrIp\\\":\\\"0.0.0.0/0\\\",\\\"Description\\\":\\\"HTTP\\\",\\\"FromPort\\\":\\\"80\\\",\\\"ToPort\\\":\\\"80\\\",\\\"IpProtocol\\\":\\\"tcp\\\"}]}\",
        \"ClientRequestToken\": \"token\"
    }

```

```
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServerSecurityGroup-CREATE_IN_PROGRESS-2025-08-21T22:37:20.186Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:20.186000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP address\", \"SecurityGroupIngress\": [{\"CidrIp\":\"0.0.0.0/0\", \"Description\":\"HTTP\", \"FromPort\":\"80\", \"ToPort\":\"80\", \"IpProtocol\":\"tcp\"}]}",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "64740fe0-7edf-11f0-8a36-06453a64f325",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2025-08-21T22:37:17.819000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceStatusReason": "User Initiated",
      "ClientRequestToken": "token"
    }
  ]
}
```

Os eventos mais recentes são relatados primeiro. A tabela a seguir descreve os campos retornados pelo comando `describe-stack-events`:

Campo	Descrição
EventId	Identificador de eventos.
StackName	Nome da pilha a qual o evento corresponde.

Campo	Descrição
StackId	Identificador pilha ao qual o evento corresponde.
LogicalResourceId	Identificador do recurso lógico.
PhysicalResourceId	Identificador do recurso físico.
ResourceProperties	Propriedades do recurso.
ResourceType	Tipo de recurso.
Timestamp	Hora em que o evento ocorreu.
ResourceStatus	O status do recurso, que pode ser um dos seguintes códigos de status: CREATE_COMPLETE CREATE_FAILED CREATE_IN_PROGRESS DELETE_COMPLETE DELETE_FAILED DELETE_IN_PROGRESS DELETE_SKIPPED IMPORT_COMPLETE IMPORT_IN_PROGRESS IMPORT_ROLLBACK_COMPLETE IMPORT_ROLLBACK_FAILED IMPORT_ROLLBACK_IN_PROGRESS REVIEW_IN_PROGRESS ROLLBACK_COMPLETE ROLLBACK_FAILED ROLLBACK_IN_PROGRESS UPDATE_COMPLETE UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_FAILED UPDATE_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_FAILED UPDATE_ROLLBACK_IN_PROGRESS O status DELETE_SKIPPED se aplica aos recursos com um atributo de manutenção de política de exclusão.
DetailedStatus	O status detalhado da pilha. Se CONFIGURATION_COMPLETE estiver presente, a fase de configuração dos recursos da pilha foi concluída e a estabilização dos recursos está em andamento.
ResourceStatusReason	Mais informações sobre o status.

Códigos de status da pilha

A tabela a seguir descreve os códigos de status da pilha:

Status da pilha e status detalhado opcional	Descrição
CREATE_COMPLETE	Criação bem-sucedida de uma ou mais pilhas.
CREATE_IN_PROGRESS	Criação contínua de uma ou mais pilhas.
CREATE_FAILED	Criação malsucedida de uma ou mais pilhas. Visualize os eventos da pilha para ver as mensagens de erro associadas. As possíveis razões para uma falha na criação incluem permissões insuficientes para trabalhar com todos os recursos na pilha, valores de parâmetro rejeitados por um produto da AWS ou um tempo limite durante a criação de recursos.
DELETE_COMPLETE	Exclusão bem-sucedida de uma ou mais pilhas. As pilhas excluídas serão retidas e poderão ser visualizadas por 90 dias.
DELETE_FAILED	Exclusão malsucedida de uma ou mais pilhas. Como a exclusão falhou, alguns recursos ainda poderão estar em execução. No entanto, não é possível trabalhar com a pilha, tampouco atualizá-la. Exclua a pilha novamente ou visualize os eventos de pilha para ver as mensagens de erro associadas.
DELETE_IN_PROGRESS	Remoção contínua de uma ou mais pilhas.
REVIEW_IN_PROGRESS	Criação contínua de uma ou mais pilhas com um StackId esperado, mas sem modelos ou recursos.

 **Important**
Uma pilha com este código de status conta contra o [número máximo possível de pilhas](#).

Status da pilha e status detalhado opcional	Descrição
ROLLBACK_COMPLETE	Remoção bem-sucedida de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha. A pilha retorna ao estado operacional anterior. Quaisquer recursos criados durante a ação de criação da pilha serão excluídos. Esse status ocorre somente após uma falha de criação de uma pilha. Ele significa que todas as operações da pilha parcialmente criada foram adequadamente limpas. Ao entrar nesse estado, a única operação possível é a exclusão.
ROLLBACK_FAILED	Remoção malsucedida de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha. Exclua a pilha ou visualize os eventos de pilha para ver as mensagens de erro associadas.
ROLLBACK_IN_PROGRESS	Remoção contínua de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha.
UPDATE_COMPLETE	Atualização bem-sucedida de uma ou mais pilhas.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Remoção contínua de recursos antigos para uma ou mais pilhas após uma atualização bem-sucedida da pilha. Para atualizações de pilha que exigem a substituição de recursos, o CloudFormation cria os novos recursos primeiro e, em seguida, exclui os recursos antigos para ajudar a reduzir as interrupções com a pilha. Nesse estado, a pilha foi atualizado e é utilizável, mas o CloudFormation ainda está excluindo os recursos antigos.
UPDATE_FAILED	Atualização bem-sucedida de uma ou mais pilhas. Visualize os eventos da pilha para ver as mensagens de erro associadas.

Status da pilha e status detalhado opcional	Descrição
UPDATE_IN_PROGRESS	Atualização contínua de uma ou mais pilhas.
UPDATE_ROLLBACK_COMPLETE	Retorno bem-sucedido de uma ou mais pilhas para um estado de trabalho anterior após uma falha na atualização da pilha.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Remoção contínua de recursos novos para uma ou mais pilhas após uma atualização falha da pilha. Nesse estado, a pilha foi revertida para seu estado de trabalho anterior e é utilizável, mas o CloudFormation ainda está excluindo os recursos novos criados durante a atualização da pilha.
UPDATE_ROLLBACK_FAILED	Retorno malsucedido de uma ou mais pilhas para um estado de trabalho anterior após uma falha na atualização da pilha. Quando estiver neste estado, você pode excluir a pilha ou continuar a reversão . Talvez você precise corrigir erros para que a pilha possa retornar a um estado operacional. Se preferir, entre em contato com o Suporte para restaurar a pilha para um estado utilizável.
UPDATE_ROLLBACK_IN_PROGRESS	Retorno contínuo de uma ou mais pilhas para o estado de trabalho anterior após falha na atualização da pilha.
IMPORT_IN_PROGRESS	A operação de importação está em andamento.
IMPORT_COMPLETE	A operação de importação foi concluída com êxito para todos os recursos da pilha que oferecem suporte a <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	A importação será revertida para a configuração do modelo anterior.
IMPORT_ROLLBACK_FAILED	Houve uma falha na operação de reversão de importação para pelo menos um recurso na pilha. Os resultados serão disponibilizados para os recursos importados com êxito pelo CloudFormation.

Status da pilha e status detalhado opcional	Descrição
IMPORT_ROLLBACK_COMPLETE	A importação foi revertida com êxito para a configuração de modelo anterior.

Visualizar eventos das pilhas por operação

Você pode visualizar os eventos das pilhas agrupados por operação para entender melhor a sequência e o escopo das alterações feitas em sua pilha. O agrupamento baseado em operações ajuda você a acompanhar eventos relacionados em conjunto, facilitando o monitoramento do progresso e a solução de problemas quando eles ocorrerem.

Cada operação de pilha (criar, atualizar, excluir, reverter) recebe um ID de operação exclusivo que agrupa todos os eventos relacionados. Isso permite que você mantenha o foco em operações específicas e identifique rapidamente a causa raiz das falhas.

Tópicos

- [Pré-requisitos](#)
- [Visualizar eventos de pilhas por operação \(console\)](#)
- [Visualizar eventos de pilhas por operação \(AWS CLI\)](#)
- [Códigos de status da pilha](#)

Pré-requisitos

Para usar a API `DescribeEvents`, você deve ter as permissões necessárias do IAM para: `DescribeEvents`.

Visualizar eventos de pilhas por operação (console)

Para visualizar os eventos de pilhas agrupados por operação

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual você criou a pilha.

3. Na página Pilhas, selecione o nome da pilha. O CloudFormation exibe a página de detalhes da pilha para a pilha selecionada.
4. Escolha a guia Eventos para visualizar os eventos de pilha que o CloudFormation gerou para sua pilha.
5. Os eventos são agrupados automaticamente por ID da operação. Cada operação aparece como uma seção expansível mostrando o seu tipo, status e carimbo de data e hora.
6. Clique no ID de uma operação para abrir uma visualização detalhada mostrando somente os eventos relacionados a essa operação específica.
7. Na exibição de detalhes da operação, marque a caixa de seleção Exibir somente falhas para exibir somente eventos com falha na análise da causa raiz.

O CloudFormation atualiza automaticamente os eventos da pilha a cada minuto. O emblema Novos eventos disponíveis aparece quando ocorrem novos eventos de pilha. Escolha o ícone de atualização para carregar esses eventos na lista.

Ao visualizar os eventos da pilha agrupados por operação, você pode entender a sequência de eventos de cada operação e identificar rapidamente qual operação específica causou problemas (se você estiver depurando a pilha).

Enquanto sua operação de pilha está em execução, ela é listada com um status de `CREATE_IN_PROGRESS`, `UPDATE_IN_PROGRESS` ou `DELETE_IN_PROGRESS`. Depois que sua operação for concluída com êxito, seu status mudará para `CREATE_COMPLETE`, `UPDATE_COMPLETE` ou `DELETE_COMPLETE`.

Para obter mais informações, consulte [Entender os eventos de criação de pilhas do CloudFormation](#) e [Monitorar o progresso da atualização de uma pilha](#).

Visualizar eventos de pilhas por operação (AWS CLI)

Você pode usar o comando `describe-events` com filtragem de ID de operação para visualizar eventos de operações específicas.

Obter IDs da última operação

A descrição da pilha disponível por meio da API `describe-stacks` agora inclui informações de `LastOperations` mostrando os IDs de operações recentes e seus tipos. Isso permite que você identifique rapidamente quais operações ocorreram e seu status atual sem analisar os logs de eventos.

```
aws cloudformation describe-stacks --stack-name MyStack
```

Veja a seguir um exemplo de resposta mostrando que a última operação foi uma reversão após uma falha na operação de atualização.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/07580010-bb79-11f0-8f6c-0289bb5c804f",
      "StackName": "MyStack",
      "Description": "A simple CloudFormation template to create an S3 bucket.",

      "CreationTime": "2025-11-07T01:28:13.778000+00:00",
      "LastUpdatedTime": "2025-11-07T01:43:39.838000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "UPDATE_ROLLBACK_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Tags": [],
      "EnableTerminationProtection": false,
      "DriftInformation": {
        "StackDriftStatus": "NOT_CHECKED"
      },
      "LastOperations": [
        {
          "OperationType": "ROLLBACK",
          "OperationId": "d0f12313-7bdb-414d-a879-828a99b36f29"
        },
        {
          "OperationType": "UPDATE_STACK",
          "OperationId": "1c211b5a-4538-4dc9-bfed-e07734371e57"
        }
      ]
    }
  ]
}
```

Filtrar eventos por ID de operação

O comando `describe-events` abaixo descreve os eventos para um ID de operação específico:

```
aws cloudformation describe-events \  
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57
```

Para visualizar somente eventos com falha para solução de problemas, use o parâmetro `--filter FailedEvents=true`:

```
aws cloudformation describe-events \  
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57 \  
  --filter FailedEvents=true
```

O novo recurso de filtragem de ID de operação permite que você se concentre em operações específicas e seus eventos relacionados. Ela é particularmente útil para:

- Solução de falhas específicas: isole os eventos de uma operação com falha para entender o que deu errado.
- Monitoramento de operações de longa duração: acompanhe o progresso de atualizações complexas ou implantações de grandes pilhas.
- Auditoria de alterações: revise todos os eventos associados a uma operação de atualização específica.
- Análise da causa raiz: use o filtro de falhas para identificar rapidamente a origem dos problemas de implantação.

Códigos de status da pilha

A tabela abaixo descreve os campos retornados pelo comando `describe-events` ao usar a operação de filtragem de ID:

Campo	Descrição
EventId	Identificador de eventos.
OperationId	Identificador exclusivo para a operação que gerou esse evento.
StackName	Nome da pilha a qual o evento corresponde.
StackId	Identificador pilha ao qual o evento corresponde.

Campo	Descrição
LogicalResourceId	Identificador do recurso lógico.
PhysicalResourceId	Identificador do recurso físico.
ResourceProperties	Propriedades do recurso.
ResourceType	Tipo de recurso.
Timestamp	Hora em que o evento ocorreu.
ResourceStatus	O status do recurso (CREATE_COMPLETE , UPDATE_FAILED etc.).
DetailedStatus	O status detalhado da pilha. Se CONFIGURATION_COMPLETE estiver presente, a fase de configuração dos recursos da pilha foi concluída e a estabilização dos recursos está em andamento.
ResourceStatusReason	Mais informações sobre o status.

Visualizar um cronograma de implantação da pilha do CloudFormation

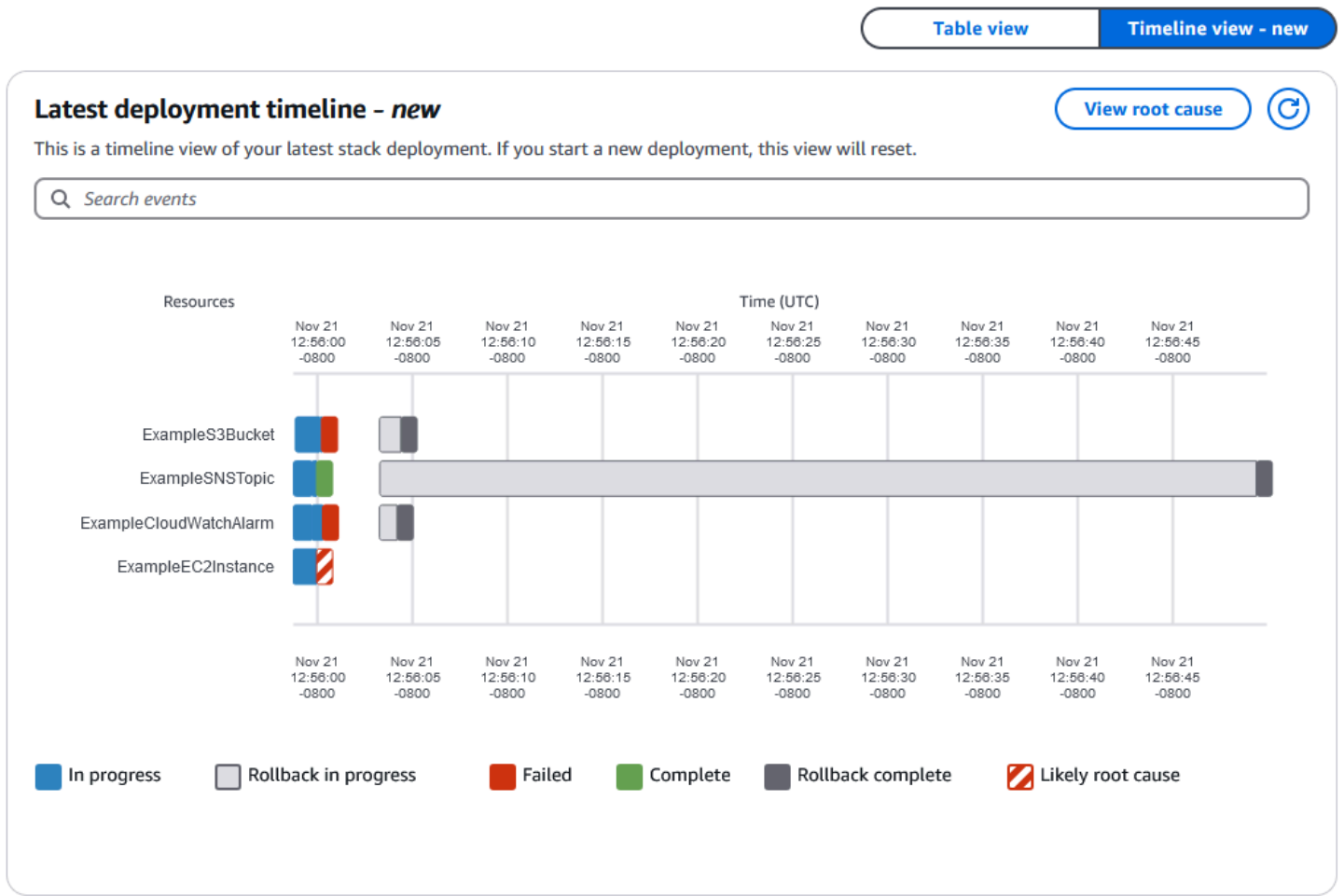
O cronograma de implantação da pilha fornece uma representação visual da cronologia da implantação da pilha. Essa visão mostra os status de implantações das pilhas e de cada um de seus recursos e quando cada status foi alterado. Os status das implantações de pilhas são representados por uma cor correspondente.

Tópicos

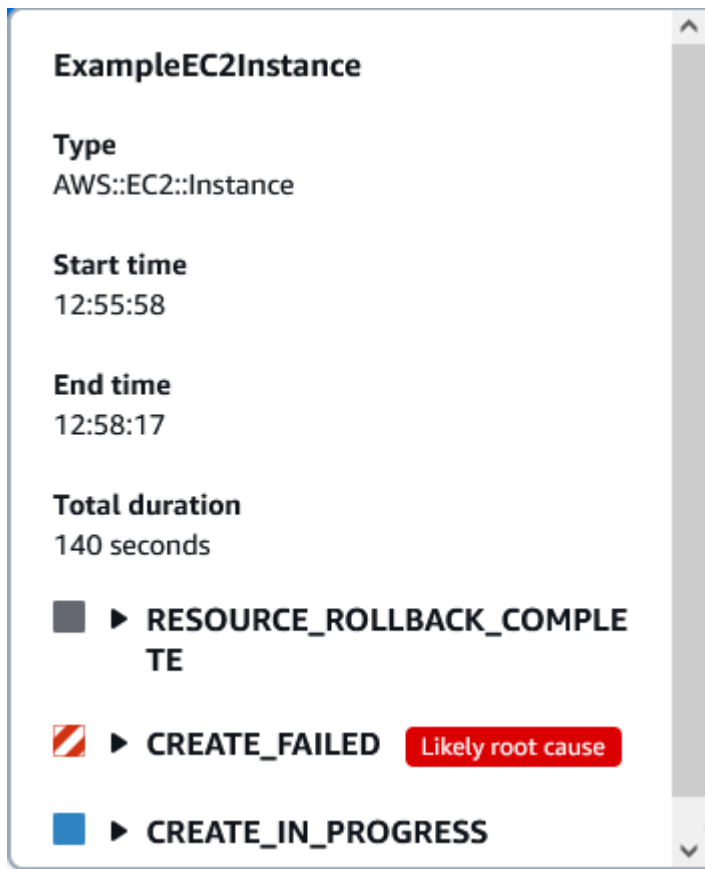
- [Entender o cronograma de implantação da pilha](#)
- [Visualizar o cronograma de implantação da pilha \(console\)](#)

Entender o cronograma de implantação da pilha

A imagem a seguir mostra o cronograma de uma implantação de pilha que não teve sucesso devido a uma falha na inicialização de um recurso de instância do Amazon EC2.



Escolher cada recurso fornece detalhes mais granulares sobre o cronograma da implantação:



Escolher um recurso mostra o Tipo, a Hora de início, a Hora de término e a Duração total da implantação. Você também encontrará a Hora de início, a Hora de término e a Duração de cada status de implantação nos menus suspensos abaixo. Se houver falha na implantação de um recurso, um motivo da falha será fornecido.

Para obter mais informações sobre os status das pilhas, consulte [Códigos de status da pilha](#).

Visualizar o cronograma de implantação da pilha (console)

Para visualizar o cronograma de implantação da pilha:

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual você criou a pilha.
3. Na página Pilhas do console do CloudFormation, escolha o nome da pilha. O CloudFormation exibe a página de detalhes da pilha para a pilha selecionada.

4. Escolha a guia Eventos para visualizar os eventos de pilha que o CloudFormation gerou para sua pilha.
5. Escolha o botão Cronograma para visualizar o cronograma da pilha.

Entender os eventos de criação de pilhas do CloudFormation

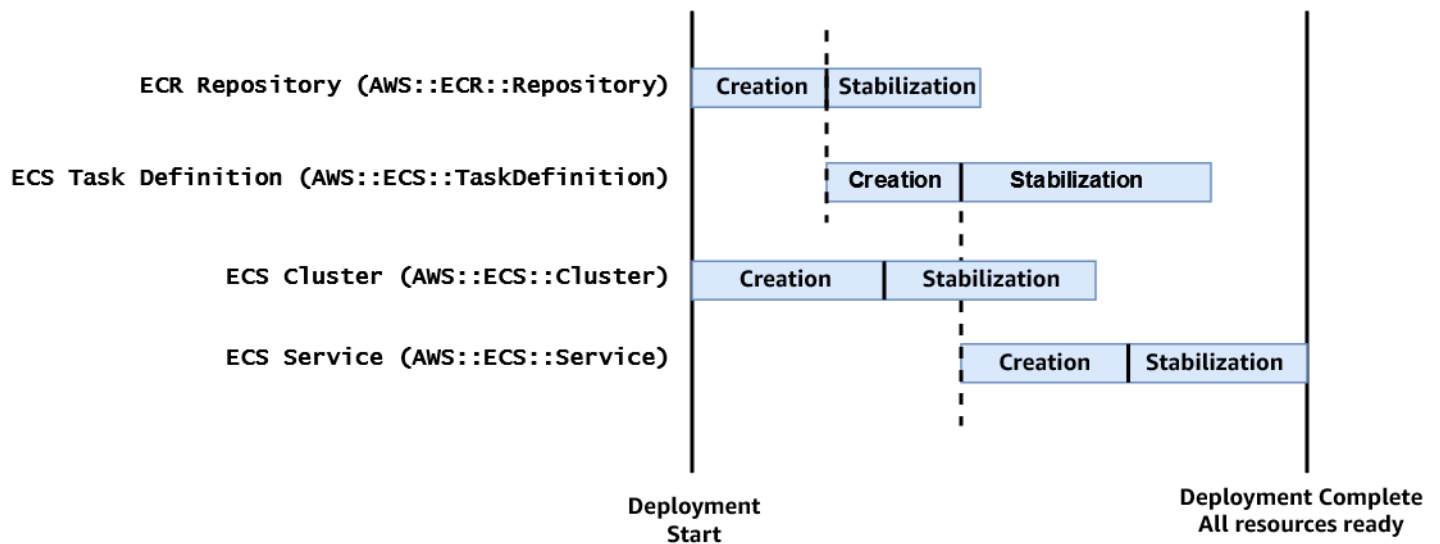
Durante a implantação da pilha, vários eventos ocorrem para criar, configurar e validar os recursos definidos no modelo da pilha. Compreender esses eventos pode ajudar você a otimizar seu processo de criação de pilhas e simplificar as implantações.

- Eventos de criação de recursos: quando cada recurso inicia o processo de criação, é definido um evento de Status de `CREATE_IN_PROGRESS`. Esse evento indica que o recurso está sendo provisionado.
- Verificação final de consistência: uma parte significativa do tempo de criação da pilha é usada na verificação final de consistência com os recursos criados pela pilha. Durante essa fase, o serviço realiza verificações internas de consistência, garantindo que o recurso esteja totalmente operacional e atenda aos critérios de estabilização de serviço definidos por cada AWS service (Serviço da AWS).
- Evento de conclusão da configuração: quando cada recurso tiver concluído a eventual fase de verificação de consistência do provisionamento, será definido um evento de Status detalhado de `CONFIGURATION_COMPLETE`.
- Evento de conclusão da criação de recursos: depois que o recurso for criado e configurado conforme especificado e a configuração corresponder ao especificado no modelo, o Status do evento `CREATE_COMPLETE` será definido.

É possível utilizar o evento `CONFIGURATION_COMPLETE` para agilizar o processo de criação de pilhas em cenários em que a verificação eventual da consistência dos recursos não é necessária, como a validação de uma configuração de pilha de pré-produção ou o provisionamento entre pilhas. Esse evento pode ser usado de várias maneiras. Por exemplo, você pode usá-lo como um sinal visual para evitar a espera pela conclusão da verificação de consistência do recurso ou da pilha. Ou você pode usá-lo para criar um mecanismo automatizado usando integração e entrega contínuas (CI/CD) para acionar ações adicionais.

⚠ Important

Embora a utilização do evento `CONFIGURATION_COMPLETE` acelere os tempos de criação da pilha, tenha em mente suas vantagens e desvantagens. Primeiro, ele só é compatível com um subconjunto de tipos de recursos que seja compatível com detecção de desvios. Para obter uma lista dos tipos de recursos que são compatíveis com detecção de desvios, consulte [Suporte a tipos de recursos](#). Essa abordagem pode não ser adequada para todos os cenários, especialmente quando os recursos exigem verificações completas de consistência para garantir a prontidão operacional total em todo o ambiente de nuvem (por exemplo, em ambientes de produção). Recomendamos avaliar cuidadosamente seus requisitos de implantação e a importância das verificações de consistência de cada recurso. Use o evento `CONFIGURATION_COMPLETE` para otimizar as velocidades de implantação sem comprometer a integridade e a confiabilidade da sua infraestrutura. Como não é garantido que o evento `CONFIGURATION_COMPLETE` seja definido, qualquer cenário que o use deverá estar preparado para lidar com um evento `CREATE_COMPLETE` quando nenhum evento `CONFIGURATION_COMPLETE` tiver sido definido.



Quando a implantação da pilha é iniciada, os recursos `AWS::ECR::Repository` e `AWS::ECS::Cluster` iniciam o processo de criação (`ResourceStatus = CREATE_IN_PROGRESS`). Quando o tipo de recurso `AWS::ECR::Repository` tiver iniciado a verificação de consistência eventual (`DetailedStatus = CONFIGURATION_COMPLETE`), o recurso `AWS::ECS::TaskDefinition` poderá iniciar o processo de criação. Da mesma forma, quando

o recurso `AWS::ECS::TaskDefinition` iniciar a verificação de consistência eventual, o recurso `AWS::ECS::Service` iniciará o processo de criação.

Eventos do **CREATE_IN_PROGRESS** e do **CREATE_COMPLETE**

- [Stack]: **CREATE_IN_PROGRESS**
- [Resource]: repositório do ECR **CREATE_IN_PROGRESS**
- [Resource]: cluster do ECS **CREATE_IN_PROGRESS**
- [Resource]: repositório do ECR **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource]: definição de tarefa do ECS **CREATE_IN_PROGRESS**
- [Resource]: cluster do ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource]: definição de tarefa do ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource]: serviço do ECS **CREATE_IN_PROGRESS**
- [Resource]: repositório do ECR **CREATE_COMPLETE**
- [Resource]: cluster do ECS **CREATE_COMPLETE**
- [Resource]: serviço do ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Stack]: **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource]: definição de tarefa do ECS **CREATE_COMPLETE**
- [Resource]: serviço do ECS **CREATE_COMPLETE**
- [Stack]: **CREATE_COMPLETE**

Monitorar o progresso da atualização de uma pilha

Monitore o progresso da atualização de uma pilha visualizando os eventos da pilha. A guia **Eventos** do console exibe cada etapa principal da criação e atualização da pilha classificada pelo tempo de cada evento, com os eventos mais recentes na parte superior. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Tópicos

- [Eventos gerados durante uma atualização de pilha bem-sucedida](#)
- [Eventos gerados quando uma atualização de recurso falha](#)

Eventos gerados durante uma atualização de pilha bem-sucedida

O início do processo de atualização da pilha é marcado com um evento `UPDATE_IN_PROGRESS` para a pilha:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

Em seguida estão os eventos que marcam o início e a conclusão da atualização de cada recurso que foi alterado no modelo de atualização. Por exemplo, a atualização de um recurso de [AWS::RDS::DBInstance](#) chamado MyDB resultaria nas seguintes entradas:

```
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

O evento `UPDATE_IN_PROGRESS` é registrado em log quando o CloudFormation relata que começou a atualizar o recurso. O evento `UPDATE_COMPLETE` é registrado em log quando o recurso é criado com êxito.

Quando o CloudFormation tiver atualizado com êxito a pilha, você verá o seguinte evento:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

Important

Durante as operações de atualização de pilha, se o CloudFormation precisar substituir um recurso existente, primeiramente ele criará um novo recurso e, em seguida, excluirá o antigo. No entanto, pode haver casos em que o CloudFormation não pode excluir o recurso antigo (por exemplo, se o usuário não tiver permissões para excluir um recurso de um determinado tipo).

O CloudFormation faz três tentativas de excluir o recurso antigo. Se o CloudFormation não puder excluir o recurso antigo, ele removerá o recurso antigo da pilha e continuará a atualizá-la. Quando a atualização da pilha estiver concluída, o CloudFormation emitirá um evento de pilha `UPDATE_COMPLETE`, mas incluirá um `StatusReason` que declarará que não foi possível excluir um ou mais recursos. O CloudFormation também emite um evento `DELETE_FAILED` para o recurso específico, com um `StatusReason` correspondente

fornecendo mais detalhes sobre o motivo pelo qual o CloudFormation não conseguiu excluir o recurso.

O recurso antigo ainda existe e continuará a gerar cobranças, mas deixará de ser acessível por meio do CloudFormation. Para excluir o recurso antigo, acesse o recurso antigo diretamente usando o console ou a API do serviço em questão.

Isso também se aplica aos recursos que você removeu do modelo de pilha e, portanto, eles serão excluídos da pilha durante a atualização da pilha.

Eventos gerados quando uma atualização de recurso falha

Se a atualização de um recurso falhar, o CloudFormation relatará um evento `UPDATE_FAILED` que incluirá um motivo para a falha. Por exemplo, se o modelo de atualização tiver especificado uma alteração de propriedade não aceita pelo recurso, como a redução do tamanho de `AllocatedStorage` para um recurso de [AWS::RDS::DBInstance](#), você verá eventos como estes:

```
2011-09-30 09:36 PDT AWS::RDS::DBInstance MyDB UPDATE_FAILED Size cannot be less than
current size; requested: 5; current: 10
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

Se um recurso de atualização falhar, o CloudFormation reverterá quaisquer recursos que tiverem sido atualizados para as configurações anteriores à atualização. Aqui está um exemplo de eventos que seriam vistos durante uma reversão de atualização:

```
2011-09-30 09:38 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_COMPLETE
2011-09-30 09:38 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:37 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
2011-09-30 09:37 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_IN_PROGRESS The
following resource(s) failed to update: [MyDB]
```

Continuar revertendo uma atualização

Às vezes, quando o CloudFormation tenta reverter uma atualização de pilha, ele não consegue reverter todas as alterações feitas durante o processo de atualização. Isso é chamado de estado `UPDATE_ROLLBACK_FAILED`. Por exemplo, é possível ter uma pilha que começa a reverter para uma instância antiga de banco de dados que foi excluída fora do CloudFormation. Como o

CloudFormation não sabe que o banco de dados foi excluído, ele pressupõe que a instância de banco de dados ainda existe e tenta reverter para ela, fazendo com que a atualização de reversão falhe.

Uma pilha no estado `UPDATE_ROLLBACK_FAILED` não pode ser atualizada, mas pode ser revertida para um estado funcional (`UPDATE_ROLLBACK_COMPLETE`). Após retornar a pilha às suas configurações originais, você pode tentar atualizá-la novamente.

Na maioria dos casos, é necessário corrigir o erro que faz com que a atualização de reversão falhe antes de continuar a reverter sua pilha. Em outros casos, é possível continuar a reverter a atualização sem qualquer alteração, por exemplo, quando uma operação de pilha expira.

 Note

Se você usa pilhas aninhadas, ao reverter a pilha pai, ela tentará reverter todas as pilhas filhos.

Para continuar revertendo uma atualização (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que a pilha está localizada.
3. Na página Pilhas, escolha a pilha que você deseja atualizar, escolha Ações da pilha e selecione Continuar reversão da atualização.

Se nenhuma das soluções no [Solucionar de problemas de erros](#) funcionou, é possível usar a opção avançada para ignorar os recursos que o CloudFormation não pode reverter com êxito. É necessário [examinar](#) e digitar os IDs lógicos dos recursos que você deseja ignorar. Especifique somente os recursos que entraram no estado `UPDATE_FAILED` durante o `UpdateRollback`, e não durante a atualização.

 Warning

O CloudFormation define o status dos recursos especificados como `UPDATE_COMPLETE` e continua a reverter a pilha. Após a reversão ser concluída, o estado dos recursos ignorados será inconsistente com o estado dos recursos no modelo de pilha. Antes de

executar outra atualização da pilha, é necessário atualizar a pilha ou os recursos para que sejam consistentes entre si. Caso contrário, as atualizações de pilha subsequentes podem falhar e a pilha se tornará irrecuperável.

Especifique o número mínimo de recursos exigidos para reverter sua pilha com êxito. Por exemplo, uma falha na atualização de recursos pode fazer com que os recursos dependentes falhem. Neste caso, não será necessário ignorar os recursos dependentes.

Para ignorar recursos que são parte de pilhas aninhadas, use o seguinte formato:

NestedStackName.ResourceLogicalID. Se você deseja especificar o ID lógico de uma pilha de recursos (Type: `AWS::CloudFormation::Stack`) na lista `ResourcesToSkip` então sua pilha incorporada correspondente deve estar em um dos seguintes estados: `DELETE_IN_PROGRESS`, `DELETE_COMPLETE` ou `DELETE_FAILED`.

Para continuar revertendo uma atualização (AWS CLI)

- Use o comando [continue-update-rollback](#) com a opção `--stack-name` para especificar o ID da pilha que você deseja continuar revertendo.

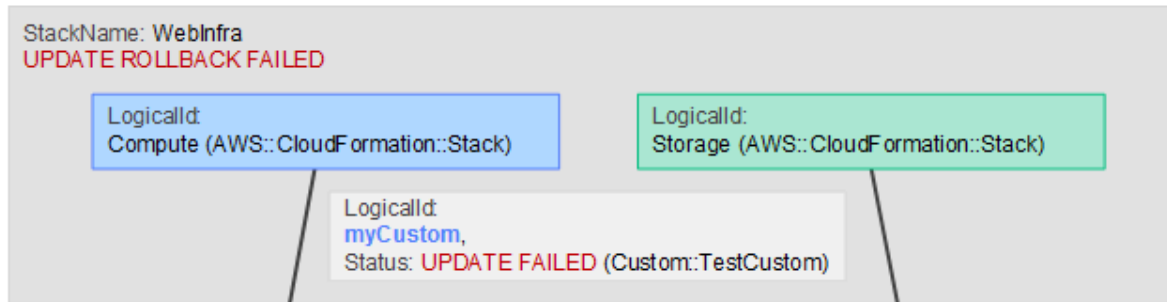
Continue revertendo de atualizações de pilhas aninhadas que falharam

Se você tiver várias pilhas aninhadas umas com as outras, talvez seja necessário pular recursos em vários níveis aninhados para que a hierarquia full-stack volte a funcionar.

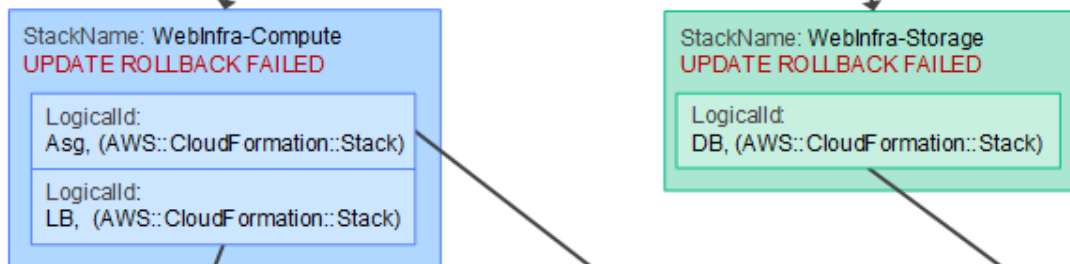
Por exemplo, você tem uma pilha raiz chamada `WebInfra` que contém duas pilhas menores dentro dela: `WebInfra-Compute` e `WebInfra-Storage`. Essas duas pilhas também têm suas próprias pilhas aninhadas dentro delas.

Se algo errado ocorrer durante uma atualização e o processo de atualização falhar, toda a hierarquia da pilha poderá acabar no estado `UPDATE_ROLLBACK_FAILED`, conforme mostrado no diagrama a seguir.

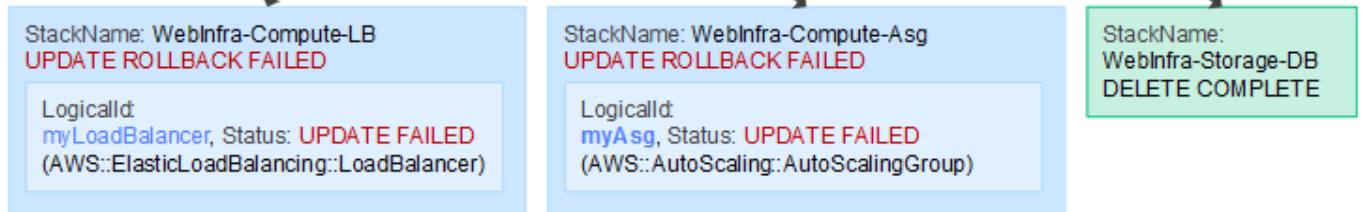
Root



Level 1



Level 2

**Note**

Os nomes das pilhas neste exemplo são truncados para simplicidade. Os nomes das pilhas filhos geralmente são gerados pelo CloudFormation e contêm strings aleatórias exclusivas, de forma que os nomes reais podem não ser acessíveis.

Para colocar a pilha raiz em um estado operável usando o comando `continue-update-rollback`, é necessário usar a opção `--resources-to-skip` para ignorar os recursos que falharam na reversão.

O exemplo `continue-update-rollback` a seguir retoma uma operação de reversão de uma atualização de pilha anterior em que houve falha. Neste exemplo, a opção `--resources-to-skip` inclui os seguintes itens:

- *myCustom*
- *WebInfra-Compute-Asg.myAsg*
- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

Para os recursos da pilha raiz, basta fornecer o ID lógico, por exemplo, *myCustom*. No entanto, para os recursos contidos em pilhas aninhadas, é necessário fornecer o nome da pilha aninhada e seu ID lógico separados por um ponto. Por exemplo, *WebInfra-Compute-Asg.myAsg*.

```
aws cloudformation continue-update-rollback --stack-name WebInfra \
  --resources-to-skip myCustom WebInfra-Compute-Asg.myAsg WebInfra-Compute-
  LB.myLoadBalancer WebInfra-Storage.DB
```

Para localizar o nome da pilha de uma pilha aninhada

É possível localizá-lo no ID da pilha ou no nome do recurso da Amazon (ARN) da pilha filha.

O exemplo de ARN a seguir refere-se a uma pilha chamada `WebInfra-Storage-Z2VKC706XKXT`.

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

Para localizar o ID lógico de uma pilha aninhada

É possível encontrar o ID lógico de uma pilha filho na definição do modelo de seu pai. No diagrama, o `LogicalId` da pilha filha `WebInfra-Storage-DB` é `DB` em sua pilha pai `WebInfra-Storage`.

No console do CloudFormation, também é possível encontrar o ID lógico na coluna ID lógico para o recurso de pilha nas guias Recursos ou Eventos. Para obter mais informações, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Determinar a causa de uma falha na pilha

Se a criação da pilha apresentar falhas, o CloudFormation poderá ajudar você a determinar o evento que, provavelmente, é a causa-raiz da falha da pilha. Dependendo do cenário e das suas

permissões, os eventos do AWS CloudTrail poderão fornecer mais detalhes sobre a causa-raiz, caso o Motivo do status fornecido em Eventos não esteja claro.

Para determinar a causa raiz de uma falha na pilha

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, selecione a pilha que apresentou a falha.
3. Escolha a guia Eventos.
4. Escolha Detectar causa-raiz. O CloudFormation analisará a falha e indicará o evento que é a provável causa da falha ao adicionar um rótulo de Causa-raiz provável ao evento específico Status. Consulte Motivo do status para obter mais explicações sobre o status no console do CloudFormation.
5. Escolha o Status de falha com o rótulo Causa-raiz provável para saber mais sobre a causa da falha. Dependendo do cenário e das suas permissões, será possível analisar um evento detalhado do CloudTrail. Confira a seguir os possíveis resultados da escolha do Status.
 - Os eventos do CloudTrail relacionados a esse problema estão disponíveis e podem ajudar na resolução. Visualize os eventos do CloudTrail.
 - Não foi possível encontrar nenhum evento do CloudTrail relacionado a esse problema que pudesse ajudar na resolução.
 - Suas permissões atuais não permitem o acesso para a visualização dos eventos do CloudTrail. Saiba mais.
 - No processo de verificação dos eventos do CloudTrail disponíveis, verifique novamente em alguns minutos.
 - Ocorreu um erro ao buscar os eventos do CloudTrail. Para realizar uma inspeção manual, acesse o console do CloudTrail.
6. Se o motivo fornecido em Motivo do status não estiver claro e a causa-raiz exibir um link para o console do CloudTrail, abra o link para visualizar o evento e encontrar uma causa-raiz detalhada.

Para obter mais informações sobre eventos do CloudTrail, consulte [Understanding CloudTrail events](#) e [CloudTrail record contents](#).

Para obter mais informações sobre o histórico de eventos do CloudTrail, consulte [Working with CloudTrail Event history](#).

 Note

As pilhas aninhadas não oferecem suporte para a opção Detectar causa-raiz.

Escolha como lidar com falhas ao provisionar recursos

Se sua operação de pilha falhar, você não precisará reverter recursos que já foram provisionados com sucesso e recomeçar do início todas as vezes. Em vez disso, você pode solucionar problemas de recursos em um status `CREATE_FAILED` ou `UPDATE_FAILED` e, em seguida, retomar o provisionamento a partir do ponto em que o problema ocorreu.

Para fazer isso, é necessário habilitar a opção preservar recursos provisionados com êxito. Essa opção está disponível para todas as implementações de pilha e operações de conjunto de alterações.

- Para criar uma pilha, se você escolher a opção Preservar recursos provisionados com sucesso, o CloudFormation preservará o estado dos recursos que foram criados com sucesso e deixará os recursos com falha até que a próxima operação de atualização seja realizada.
- Durante as operações de atualização e de conjunto de alterações, a opção Preservar recursos provisionados com êxito preserva o estado dos recursos bem-sucedidos e reverte os recursos com falha para seu último estado estável conhecido. Os recursos com falha estarão em um estado de `UPDATE_FAILED`. Os recursos sem um último estado estável conhecido serão excluídos na próxima operação de pilha.

Tópicos

- [Visão geral de opções de falha de pilhas](#)
- [Condições necessárias para pausar a reversão da pilha](#)
- [Preservar recursos provisionados com êxito \(console\)](#)
- [Preservar recursos provisionados com êxito \(AWS CLI\)](#)

Visão geral de opções de falha de pilhas

Antes de emitir uma operação usando o console, a API ou a AWS CLI do CloudFormation, especifique o comportamento caso ocorra falha do recurso provisionado. Em seguida, prossiga com o processo de implantação dos seus recursos sem outras modificações. No caso de uma falha

operacional, o CloudFormation interrompe na primeira falha em cada caminho de provisionamento independente. O CloudFormation identifica dependências entre recursos a fim de paralelizar ações de provisionamento independentes. Em seguida, continua a provisionar recursos em cada caminho de provisionamento independente até se deparar com uma falha. Uma falha em um caminho não afeta os demais caminhos de provisionamento. O CloudFormation continuará a provisionar os recursos até a conclusão ou até interromper em uma falha diferente.

Corrija todos os problemas para dar continuidade ao processo de implantação. O CloudFormation faz as atualizações necessárias antes de tentar novamente as ações de provisionamento em recursos que não puderam ser provisionados com êxito antes. Solucione problemas enviando operações Retry (Tentar novamente), Update (Atualizar) ou Roll back (Reverter). Por exemplo, se estiver provisionando uma instância do Amazon EC2 e ela falhar durante uma operação de criação, talvez você queira investigar o erro em vez de reverter o recurso com falha imediatamente. É possível revisar as verificações de status do sistema e as verificações de status de instâncias e, em seguida, selecionar a operação Retry (Tentar novamente) assim que os problemas forem resolvidos.

Quando uma operação de pilha falha e você especifica *Preserve successfully provisioned resources* (Preservar recursos provisionados com êxito) no menu *Stack failure options* (Opções de falha da pilha), é possível selecionar as seguintes opções.

- **Retry (Tentar novamente):** repete a operação de provisionamento em recursos com falha e continua provisionando o modelo até a conclusão bem-sucedida da operação da pilha ou até a próxima falha. Selecione essa opção se o provisionamento do recurso tiver falhado devido a um problema que não requer modificações no modelo, como uma permissão do AWS Identity and Access Management (IAM).
- **Update (Atualizar):** os recursos que foram provisionados são atualizados em atualizações de modelos. Os recursos que não tiverem sido criados ou atualizados serão repetidos. Selecione essa opção se o provisionamento do recurso tiver falhado devido a erros de modelo, e você tiver modificado o modelo. Ao atualizar uma pilha em estado **FAILED**, você deve selecionar *Preserve successfully provisioned resources* (Preservar recursos provisionados com êxito) em *Stack failure options* (Opções de falha da pilha) para continuar atualizando sua pilha.
- **Roll back (Reverter):** o CloudFormation reverte a pilha ao último estado estável conhecido.

Condições necessárias para pausar a reversão da pilha

Para impedir que o CloudFormation reverta e exclua automaticamente recursos que foram criados com êxito, as seguintes condições precisam ser atendidas.

1. Ao criar ou atualizar a pilha, você deve escolher a opção Preservar recursos provisionados com sucesso. Isso faz com que o CloudFormation não exclua os recursos que foram criados com sucesso, mesmo que a operação geral da pilha falhe.
2. A operação da pilha deve ter falhado, o que significa que o status da pilha é CREATE_FAILED ou UPDATE_FAILED.

 Note

Não há suporte para tipos de atualização imutáveis.

Preservar recursos provisionados com êxito (console)

Create stack

Para preservar recursos provisionados com êxito durante uma operação de criação de pilha

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas, escolha Criar pilha no canto superior direito e depois Com novos recursos (padrão).
3. Em Pré-requisito: preparar modelo, escolha Escolher um modelo existente.
4. Em Specify template (Especificar modelo), escolha especificar o URL para o bucket do S3 que contém o modelo de pilha ou fazer upload de um arquivo de modelo de pilha. Em seguida, escolha Próximo.
5. Na página Specify stack details (Especificar detalhes da pilha), insira o nome de uma pilha na caixa Stack name (Nome da pilha).
6. Na seção Parameters (Parâmetros), especifique os parâmetros que são definidos no seu modelo da pilha.

Você pode usar ou alterar todos os parâmetros com valores padrão.

7. Quando estiver satisfeito com os valores dos parâmetros, escolha Next (Próximo).
8. Na página Configure stack options (Configurar opções de pilha), é possível definir opções adicionais para a pilha.
9. Em Stack failure options (Opções de falha da pilha), selecione Preserve successfully provisioned resources (Preservar recursos provisionados com êxito).

10. Quando estiver satisfeito com as opções da pilha, escolha Next (Próximo).
11. Revise sua pilha na página Review (Revisar) e selecione Create stack (Criar pilha).

Resultados: os recursos com falha na criação farão com que o status da pilha mude para CREATE_FAILED, a fim de evitar que esta seja revertida quando a operação de pilha se deparar com uma falha. Os recursos provisionados com êxito estão em um estado CREATE_COMPLETE. É possível monitorar a pilha na guia Stack events (Eventos da pilha).

Update stack

Para preservar recursos provisionados com êxito durante uma operação de atualização de pilha

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Selecione a pilha que você deseja atualizar e escolha Update (Atualizar).
3. Na página Update stack (Atualizar pilha), escolha um modelo de pilha usando uma das seguintes opções:
 - Usar modelo existente
 - Replace current template (Substituir modelo atual)
 - Editar modelo no Infrastructure Composer

Aceite suas configurações e selecione Next (Próximo).

4. Na página Specify stack details (Especificar detalhes da pilha), especifique os parâmetros que estão definidos no modelo da pilha.

Você pode usar ou alterar todos os parâmetros com valores padrão.

5. Quando estiver satisfeito com os valores dos parâmetros, escolha Next (Próximo).
6. Na página Configure stack options (Configurar opções de pilha), é possível definir opções adicionais para a pilha.
7. Para Behavior on provisioning failure (Comportamento na falha de provisionamento), selecione Preserve successfully provisioned resources (Preservar recursos provisionados com êxito).
8. Quando estiver satisfeito com as opções da pilha, escolha Next (Próximo).
9. Revise sua pilha na página Review (Revisar) e selecione Update stack (Atualizar pilha).

Resultados: os recursos com falha na atualização farão com que o status da pilha mude para `UPDATE_FAILED` e reverterá a pilha para o último estado estável conhecido. Os recursos sem um último estado estável conhecido serão excluídos pelo CloudFormation na próxima operação de pilha. Os recursos provisionados com êxito estão em um estado `CREATE_COMPLETE` ou `UPDATE_COMPLETE`. É possível monitorar a pilha na guia Stack events (Eventos da pilha).

Change set

Note

É possível iniciar um conjunto de alterações para uma pilha com um status de `CREATE_FAILED` ou `UPDATE_FAILED`, mas não para um status de `UPDATE_ROLLBACK_FAILED`.

Para preservar recursos provisionados com êxito durante uma operação de conjunto de alterações

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Selecione a pilha que contém o conjunto de alterações que você deseja iniciar e depois escolha a guia Change sets (Conjuntos de alterações).
3. Selecione o conjunto de alterações e escolha Execute (Executar).
4. Para Execute change set (Executar conjunto de alterações), selecione a opção Preserve successfully provisioned resources (Preservar recursos provisionados com êxito).
5. Selecione Execute change set (Executar conjunto de alterações).

Resultados: os recursos com falha na atualização farão com que o status da pilha mude para `UPDATE_FAILED` e reverterá a pilha para o último estado estável conhecido. Os recursos sem um último estado estável conhecido serão excluídos pelo CloudFormation na próxima operação de pilha. Os recursos provisionados com êxito estão em um estado `CREATE_COMPLETE` ou `UPDATE_COMPLETE`. É possível monitorar a pilha na guia Stack events (Eventos da pilha).

Preservar recursos provisionados com êxito (AWS CLI)

Create stack

Para preservar recursos provisionados com êxito durante uma operação de criação de pilha

Especifique a opção `--disable-rollback` ou a enumeração `on-failure DO_NOTHING` durante uma operação [create-stack](#).

1. Forneça um nome de pilha e um modelo para o comando `create-stack` com a opção `--disable-rollback`.

```
aws cloudformation create-stack --stack-name myteststack \  
  --template-body file://template.yaml \  
  --disable-rollback
```

O comando retorna a seguinte saída.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. Descreva o estado da pilha usando o comando `describe-stacks`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

O comando retorna a seguinte saída.

```
{  
  "Stacks": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",  
      "Description": "AWS CloudFormation Sample Template",  
      "Tags": [],  
      "Outputs": [],  
      "StackStatusReason": "The following resource(s) failed to create:  
[MyBucket]",  
      "CreationTime": "2013-08-23T01:02:15.422Z",  
      "Capabilities": [],
```

```
        "StackName": "myteststack",
        "StackStatus": "CREATE_FAILED",
        "DisableRollback": true
    }
]
```

Update stack

Para preservar com êxito os recursos provisionados durante uma operação de atualização de pilha

1. Forneça um nome de pilha existente e um modelo para o comando `update-stack` com a opção `--disable-rollback`.

```
aws cloudformation update-stack --stack-name myteststack \
    --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template
    --disable-rollback
```

O comando retorna a seguinte saída.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Descreva o estado da pilha usando o comando `describe-stacks` ou `describe-stack-events`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

O comando retorna a seguinte saída.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
    }
  ]
}
```

```

        "CreationTime": "2013-08-23T01:02:15.422Z",
        "Capabilities": [],
        "StackName": "myteststack",
        "StackStatus": "UPDATE_COMPLETE",
        "DisableRollback": true
    }
]
}
```

Change set

Note

É possível iniciar um conjunto de alterações de uma pilha com um status de `CREATE_FAILED` ou `UPDATE_FAILED`, mas não para um status de `UPDATE_ROLLBACK_FAILED`.

Para preservar recursos provisionados com êxito durante uma operação de conjunto de alterações

Especifique a opção `--disable-rollback` durante uma operação [execute-change-set](#).

1. Forneça um nome de pilha e um modelo para o comando `execute-change-set` com a opção `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --template-body file://template.yaml
```

O comando retorna a seguinte saída.

```
{
  "Id": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Inicie o conjunto de alterações com a opção `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \  
  --change-set-name my-change-set --disable-rollback
```

3. Determine o status da pilha usando o comando `describe-stacks` ou `describe-stack-events`.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

O comando retorna a seguinte saída.

```
{  
  "StackEvents": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",  
      "EventId": "49c966a0-7b74-11ea-8071-024244bb0672",  
      "StackName": "myteststack",  
      "LogicalResourceId": " MyBucket",  
      "PhysicalResourceId": "myteststack-MyBucket-abcdefghijkl",  
      "ResourceType": "AWS::S3::Bucket",  
      "Timestamp": "2020-04-10T21:43:17.015Z",  
      "ResourceStatus": "UPDATE_FAILED"  
      "ResourceStatusReason": "User XYZ is not allowed to perform  
S3::UpdateBucket on MyBucket"  
    }  
  ]  
}
```

4. Corrija os erros de permissão e repita a operação.

```
aws cloudformation update-stack --stack-name myteststack \  
  --use-previous-template --disable-rollback
```

O comando retorna a seguinte saída.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

5. Descreva o estado da pilha usando o comando `describe-stacks` ou `describe-stack-events`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

O comando retorna a seguinte saída.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "UPDATE_COMPLETE",
      "DisableRollback": true
    }
  ]
}
```

Reverter uma pilha

É possível usar o comando [rollback-stack](#) para reverter uma pilha com um status de pilha CREATE_FAILED ou UPDATE_FAILED ao seu último estado estável.

O comando rollback-stack a seguir reverte a pilha especificada.

```
aws cloudformation rollback-stack --stack-name myteststack
```

O comando retorna a seguinte saída.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

 Note

A operação `rollback-stack` excluirá uma pilha se ela não contiver um último estado estável conhecido.

Reverta sua pilha do CloudFormation em caso de violação de alarme com acionadores de reversão

Com os acionadores de reversão, você pode fazer com que o CloudFormation monitore o estado da sua aplicação durante a criação e a atualização da pilha e reverter essa operação se a aplicação ultrapassar o limite dos alarmes que você especificou. Para cada trigger de reversão criado, você especifica o alarme do CloudWatch que o CloudFormation deve monitorar. O CloudFormation monitora os alarmes especificados durante a operação de criação ou atualização da pilha, e pelo tempo especificado depois que todos os recursos foram implantados. Se qualquer um dos alarmes entrar no estado `ALARM` durante a operação de pilha ou o período de monitoramento, o CloudFormation reverterá toda a operação de pilha.

Você pode definir um tempo de monitoramento de 0 (padrão) até 180 minutos. Durante esse tempo, o CloudFormation monitora todos os triggers de reversão depois da implantação de todos os recursos necessários pela operação de criação ou atualização da pilha. Se qualquer um dos alarmes entrar no estado `ALARM` durante a operação de pilha ou esse período de monitoramento, o CloudFormation reverterá toda a operação de pilha. Em seguida, para as operações de atualização, se o período de monitoramento expirar sem que qualquer alarme entre no estado `ALARM`, o CloudFormation continuará a descartar recursos antigos normalmente. Se você definir um tempo de monitoramento, mas não especificar um trigger de reversão, o CloudFormation ainda aguardará o tempo especificado antes de limpar os recursos antigos para atualizar as operações. Você pode usar este período de monitoramento para executar qualquer validação manual desejada na pilha, e cancelar ou atualizar a criação da pilha manualmente conforme necessário. Se você definir um tempo de monitoramento de 0 minutos, o CloudFormation ainda fará o monitoramento dos triggers de reversão durante as operações de criação e atualização da pilha e reverterá a operação se um alarme entrar no estado `ALARM`. Em seguida, para as operações de atualização sem alarmes de violação, ele começará a eliminar os recursos antigos imediatamente assim que a operação for concluída.

Por padrão, o CloudFormation somente reverte as operações da pilha se um alarme entrar no estado `ALARM`, não no estado `INSUFFICIENT_DATA`. Para que o CloudFormation também reverta

a operação de pilha caso um alarme entre no estado `INSUFFICIENT_DATA`, edite o alerta do CloudWatch para tratar a ausência de dados como `breaching`. Para obter mais informações, consulte [Configuração da forma como os alarmes do CloudWatch tratam dados ausentes](#) no Manual do usuário do Amazon CloudWatch.

O CloudFormation não monitora os triggers de reversão quando ele reverte uma pilha durante uma operação de atualização.

Você pode adicionar um máximo de cinco triggers de reversão. Para adicionar um trigger de reversão, especifique o nome do recurso da Amazon (ARN) do alarme do CloudWatch. No momento, `AWS::CloudWatch::Alarm` e `AWS::CloudWatch::CompositeAlarm` podem ser utilizados como gatilhos de reversão. Para obter mais informações sobre os alarmes do CloudWatch, consulte [Uso dos alarmes do CloudWatch](#), no Guia do usuário do Amazon CloudWatch.

Se um determinado alarme do CloudWatch estiver ausente, toda a operação de pilha falhará e será revertida.

Lembre-se de que o acesso ao CloudWatch requer credenciais. Essas credenciais devem ter permissões para acessar os recursos da AWS, como recuperar dados de métricas do CloudWatch sobre seus recursos. Para obter mais informações, consulte [Autenticação e controle de acesso para o CloudWatch](#), no Guia do usuário do Amazon CloudWatch.

Adicionar triggers de reversão durante a criação ou atualização da pilha

Para adicionar gatilhos de reversão durante a criação ou a atualização de pilhas (console)

1. Durante a criação ou a atualização de uma pilha, na página Configurar opções de pilha, em Opções avançadas, expanda a seção Configuração de reversão.
2. Insira um tempo de monitoramento entre 0 e 180 minutos. O valor padrão é 0.
3. Especifique o ARN do alarme ou alarme composto do CloudWatch que você deseja usar como gatilho de reversão e escolha Adicionar ARN de alarme do CloudWatch.

Por exemplo, o seguinte é um ARN para um alarme do CloudWatch ou um alarme composto, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Escolha Avançar e revise os detalhes da pilha.
5. Quando estiver pronto, escolha Enviar para criar ou atualizar a pilha.

Para adicionar gatilhos de reversão durante a criação ou atualização de pilhas (AWS CLI)

Use o comando [create-stack](#) ou [update-stack](#) com a opção `--rollback-configuration`.

Por exemplo, o seguinte comando `update-stack` define *MyCompositeAlarm* como um gatilho de reversão com um período de monitoramento de 5 minutos:

```
aws cloudformation update-stack --stack-name MyStack \  
  --use-previous-template \  
  --rollback-configuration \  
  "RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-  
east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTim
```

Adicionar gatilhos de reversão a um conjunto de alterações

Para adicionar gatilhos de reversão a um conjunto de alterações (console)

1. Durante a criação ou a atualização de um conjunto de alterações, na página Configurar opções de pilha, em Opções avançadas, expanda a seção Configuração de reversão.
2. Insira um tempo de monitoramento entre 0 e 180 minutos. O valor padrão é 5.
3. Especifique o ARN do alarme ou alarme composto do CloudWatch que você deseja usar como gatilho de reversão e escolha Adicionar ARN de alarme do CloudWatch.

Por exemplo, o seguinte é um ARN para um alarme do CloudWatch ou um alarme composto, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Escolha Avançar e revise os detalhes do conjunto de alterações.
5. Quando estiver pronto, escolha Criar conjunto de alterações para criar o conjunto de alterações.

Para adicionar gatilhos de reversão a um conjunto de alterações (AWS CLI)

Use o comando [create-change-set](#) com a opção `--rollback-configuration`.

Visualizar gatilhos de reversão para uma pilha

Para visualizar gatilhos de reversão para uma pilha, consulte a seção Rollback configuration (Configuração de reversão).

1. Na página Pilhas, escolha, na lista à esquerda, a pilha que você deseja visualizar.
2. Na guia Informações da pilha, expanda a seção Configuração de reversão para visualizar os gatilhos de reversão.

Visualizar gatilhos de reversão de um conjunto de alterações

Para visualizar gatilhos de reversão para um conjunto de alterações, consulte a seção Rollback configuration (Configuração de reversão).

1. Na página Pilhas, escolha, na lista à esquerda, a pilha que você deseja visualizar.
2. Selecione a guia Conjuntos de alterações e depois escolha o conjunto de alterações que você deseja visualizar.
3. Escolha a guia Entrada e visualize a seção Configuração de reversão.

Detectar alterações de configuração não gerenciadas em pilhas e recursos com detecção de desvios

Mesmo que você gerencie seus recursos com o CloudFormation, os usuários podem alterá-los fora do CloudFormation. Os usuários podem editar recursos diretamente usando o serviço subjacente que os criou. Por exemplo, você pode usar o console do Amazon EC2 para atualizar uma instância de servidor que foi criada como parte de uma pilha do CloudFormation. Algumas mudanças podem ser acidentais e algumas podem ser feitas intencionalmente para responder a eventos operacionais sensíveis ao tempo. Independentemente disso, as alterações feitas fora do CloudFormation podem complicar as operações de atualização ou exclusão de pilha. Você pode usar a detecção de desvio para identificar recursos de pilha para os quais alterações de configuração foram feitas fora do gerenciamento do CloudFormation. Depois, é possível tomar medidas corretivas para que seus recursos de pilha estejam novamente em sincronia com suas definições no modelo de pilha, como atualizar os recursos desviados diretamente para que eles correspondam à definição do modelo. Resolver o desvio ajuda a garantir a consistência da configuração e operações de pilha bem-sucedidas.

Tópicos

- [O que é desvio?](#)
- [Códigos de status de detecção de desvio](#)
- [Considerações ao detectar o desvio](#)
- [Detectar desvio em uma pilha inteira do CloudFormation](#)
- [Detectar desvios em recursos de pilha individuais](#)
- [Resolver o desvio com uma operação de importação](#)

O que é desvio?

A detecção de desvio permite detectar se a configuração real de uma pilha difere ou se desviou da configuração esperada. Use o CloudFormation para detectar desvios em uma pilha inteira ou em recursos individuais dentro da pilha. Considera-se que um recurso foi desviado se algum dos seus valores de propriedade reais for diferente dos valores de propriedade esperados. Isso inclui se a propriedade ou o recurso foi excluído. Considera-se que uma pilha foi desviada se um ou mais de seus recursos foram desviados.

Para determinar se um recurso foi desviado, o CloudFormation determina os valores de propriedades de recursos esperados, conforme definido no modelo de pilha, e quaisquer valores especificados como parâmetros de modelo. Em seguida, o CloudFormation compara esses valores esperados com os valores reais dessas propriedades de recursos existentes atualmente na pilha. Considera-se que um recurso foi desviado quando uma ou mais de suas propriedades foi excluídas ou teve seu valor alterado.

O CloudFormation gera informações detalhadas sobre cada recurso na pilha que se desviou.

O CloudFormation detecta desvios nos recursos da AWS que suportam a detecção de desvios. Recursos sem suporte para a detecção de desvio recebem um status de desvio de NOT_CHECKED. Para obter uma lista de recursos da AWS que oferecem suporte à detecção de desvios, consulte [Suporte a tipos de recursos](#).

Além disso, o CloudFormation é compatível com a detecção de desvios em tipos de recursos privados provisionáveis, ou seja, com tipo de provisionamento FULLY_MUTABLE ou IMMUTABLE. Para realizar a detecção de desvios em um recurso de tipo privado, a versão padrão do tipo de recurso que você registrou em sua conta deve ser provisionável. Para obter mais informações sobre o tipo de provisão de recursos, consulte o parâmetro ProvisioningType da ação [DescribeType](#) na Referência de APIs do AWS CloudFormation e o comando [DescribeType](#) na Referência de comandos da AWS CLI. Para obter mais informações sobre recursos privados, consulte [Gerenciar extensões com o registro do CloudFormation](#).

Você pode realizar a detecção de desvio em pilhas com os seguintes status: CREATE_COMPLETE, UPDATE_COMPLETE, UPDATE_ROLLBACK_COMPLETE e UPDATE_ROLLBACK_FAILED.

Ao detectar o desvio em uma pilha, o CloudFormation não detecta desvios em nenhuma pilha aninhada que pertença a essa pilha. Para obter mais informações, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#). Em vez disso, você pode iniciar uma operação de detecção de desvio diretamente na pilha aninhada.

 Note

O CloudFormation determina apenas o desvio para valores de propriedade explicitamente definidos, seja por meio do modelo de pilha ou especificando parâmetros de modelo. Isso não inclui valores padrão para propriedades de recursos. Para que o CloudFormation controle uma propriedade de recurso para fins de determinação de desvio, defina explicitamente o valor da propriedade, mesmo que você a esteja definindo com o valor padrão.

Códigos de status de detecção de desvio

As tabelas nesta seção descrevem os vários tipos de status usados com a detecção de desvio:

- Status da operação de detecção de desvio descreve o estado atual da operação de desvio.
- Status do desvio

Em conjuntos de pilhas, isso descreve o status do desvio do conjunto de pilhas como um todo, com base no status do desvio das instâncias de pilha que pertencem a ele.

Em instâncias de pilha, isso descreve o status do desvio da instância de pilha, com base no status do desvio da pilha associada.

Em pilhas, isso descreve o status do desvio da pilha como um todo, com base no status do desvio dos recursos.

- Status de desvio de recursos descreve o status de desvio de um recurso individual.

A tabela a seguir lista os códigos de status que o CloudFormation atribui a operações de detecção de desvio de pilha.

Status da operação de detecção de desvio	Descrição
DETECTION_COMPLETE	A operação de detecção de desvio de pilha foi concluída com êxito para todos os recursos na pilha compatível com a detecção de desvio.

Status da operação de detecção de desvio	Descrição
DETECTION_FAILED	A operação de detecção de desvio de pilha falhou para pelo menos um recurso na pilha. Os resultados estarão disponíveis para recursos nos quais o CloudFormation concluiu com êxito a detecção de desvio.
DETECTION_IN_PROGRESS	A operação de detecção de desvio de pilha está em andamento.

A tabela a seguir lista os códigos de status de desvio do CloudFormation atribuídos a pilhas.

Status do desvio	Descrição
DRIFTED	Em pilhas: a pilha é diferente, ou foi desviada de sua configuração de modelo esperada. Considera-se que uma pilha foi desviada se um ou mais de seus recursos foram desviados. Em instâncias de pilha: uma instância de pilha é considerada com desvio se a pilha associada a ela tiver desviado. Em conjuntos de pilhas: um conjunto de pilhas é considerado com desvio se uma ou mais instâncias de pilha tiverem desviado.
NOT_CHECKED	O CloudFormation não verificou se a pilha, o conjunto de pilha ou a instância de pilha difere da configuração de modelo esperada.
IN_SYNC	A configuração atual de cada recurso compatível corresponde à sua configuração de modelo esperada. Uma pilha, um conjunto de pilha ou uma instância de pilha sem recursos que ofereçam suporte à detecção de desvio também terá um status de IN_SYNC.

A tabela a seguir lista os códigos de status de desvio que o CloudFormation atribui a recursos de pilha.

Status de desvio do recurso	Descrição
DELETED	O recurso difere de sua configuração de modelo esperada porque o recurso foi excluído.
MODIFIED	O recurso difere de sua configuração de modelo esperada.
NOT_CHECKED	O CloudFormation não verificou se o recurso é diferente da configuração de modelo esperada.
IN_SYNC	A configuração atual do recurso corresponde à configuração do modelo esperada.

A tabela a seguir lista os códigos de status de tipos de diferença que o CloudFormation atribui às propriedades do recurso que diferem de configurações de modelo esperadas.

Tipos de diferenças de propriedades	Descrição
ADD	Um valor foi adicionado a uma propriedade de recurso que é um tipo de dados de matriz ou lista.
REMOVE	A propriedade foi removida da configuração atual do recurso.
NOT_EQUAL	O valor da propriedade atual difere de seu valor esperado, conforme definido no modelo de pilha.

Considerações ao detectar o desvio

Para realizar com êxito a detecção de desvio em uma pilha, um usuário deve ter as seguintes permissões:

- Permissão de leitura para cada recurso compatível com a detecção de desvio incluída na pilha. Por exemplo, se a pilha inclui um recurso `AWS::EC2::Instance`, você deverá ter a permissão `ec2:DescribeInstances` para realizar a detecção de desvio na pilha.
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`
- `cloudformation:BatchDescribeTypeConfigurations`

Para obter mais informações sobre a configuração de permissões no CloudFormation, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Em certos casos de ponto, o CloudFormation pode não ser capaz de sempre retornar resultados precisos de desvio. Você deve estar ciente desses casos de ponto para interpretar corretamente seus resultados de detecção de desvio.

- Em certos casos, os objetos contidos em matrizes de propriedades serão relatados como desvios, quando, na verdade, são valores padrão fornecidos à propriedade a partir do serviço subjacente responsável pelo recurso.
- Certos recursos têm relacionamentos de anexo com recursos relacionados, de modo que um recurso pode realmente anexar ou remover valores de propriedade de outro recurso, definidos no mesmo modelo ou em outro. Por exemplo, os recursos `AWS::EC2::SecurityGroupIngress` e `AWS::EC2::SecurityGroupEgress` podem ser usados para prender e remover valores de recursos `AWS::EC2::SecurityGroup`. Nesses casos, o CloudFormation analisa o modelo de pilha de anexos antes de realizar a comparação de desvio. No entanto, o CloudFormation não pode executar essa análise entre as pilhas e, portanto, pode não retornar resultados precisos de desvio quando os recursos anexados residem em pilhas diferentes.

Recursos compatível com a detecção de desvio e que permitem ou exigem anexos de outros recursos incluem:

Tipo de recurso	Tipo de recurso de anexo
<code>AWS::SNS::Topic</code>	<code>AWS::SNS::Subscription</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::UserToGroupAddition</code>
<code>AWS::IAM::Group</code>	<code>AWS::IAM::ManagedPolicy</code>

Tipo de recurso	Tipo de recurso de anexo
AWS::IAM::Role	AWS::IAM::Policy
AWS::IAM::User	AWS::IAM::RolePolicy
AWS::ElasticLoadBalancingV2::Listener	AWS::ElasticLoadBalancingV2::ListenerCertificate
AWS::EC2::SecurityGroup	AWS::EC2::SecurityGroupEgress AWS::EC2::SecurityGroupIngress

- O CloudFormation não realiza a detecção de desvio na propriedade `KMSKeyId` de recursos. Como chaves do AWS KMS podem ser referenciadas por vários aliases, o CloudFormation não pode garantir resultados de desvio consistentemente precisos para essa propriedade.
- Há certas propriedades de recurso que você pode especificar em seu modelo de pilha que, pela própria natureza, o CloudFormation não será capaz de comparar com as propriedades nos recursos de pilha resultantes. Portanto, essas propriedades não podem ser incluídas nos resultados da detecção de desvio. Essas propriedades se dividem em duas grandes categorias:
 - Valores de propriedade que o CloudFormation não pode mapear de volta para o valor de propriedade de recurso inicial no modelo de pilha.

Por exemplo, o CloudFormation não pode mapear o código-fonte de uma função do Lambda de volta ao tipo de propriedade `Code` do recurso `AWS::Lambda::Function` e, portanto, o CloudFormation não pode incluí-lo em resultados de detecção de desvio.

- Os valores de propriedades que o serviço responsável pelo recurso não retorna.

Há determinados valores de propriedades que, por natureza, nunca são retornados pelo serviço ao qual o recurso pertence. Eles tendem a conter informações confidenciais, como senhas ou outros dados confidenciais que não devem ser expostos. Por exemplo, o serviço do IAM nunca retornará o valor da propriedade `Password` do tipo de propriedade `AWS::IAM::UserLoginProfile`, e, portanto, o CloudFormation não poderá incluí-lo em resultados de detecção de desvio.

- Objetos em uma matriz podem ser, na verdade, padrões de serviço, e não desvios adicionados manualmente.

- Se você encontrar algum falso positivo, envie seus comentários usando o link de feedback no console do CloudFormation ou entre em contato conosco em [AWS re:Post](#).
- Algumas propriedades podem ter valores de entrada equivalentes, mas não idênticos. Para evitar falsos positivos, você deve garantir que a configuração esperada corresponda à configuração real.
- Por exemplo, a configuração esperada da propriedade do recurso pode ser 1024 MB e a configuração real da mesma propriedade de recurso pode ser 1 GB. 1024 MB e 1 GB são equivalentes, mas não idênticos.

Quando a detecção de desvio é executada nessa propriedade de recurso, ela sinaliza resultados com desvios.

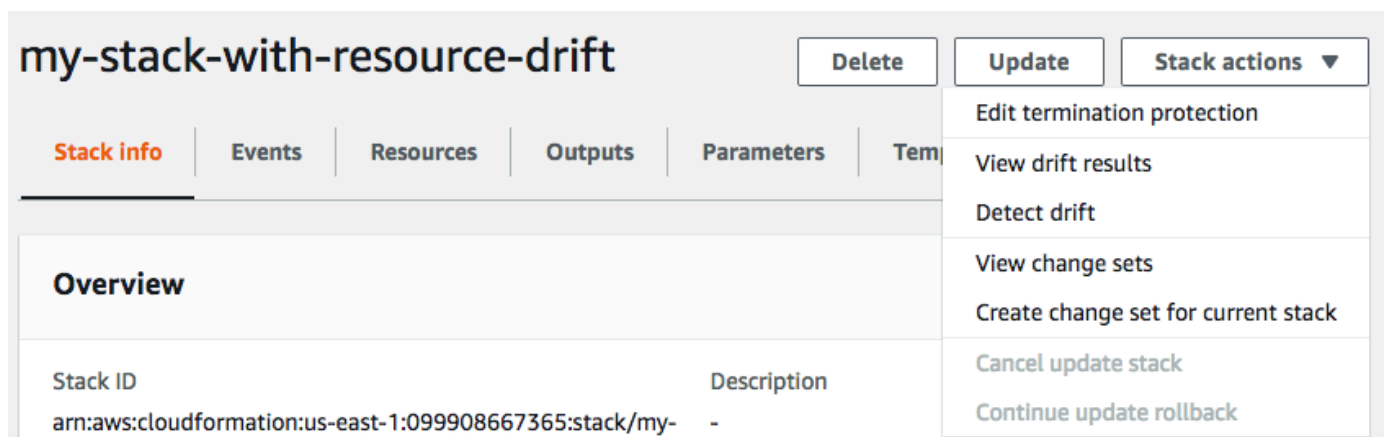
Para evitar esse falso positivo, altere a configuração esperada da propriedade do recurso para 1024 MB e depois execute a detecção de desvio.

Detectar desvio em uma pilha inteira do CloudFormation

A realização de uma operação de detecção de desvio em uma pilha determina se essa pilha se desviou de sua configuração de modelo esperada e retorna informações detalhadas sobre o status de desvio de cada recurso na pilha que oferece suporte para a detecção de desvio.

Para detectar o desvio em uma pilha inteira usando o Console de gerenciamento da AWS

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na lista de pilhas, selecione a pilha em que deseja executar a detecção de desvio. No painel de detalhes da pilha, escolha Stack actions (Ações da pilha) e selecione Detect drift (Detectar desvio).



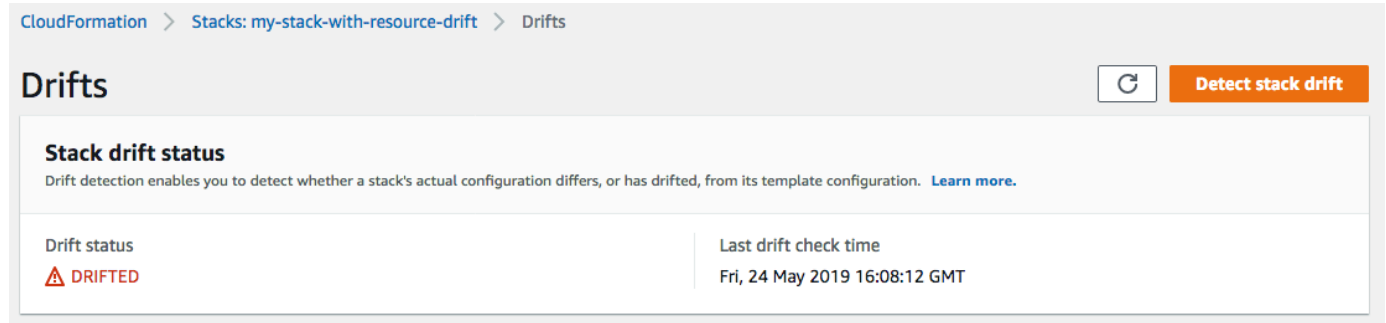
O CloudFormation exibe uma barra de informações que declara que a detecção de desvio foi iniciada para a pilha selecionada.

3. Aguarde até que o CloudFormation conclua a operação de detecção de desvios. Quando a operação de detecção de desvio for concluída, o CloudFormation atualizará o Status de desvio e o Tempo da última verificação de desvio para sua pilha. Esses campos estão listados na seção Overview (Visão geral) do painel Stack info (Informações da pilha) da página de detalhes da pilha.

A operação de detecção de desvio pode levar vários minutos, dependendo do número de recursos incluídos na pilha. Você só pode executar uma operação de detecção de desvio em uma determinada pilha ao mesmo tempo. O CloudFormation continua a operação de detecção de desvios mesmo depois de você descartar a barra de informações.

4. Revise os resultados da detecção de desvio para a pilha e seus recursos. Com a pilha selecionada, no menu Stack actions (Ações da pilha) selecione View drift results (Visualizar os resultados de desvio).

O CloudFormation lista o status geral de desvio da pilha, bem como a última vez que a detecção de desvio foi iniciada na pilha ou em qualquer um dos recursos individuais dela. Considera-se que uma pilha foi desviada se um ou mais de seus recursos foram desviados.



Na seção Status de desvio do recurso, o CloudFormation lista cada recurso de pilha, o status de desvio dele e a última vez em que a detecção de desvios foi iniciada no recurso. O ID lógico e o ID físico de cada recurso são exibidos para ajudar você a identificá-los. Além disso, para recursos com status MODIFIED, o CloudFormation exibe detalhes de desvios do recurso.

É possível classificar os recursos com base em seus status de desvio usando a coluna Drift status (Status de desvio).

- Para visualizar os detalhes em um recurso modificado.

- Com o recurso modificado selecionado, escolha View drift details (Visualizar detalhes de desvio).

O CloudFormation exibe a página de detalhes de desvios desse recurso. Esta página lista os valores de propriedade esperados e atuais do recurso e quaisquer diferenças entre os dois.

Para destacar uma diferença, na seção Differences (Diferenças), selecione o nome da propriedade.

- As propriedades adicionadas são destacadas em verde na coluna Current (Atuais) da seção Details (Detalhes).
- As propriedades excluídas são destacadas em vermelho na coluna Expected (Esperadas) da seção Details (Detalhes).
- As propriedades cujo valor foi alterado são destacadas em amarelo nas colunas Expected (Esperadas) e Current (Atuais).

Differences (4)

<1>

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName":"stuff","AttributeType":"S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key":"test","Value":"test"}]	-

Details

Expected

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Para detectar o desvio em uma pilha inteira usando o AWS CLI

 Important

Consulte o Horário da última verificação de desvio da pilha e confirme se é anterior ao carimbo de data/hora mostrado nos resultados do desvio do recurso para evitar o uso de dados desatualizados.

Para detectar desvios em uma pilha inteira usando a AWS CLI, use os seguintes comandos AWS CLI:

- `detect-stack-drift` para iniciar uma operação de detecção de desvio em uma pilha.
 - `describe-stack-drift-detection-status` para monitorar o status da operação de detecção de desvio da pilha.
 - `describe-stack-resource-drifts` para rever os detalhes da operação de detecção de desvio da pilha.
1. Use o `detect-stack-drift` para detectar o desvio em uma pilha inteira. Especifique o nome da pilha ou o ARN. Você também pode especificar os IDs lógicos de quaisquer recursos específicos que você deseja usar como filtros para essa operação de detecção de desvio.

```
aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
```

Resultado:

```
{
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
}
```

2. Como as operações de detecção de desvio de pilha podem ser de longa execução, use `describe-stack-drift-detection-status` para monitorar o status da operação de desvio. Esse comando usa o ID de detecção de desvio da pilha retornado pelo comando `detect-stack-drift`.

No exemplo abaixo, pegamos o ID de detecção de desvio da pilha retornado pelo exemplo acima de `detect-stack-drift` e o transmitimos como um parâmetro para `describe-stack-drift-detection-status`. O parâmetro retorna detalhes da operação que mostram que a operação de detecção de desvio foi concluída, que um recurso de pilha única foi desviado e que a pilha inteira é considerada como tendo sido desviada como resultado.

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

Resultado:

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",

```

```

    "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample",
    "StackDriftStatus": "DRIFTED",
    "Timestamp": "2018-03-26T17:23:22.279Z",
    "DetectionStatus": "DETECTION_COMPLETE",
    "DriftedStackResourceCount": 1
  }

```

- Quando a operação de detecção de desvio de pilha estiver concluída, use o comando `describe-stack-resource-drifts` para rever os resultados, incluindo valores de propriedade reais e esperados para os recursos que se desviaram.

O exemplo abaixo usa a opção `--stack-resource-drift-status-filters` para solicitar informações de desvio de pilha para os recursos que foram modificados ou excluídos. A solicitação retorna informações sobre o único recurso que foi modificado, incluindo detalhes sobre duas de suas propriedades cujos valores foram alterados. Nenhum recurso foi excluído.

```

aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-resource-drift --stack-resource-drift-status-filters MODIFIED DELETED

```

Resultado:

```

{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0, \"DelaySeconds\":120, \"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\", \"maxReceiveCount\":12}, \"MessageRetentionPeriod\":345600, \"MaximumMessageSize\":262144, \"VisibilityTimeout\":60, \"QueueName\":\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}",
      "ResourceType": "AWS::SQS::Queue",
      "Timestamp": "2018-03-26T17:23:34.489Z",
      "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-Queue-494PBHC076H4",
      "StackResourceDriftStatus": "MODIFIED",
      "ExpectedProperties": "{\"ReceiveMessageWaitTimeSeconds\":0, \"DelaySeconds\":20, \"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\", \"maxReceiveCount\":10}, \"MessageRetentionPeriod\":345600, \"MaximumMessageSize"
    }
  ]
}

```

```

\" :262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-
Queue-494PBHC076H4\"}],
  \"PropertyDifferences\": [
    {
      \"PropertyPath\": \"/DelaySeconds\",
      \"ActualValue\": \"120\",
      \"ExpectedValue\": \"20\",
      \"DifferenceType\": \"NOT_EQUAL\"
    },
    {
      \"PropertyPath\": \"/RedrivePolicy/maxReceiveCount\",
      \"ActualValue\": \"12\",
      \"ExpectedValue\": \"10\",
      \"DifferenceType\": \"NOT_EQUAL\"
    }
  ],
  \"LogicalResourceId\": \"Queue\"
}
]
}

```

Detectar desvios em recursos de pilha individuais

Você pode detectar desvios em recursos específicos dentro de uma pilha, em vez de na pilha inteira. Isso é especialmente útil quando você só precisa determinar se recursos específicos agora correspondem às configurações de modelo esperadas novamente.

Ao realizar a detecção de desvio em um recurso, o CloudFormation também atualiza o status global de desvio da pilha e o Tempo da última verificação de desvio, se aplicável. Por exemplo, suponha que uma pilha tenha um status de desvio de `IN_SYNC`. O CloudFormation realiza a detecção de desvio de um ou mais recursos contidos nessa pilha, e o CloudFormation detecta que um ou mais desses recursos se desviou. O CloudFormation atualiza o status de desvio da pilha para `DRIFTED`. Por outro lado, suponha que você tenha uma pilha com um status de desvio de `DRIFTED` como resultado de um único recurso desviado. Se você definir esse recurso de volta para os valores de propriedade esperados e, em seguida, detectar o desvio no recurso novamente, o CloudFormation atualizará o status de desvio de recurso e o status de desvio de pilha para `IN_SYNC` sem exigir que você detecte o desvio na pilha inteira novamente.

Para detectar desvios em um recurso individual usando o Console de gerenciamento da AWS

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na lista de pilhas, selecione a pilha que contém o recurso. O CloudFormation exibe os detalhes dessa pilha.
3. No painel de navegação esquerdo, em Pilhas, escolha Ações de pilha e Detectar desvio.
4. Em Resource drift status (Status de desvio do recurso), escolha o recurso e selecione Detect drift for resource (Detectar desvio para o recurso).

O CloudFormation realiza a detecção de desvio no recurso selecionado. Se tiver êxito, o CloudFormation atualizará o status de desvio do recurso e o status geral de desvio da pilha, se necessário. O CloudFormation também atualizará o timestamp de quando a detecção de desvio foi executada pela última vez no recurso, e a pilha como um todo. Se o recurso tiver sido modificado, o CloudFormation exibirá informações detalhadas de desvio sobre os valores de propriedade esperados e atuais do recurso.

5. Revise os resultados da detecção de desvio do recurso.
 - Para visualizar os detalhes em um recurso modificado.
 - Com o recurso modificado selecionado, escolha View drift details (Visualizar detalhes de desvio).

O CloudFormation exibe os detalhes de desvio para esse recurso, incluindo os valores de propriedade esperados e atuais do recurso e quaisquer diferenças entre os dois.

Para destacar uma diferença, na seção Differences (Diferenças), selecione o nome da propriedade.

- As propriedades adicionadas são destacadas em verde na coluna Current (Atuais) da seção Details (Detalhes).
- As propriedades excluídas são destacadas em vermelho na coluna Expected (Esperadas) da seção Details (Detalhes).
- As propriedades cujo valor foi alterado são destacadas em amarelo nas colunas Expected (Esperadas) e Current (Atuais).

Differences (4)				
☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName":"stuff","AttributeType":"S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key":"test","Value":"test"}]	-

Details

Expected

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Para detectar desvios em um recurso individual usando o AWS CLI

- 

Important

Consulte o Horário da última verificação de desvio do recurso da pilha e confirme se é anterior ao carimbo de data/hora mostrado nos resultados do desvio do recurso para evitar o uso de dados desatualizados.

Para detectar desvios em um recurso individual usando a AWS CLI, use o comando `detect-stack-resource-drift`. Especifique o ID lógico do recurso, bem como a pilha na qual ele está contido.

O exemplo a seguir executa uma operação de detecção de desvio em recursos de pilha específicos, `my-drifted-resource`. A resposta retorna informações que confirmam que o recurso foi modificado, incluindo detalhes sobre duas de suas propriedades cujos valores foram alterados.

```
aws cloudformation detect-stack-resource-drift \
  --stack-name my-stack-with-resource-drift \
  --logical-resource-id my-drifted-resource
```

Resultado:

```
{
  "StackResourceDrift": {
    "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
    "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds\":120,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",\"maxReceiveCount\":12},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}",
    "ResourceType": "AWS::SQS::Queue",
    "Timestamp": "2018-03-26T18:54:28.462Z",
    "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-Queue-494PBHC076H4",
    "StackResourceDriftStatus": "MODIFIED",
    "ExpectedProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds\":20,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",\"maxReceiveCount\":10},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}",
    "PropertyDifferences": [
      {
        "PropertyPath": "/DelaySeconds",
        "ActualValue": "120",
```

```
        "ExpectedValue": "20",
        "DifferenceType": "NOT_EQUAL"
    },
    {
        "PropertyPath": "/RedrivePolicy/maxReceiveCount",
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
    }
],
"LogicalResourceId": "my-drifted-resource"
}
```

Resolver o desvio com uma operação de importação

Pode haver casos em que a configuração de um recurso tenha se desviado da configuração pretendida e você queira aceitar a nova configuração como a configuração pretendida. Na maioria dos casos, você resolveria os resultados do desvio atualizando a definição de recurso no modelo de pilha com uma nova configuração e, depois, executaria uma atualização de pilha. No entanto, se a nova configuração atualizar uma propriedade de recurso que exija substituição, o recurso será recriado durante a atualização da pilha. Se desejar reter o recurso existente, você poderá usar a importação de recursos para atualizar o recurso e resolver os resultados de desvio sem substituí-lo.

Resolver o desvio de um recurso por meio de uma operação de importação consiste nas seguintes etapas básicas:

- [Adicione um atributo `DeletionPolicy`, definido como `Retain`, ao recurso](#). Isso garante que o recurso existente seja retido em vez de excluído quando ele é removido da pilha.
- [Remova o recurso do modelo e execute uma operação de atualização de pilha](#). Isso remove o recurso da pilha, mas não o exclui.
- [Descreva o estado real do recurso no modelo de pilha e, depois, importe o recurso existente de volta para a pilha](#). Isso adiciona o recurso de volta à pilha e resolve as diferenças de propriedade que estavam causando os resultados de desvio.

Para obter mais informações sobre importação de recursos, consulte [Importar recursos da AWS para uma pilha do CloudFormation manualmente](#). Para obter uma lista de recursos que oferecem suporte à importação, consulte [Suporte a tipos de recursos](#).

Neste exemplo, usamos o seguinte modelo, chamado `templateToImport.json`.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
```

```
        "KeyType": "HASH"
    },
    ],
    "BillingMode": "PROVISIONED",
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
    }
}
}
```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  GamesTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Games
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED

```

```
ProvisionedThroughput:
  ReadCapacityUnits: 5
  WriteCapacityUnits: 1
```

Neste exemplo, vamos supor que um usuário alterou um recurso fora do CloudFormation. Depois de executar a detecção de desvios, descobrimos que GamesTable foi modificado BillingMode para PAY_PER_REQUEST. Para obter mais informações sobre a detecção de desvios, consulte [Detectar alterações de configuração não gerenciadas em pilhas e recursos com detecção de desvios](#).

Differences (3)					< 1 >
☒	Property	Change	Expected value	Current value	
☒	BillingMode	NOT_EQUAL	PROVISIONED	PAY_PER_REQUEST	
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	0	
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	1	0	

Details	
Expected	Actual
<pre>{ "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PROVISIONED", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 5, "WriteCapacityUnits": 1 }, "TableName": "Games" }</pre>	<pre>{ "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PAY_PER_REQUEST", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 0, "WriteCapacityUnits": 0 }, "TableName": "Games" }</pre>

Nossa pilha agora está desatualizada, nossos recursos estão ativos, mas queremos preservar a configuração de recursos pretendida. Podemos fazer isso resolvendo o desvio com uma operação de importação, sem interromper os serviços.

Resolver desvio com uma operação de importação usando o console do CloudFormation

Etapa 1. Atualizar pilha com a política de exclusão Retain

Para atualizar a pilha usando um atributo **DeletionPolicy** com a opção **Retain**

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Pilhas escolha a pilha com desvio.
3. Escolha Atualizar, e, depois, selecione Substituir modelo atual no painel de detalhes da pilha.
4. Na página Especificar modelo forneça o modelo atualizado que contém o atributo DeletionPolicy com a opção Retain usando um dos seguintes métodos:

- Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
- Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.

Em seguida, escolha Próximo.

5. Revise a página Especificar detalhes da pilha e escolha Próximo.
6. Revise a página Configurar opções de pilha e escolha Próximo.
7. Na página Revisar **nome da pilha**, escolha Atualizar pilha.

Resultados: na página Eventos da sua pilha, o status é UPDATE_COMPLETE.

Para resolver o desvio com uma operação de importação, sem interromper os serviços, especifique uma Retain [DeletionPolicy](#) para os recursos que você deseja remover da pilha. Neste exemplo, adicionamos um atributo [DeletionPolicy](#) definido como Retain, ao recurso GameTable.

Example JSON

```
"GameTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
```

Example YAML

```
GameTable:
  Type: AWS::DynamoDB::Table
  DeletionPolicy: Retain
  Properties:
    TableName: Games
```

Etapa 2. Remover recursos com desvio, parâmetros relacionados e saídas

Como remover recursos com desvio, parâmetros relacionados e saídas

1. Escolha Atualizar, e, depois, selecione Substituir modelo atual no painel de detalhes da pilha.

2. Na página Especificar modelo, forneça ao modelo atualizado seus recursos, parâmetros relacionados e saídas removidas do modelo de pilha usando um dos seguintes métodos:
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.

Em seguida, escolha Próximo.

3. Revise a página Especificar detalhes da pilha e escolha Próximo.
4. Revise a página Configurar opções de pilha e escolha Próximo.
5. Na página Revisar **nome da pilha**, escolha Atualizar pilha.

Resultados: o ID Lógico GameSTable tem o status DELETE_SKIPPED na página Eventos da sua pilha.

Aguarde até que o CloudFormation conclua a operação de atualização da pilha. Após a conclusão da operação de atualização da pilha, remova o recurso, os parâmetros relacionados e as saídas do modelo de pilha. Depois, importe o modelo atualizado. Depois de concluir essas ações, o modelo demonstrativo agora se parece com o seguinte.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
```



```

        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "BillingMode": "PROVISIONED",
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}
}

```

Example YAML

```

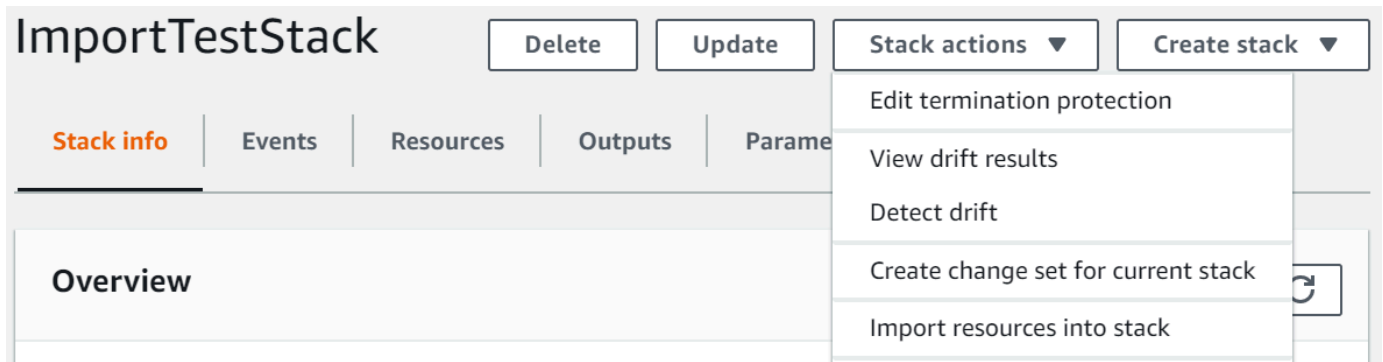
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1

```

Etapa 3. Atualizar modelo para corresponder ao estado ativo dos seus recursos

Como atualizar o modelo para corresponder ao estado ativo dos recursos

1. Para importar o modelo atualizado, escolha Ações da pilha e, depois, escolha Importar recursos para a pilha.



2. Leia a página Visão geral da importação para obter uma lista de itens a serem fornecidos durante esta operação, e, depois, escolha Próximo.
3. Na páginaEspecificar modelo, forneça o modelo atualizado usando um dos seguintes métodos:
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.

Em seguida, escolha Próximo.

4. Na página Identify resources (Identificar recursos), identifique todos os recursos de destino. Para obter mais informações, consulte [Identificadores de recursos](#).
 - a. Em Identifier property (Propriedade do identificador), escolha o tipo de identificador do recurso. Por exemplo, a propriedade TableName identifica o recurso `AWS::DynamoDB::Table`.
 - b. Em Valor do identificador, digite o valor real da propriedade. No modelo demonstrativo, o TableName para o recurso GamesTable é Games.
 - c. Escolha Próximo.
5. Revise a página Especificar detalhes da pilha e escolha Próximo.
6. Na página Visão geral da importação, revise os recursos que estão sendo importados e escolha Importar recursos. Isso importará o tipo de recurso `AWS::DynamoDB::Table` de volta para sua pilha.

Resultados: neste exemplo, resolvemos o desvio de recursos com uma operação de importação, sem interromper os serviços. É possível verificar o andamento de uma ação de importação no console do CloudFormation na guia Eventos. Os recursos importados terão o status

IMPORT_COMPLETE seguido pelo status CREATE_COMPLETE com Importação de recursos concluída como o motivo do status.

Aguarde até que o CloudFormation conclua a operação de atualização da pilha. Depois que a operação de atualização da pilha for concluída, atualize seu modelo para corresponder ao estado real e com desvio de seus recursos. Por exemplo, o BillingMode será definido como PAY_PER_REQUEST e ReadCapacityUnits e WriteCapacityUnits serão definidos como 0.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
```

```
{
    "AttributeName": "key",
    "AttributeType": "S"
},
"KeySchema": [
    {
        "AttributeName": "key",
        "KeyType": "HASH"
    }
],
"BillingMode": "PAY_PER_REQUEST",
"ProvisionedThroughput": {
    "ReadCapacityUnits": 0,
    "WriteCapacityUnits": 0
}
}
```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
    ServiceTable:
        Type: AWS::DynamoDB::Table
        Properties:
            TableName: Service
            AttributeDefinitions:
                - AttributeName: key
                  AttributeType: S
            KeySchema:
                - AttributeName: key
                  KeyType: HASH
            BillingMode: PROVISIONED
            ProvisionedThroughput:
                ReadCapacityUnits: 5
                WriteCapacityUnits: 1
    GamesTable:
        Type: AWS::DynamoDB::Table
        DeletionPolicy: Retain

```

```
Properties:
  TableName: Games
  AttributeDefinitions:
    - AttributeName: key
      AttributeType: S
  KeySchema:
    - AttributeName: key
      KeyType: HASH
  BillingMode: PAY_PER_REQUEST
  ProvisionedThroughput:
    ReadCapacityUnits: 0
    WriteCapacityUnits: 0
```

Importar recursos da AWS para uma pilha do CloudFormation

Você pode importar recursos existentes para uma pilha do CloudFormation. Isso é útil se você quiser começar a usar o CloudFormation para gerenciar atributos que foram criados fora do CloudFormation sem precisar excluí-los e recriá-los.

O CloudFormation oferece as seguintes opções para importar recursos existentes em uma pilha:

- O [gerador de IaC](#) é uma ferramenta que verifica automaticamente seus recursos existentes e gera um modelo do CloudFormation com base em seu estado atual. Esse modelo pode então ser usado para importar esses recursos em uma pilha.
- A [importação de recursos](#) é um processo manual em que você descreve os recursos existentes em seu modelo do CloudFormation e depois os importa para uma pilha. Essa abordagem exige que você especifique manualmente as propriedades e configurações do recurso no modelo.
- A [importação automática](#) é um processo automático que descreve os recursos existentes no modelo do CloudFormation e no qual o CloudFormation importa os recursos com nomes personalizados correspondentes para uma pilha.
- A [refatoração de pilhas](#) é um recurso que simplifica a reorganização dos recursos em suas pilhas do CloudFormation e, ao mesmo tempo, preserva as propriedades e os dados dos recursos existentes. Com a refatoração de pilhas, é possível mover recursos entre pilhas, dividir pilhas monolíticas em componentes menores ou consolidar várias pilhas em uma.

Além de colocar os recursos existentes sob o gerenciamento do CloudFormation, o recurso de importação de recursos pode ser útil nos seguintes cenários:

- Movendo recursos entre pilhas: você pode importar recursos de uma pilha para outra, permitindo que você reorganize sua infraestrutura conforme necessário.
- Aninhamento de pilhas existentes: você pode importar uma pilha existente como uma pilha aninhada dentro de outra pilha, permitindo projetos de infraestrutura modulares e reutilizáveis.

O CloudFormation é compatível com a importação de uma ampla gama de recursos. Para obter mais informações, consulte [Suporte a tipos de recursos](#).

Tópicos

- [Importar recursos da AWS para uma pilha do CloudFormation manualmente](#)
- [Importar recursos da AWS para uma pilha do CloudFormation automaticamente](#)
- [Reverter uma operação de importação](#)

Importar recursos da AWS para uma pilha do CloudFormation manualmente

Com a importação de recursos, você pode importar recursos existentes da AWS para uma pilha nova ou existente do CloudFormation. Durante uma operação de importação, você cria um conjunto de alterações importando recursos existentes para uma pilha ou cria uma pilha com os recursos existentes. Você fornece o seguinte durante a importação.

- Um modelo que descreve toda a pilha, incluindo os recursos de pilha originais e os recursos que estão sendo importados. Todos os recursos que serão importados deverão ter um [atributo DeletionPolicy](#).
- Identificadores para os recursos que você está importando, que o CloudFormation pode usar para realizar o mapeamento dos IDs lógicos no modelo com os recursos existentes.

Note

O CloudFormation é compatível somente com um nível de aninhamento usando a importação de recursos. Isso significa que você não pode importar uma pilha em uma pilha filho ou importar uma pilha com filhos.

Tópicos

- [Identificadores de recursos](#)
- [Validação da importação de recursos](#)
- [Códigos de status de importação de recursos](#)
- [Considerações durante uma operação de importação](#)
- [Recursos adicionais](#)
- [Criar uma pilha com recursos existentes](#)
- [Importar recursos existentes para uma pilha](#)
- [Mover recursos entre pilhas](#)
- [Aninhar uma pilha existente](#)

Identificadores de recursos

Você fornece dois valores para identificar cada recurso que está sendo importado.

- Uma propriedade identificadora. Esta é uma propriedade de recursos que pode ser usada para identificar cada tipo de recurso. Por exemplo, um recurso `AWS::S3::Bucket` pode ser identificado usando o `BucketName`.

A propriedade do recurso usada para identificar o recurso que está sendo importado varia de acordo com o tipo de recurso. É possível encontrar a propriedade do recurso no console do CloudFormation. Depois de criar um modelo que inclua o recurso a ser importado, você poderá iniciar o processo de importação, no qual encontrará as propriedades do identificador para os recursos que estão sendo importados. Para alguns tipos de recursos, pode existir diversas maneiras de identificá-los, e você pode selecionar qual propriedade deseja usar nas listas suspensas.

Como alternativa, é possível obter as propriedades do identificador para os recursos que estão sendo importados ao chamar o comando [get-template-summary](#) da CLI e ao selecionar o URL do S3 do modelo de pilha como o valor da opção `--template-url`.

- Um valor do identificador. Esse é o valor real da propriedade do recurso. Por exemplo, o valor real da propriedade `BucketName` deve ser `MyS3Bucket`.

É possível obter o valor da propriedade do identificador usando o console de serviço do recurso.

Validação da importação de recursos

Durante uma operação de importação, o CloudFormation realiza as seguintes validações.

- O recurso a ser importado já existe.
- As propriedades e os valores de configuração de cada recurso a ser importado aderem ao esquema do tipo do recurso, que define as propriedades aceitas, as propriedades obrigatórias e os valores de propriedades compatíveis.
- As propriedades obrigatórias são especificadas no modelo. As propriedades obrigatórias de cada tipo de recurso estão listadas na [Referência de tipos de propriedades e recursos da AWS](#).
- O recurso a ser importado não pertence a outra pilha na mesma região.

O CloudFormation não verifica se a configuração do modelo corresponde à configuração real das propriedades de recursos.

Important

Verifique se os recursos e suas propriedades definidas no modelo correspondem à configuração pretendida da importação de recursos para evitar alterações inesperadas.

Códigos de status de importação de recursos

Essa tabela descreve os vários tipos de status usados com o recurso de importação de recursos.

Status da operação de importação	Descrição
IMPORT_IN_PROGRESS	A operação de importação está em andamento.
IMPORT_COMPLETE	A operação de importação foi concluída para todos os recursos da pilha.
IMPORT_ROLLBACK_IN_PROGRESS	A operação de reversão da importação está sendo revertida para a configuração de modelo anterior.
IMPORT_ROLLBACK_FAILED	Falha na operação de reversão da importação.

Status da operação de importação	Descrição
IMPORT_ROLLBACK_COMPLETE	A importação foi revertida para a configuração de modelo anterior.

Considerações durante uma operação de importação

- Após a importação ter sido concluída, e antes de realizar as operações de pilhas subsequentes, recomendamos executar a detecção de desvios nos recursos importados. A detecção de desvios garante que a configuração do modelo corresponda à configuração real. Para obter mais informações, consulte [Detectar desvio em uma pilha inteira do CloudFormation](#).
- As operações de importação não permitem criações de novos recursos, exclusões de recursos ou alterações para configurações de propriedade.
- Cada recurso a ser importado deve ter um atributo `DeletionPolicy` para que a operação de importação tenha êxito. O `DeletionPolicy` pode ser definido como qualquer valor possível. Somente os recursos que você está importando precisam de uma `DeletionPolicy`. Os recursos que já fazem parte da pilha não precisam de `DeletionPolicy`.
- Não é possível importar o mesmo recurso em várias pilhas.
- Você pode usar a condição da política `cloudformation:ImportResourceTypes` do IAM para controlar com quais tipos de recursos os usuários podem trabalhar durante uma operação de importação. Para obter mais informações, consulte [Chaves de condição de política do CloudFormation](#).
- Os limites de pilha do CloudFormation se aplicam à importação de recursos. Para obter mais informações sobre limites, consulte [Entender as cotas do CloudFormation](#).

Recursos adicionais

Para resolver o desvio de pilha com uma importação de recursos, consulte [Resolver o desvio com uma operação de importação](#).

Criar uma pilha com recursos existentes

Este tópico mostra como criar uma pilha a partir dos recursos da AWS existentes descrevendo-os em um modelo. Para, em vez disso, verificar os recursos existentes e gerar automaticamente um modelo que você possa usar para importar recursos existentes para o CloudFormation ou replicar os

recursos em uma nova conta, consulte [Gerar modelos a partir de recursos existentes com o gerador de IaC](#).

Pré-requisitos

Antes de começar, você deve ter o seguinte:

- Um modelo que descreve todos os recursos que você deseja em uma nova pilha. Salve o modelo do localmente ou em um bucket do Amazon S3.
- Para cada recurso que você deseja importar, inclua o seguinte:
 - as propriedades e os valores das propriedades que definem a configuração atual do recurso.
 - o identificador exclusivo do recurso, como seu nome. Para obter mais informações, consulte [Identificadores de recursos](#).
 - o [atributos DeletionPolicy](#).

Tópicos

- [Exemplo de modelo](#)
- [Criar uma pilha com recursos existentes usando a Console de gerenciamento da AWS](#)
- [Criar uma pilha com recursos existentes usando a AWS CLI](#)

Exemplo de modelo

Nesta demonstração, partimos do pressuposto de que você esteja usando o exemplo de modelo a seguir, denominado `TemplateToImport.json`, que especifica duas tabelas do DynamoDB que foram criadas fora do CloudFormation. A `ServiceTable` e a `GameTable` são os destinos de importação.

Note

Este modelo serve apenas de exemplo. Para usá-lo em seus próprios testes, substitua os recursos do exemplo por recursos da sua conta.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
```

```
"ServiceTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Service",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
},
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

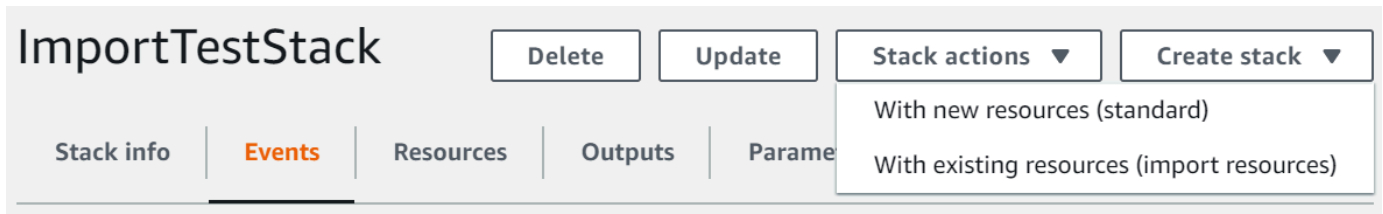
```

    }
  }
}

```

Criar uma pilha com recursos existentes usando a Console de gerenciamento da AWS

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Stacks (Pilhas), escolha Create stack (Criar pilha) e With existing resources (import resources) (Com recursos existentes (importar recursos)).



3. Leia a página Import overview (Visão geral da importação) para obter uma lista de itens que você deve fornecer durante esta operação. Em seguida, escolha Próximo.
4. Na página Specify template (Especificar modelo), forneça o modelo usando um dos métodos a seguir e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.
5. Na página Identify resources (Identificar recursos), identifique todos os recursos de destino. Para obter mais informações, consulte [Identificadores de recursos](#).
 - a. Em Identifier property (Propriedade do identificador), escolha o tipo de identificador do recurso. Por exemplo, o recurso `AWS::DynamoDB::Table` pode ser identificado usando a propriedade `TableName`.
 - b. Em Identifier value (Valor do identificador), digite o valor real da propriedade. Por exemplo, o `TableName` para o recurso `GamesTable` no modelo de exemplo é *Games*.
 - c. Escolha Próximo.
6. Na página Specify stack details (Especificar detalhes da pilha), modifique os parâmetros e escolha Next (Próximo). Isso cria automaticamente um conjunto de alterações.

Important

A operação de importação falhará se você modificar os parâmetros existentes que iniciam uma operação de criação, atualização ou exclusão.

7. Na página Review **stack-name** (Revisar nome-pilha), confirme se os recursos corretos estão sendo importados e escolha Import resources (Importar recursos). Isso executa automaticamente o conjunto de alterações criado na última etapa.

O painel Events (Eventos) da página Stack details (Detalhes da pilha) da nova pilha é exibido.

ImportTestStack Delete Update Stack actions ▼ Create stack ▼

Stack info **Events** Resources Outputs Parameters Template Change sets

Events 🔄

🔍 Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	🟢 IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🟢 UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	🟢 IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	🔄 IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	🔄 IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	🔄 REVIEW_IN_PROGRESS	User Initiated

8. (Opcional) Execute a detecção de desvios na pilha para garantir que o modelo e a configuração real dos recursos importados sejam correspondentes. Para obter mais informações sobre como detectar desvios, consulte [Detectar desvio em uma pilha inteira do CloudFormation](#).
9. (Opcional) Se os recursos importados não corresponderem às configurações de modelo esperadas, corrija as configurações do modelo ou atualize diretamente os recursos. Nesta demonstração, corrigimos as configurações do modelo para corresponder às configurações reais.
 - a. [Reverter a operação de importação](#) para os recursos afetados.
 - b. Adicione os destinos de importação ao modelo novamente, garantindo que as configurações do modelo correspondam às configurações atuais.
 - c. Repita as etapas de 2 a 8 usando o modelo modificado para importar os recursos novamente.

Criar uma pilha com recursos existentes usando a AWS CLI

1. Para saber quais propriedades identificam cada tipo de recurso no modelo, execute o comando `get-template-summary`, especificando o URL do S3 do modelo. Por exemplo, o recurso `AWS::DynamoDB::Table` pode ser identificado usando a propriedade `TableName`. Para o recurso `GamesTable` no modelo de exemplo, o valor de `TableName` é `Games`. Você precisará dessas informações na próxima etapa.

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

Para obter mais informações, consulte [Identificadores de recursos](#).

2. Componha uma lista com os recursos reais do seu modelo e os identificadores exclusivos no formato de string JSON apresentado a seguir.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{
  "TableName":"Games"}},
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":{
  "TableName":"Service"}}]
```

Como alternativa, é possível especificar os parâmetros formatados em JSON em um arquivo de configuração.

Por exemplo, para importar `ServiceTable` e `GamesTable`, é possível criar um arquivo *ResourcesToImport.txt* que contém a configuração apresentada a seguir.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier":{
      "TableName":"Games"
    }
  },
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"ServiceTable",
    "ResourceIdentifier":{
      "TableName":"Service"
    }
  }
]
```

```
    }
  }
]
```

3. Para criar um conjunto de alterações, use o comando `create-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. Para a opção `--change-set-type` especifique um valor de **IMPORT**. Para a opção `--resources-to-import`, substitua a string JSON de exemplo pela string JSON real que você acabou de criar.

```
aws cloudformation create-change-set \
  --stack-name TargetStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json \
  --resources-to-
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifi
{"TableName":"Games"}},
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifi
{"TableName":"Service"}}]'
```

Note

A opção `--resources-to-import` não oferece suporte para a linguagem YAML incorporada. Os requisitos para usar sequências de escape de aspas em uma string JSON variam de acordo com o terminal utilizado. Para obter mais informações, consulte [Using quotation marks inside strings](#) no Guia do usuário da AWS Command Line Interface.

Como alternativa, é possível usar um URL de arquivo como entrada para a opção `--resources-to-import`, conforme mostrado no exemplo a seguir.

```
--resources-to-import file://ResourcesToImport.txt
```

4. Revise o conjunto de alterações para garantir que os recursos corretos sejam importados.

```
aws cloudformation describe-change-set \
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Para iniciar o conjunto de alterações e importar os recursos, use o comando `execute-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. Após a conclusão bem-sucedida da operação (`IMPORT_COMPLETE`), os recursos são importados com êxito.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opcional) Execute a detecção de desvios na pilha `IMPORT_COMPLETE` para garantir que o modelo e a configuração real dos recursos importados sejam correspondentes. Para obter mais informações sobre como detectar desvios, consulte [Detectar desvios em recursos de pilha individuais](#).
 - a. Execute a detecção de desvio na pilha especificada.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Se a detecção ocorrer com êxito, esse comando retornará o exemplo de saída apresentado a seguir.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Visualize o progresso de uma operação de detecção de desvio para o ID de detecção de desvio de pilha especificado.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Visualize as informações de desvio para os recursos que foram verificados em busca de desvios na pilha especificada.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opcional) Se os recursos importados não corresponderem às configurações de modelo esperadas, corrija as configurações do modelo ou atualize diretamente os recursos. Nesta demonstração, corrigimos as configurações do modelo para corresponder às configurações reais.
 - a. [Reverter a operação de importação](#) para os recursos afetados.

- b. Adicione os destinos de importação ao modelo novamente, garantindo que as configurações do modelo correspondam às configurações atuais.
- c. Repita as etapas de 3 a 6 usando o modelo modificado para importar os recursos novamente.

Importar recursos existentes para uma pilha

Este tópico mostra como importar recursos existentes da AWS para uma pilha existente, descrevendo-os em um modelo. Para, em vez disso, verificar os recursos existentes e gerar automaticamente um modelo que você possa usar para importar recursos existentes para o CloudFormation ou replicar os recursos em uma nova conta, consulte [Gerar modelos a partir de recursos existentes com o gerador de IaC](#).

Pré-requisitos

Antes de começar, você deve ter o seguinte:

- Um modelo que descreva toda a pilha, incluindo tanto os recursos que já fazem parte da pilha quanto os recursos a serem importados. Salve o modelo do localmente ou em um bucket do Amazon S3.

Para obter uma cópia de um modelo de pilha em execução

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation/>.
 2. Na lista de pilhas, escolha a pilha da qual deseja recuperar o modelo.
 3. No painel de detalhes da pilha, escolha a guia Modelo e escolha Copiar para a área de transferência.
 4. Cole o código em um editor de texto para começar a adicionar outros recursos ao modelo.
- Para cada recurso que você deseja importar, inclua o seguinte:
 - as propriedades e os valores das propriedades que definem a configuração atual do recurso.
 - o identificador exclusivo do recurso, como seu nome. Para obter mais informações, consulte [Identificadores de recursos](#).
 - o [atributo DeletionPolicy](#).

Tópicos

- [Exemplo de modelo](#)

- [Importar um recurso existente para a pilha usando a Console de gerenciamento da AWS](#)
- [Importar um recurso existente para a pilha usando a AWS CLI](#)

Exemplo de modelo

Nesta demonstração, partimos do pressuposto de que você esteja usando o exemplo de modelo a seguir, denominado `TemplateToImport.json`, que especifica duas tabelas do DynamoDB. A `ServiceTable` faz parte da pilha atualmente e a `GamesTable` é a tabela que você deseja importar.

Note

Este modelo serve apenas de exemplo. Para usá-lo em seus próprios testes, substitua os recursos do exemplo por recursos da sua conta.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
  },
}
```

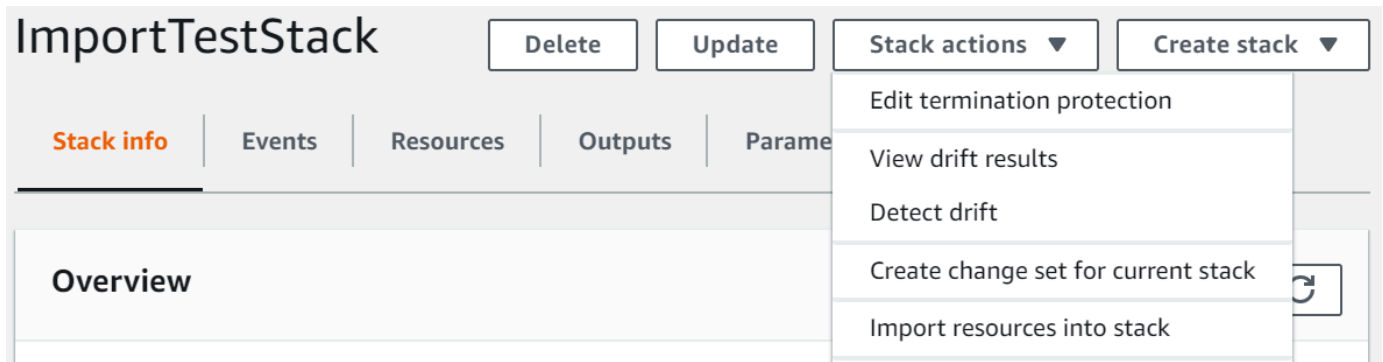
```
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

Importar um recurso existente para a pilha usando a Console de gerenciamento da AWS

 Note

O console do CloudFormation não oferece suporte ao uso da função intrínseca `Fn::Transform` ao importar recursos. É possível usar o AWS CLI para importar recursos que usem a função `Fn::Transform`.

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página Stacks (Pilhas), escolha a pilha para a qual deseja importar os recursos.
3. Escolha Stack actions (Ações da pilha) e Import resources into stack (Importar recursos para a pilha).



4. Consulte a página Import overview (Visão geral da importação) e escolha Next (Próximo).
5. Na página Specify template (Especificar modelo), forneça o modelo atualizado usando um dos métodos a seguir e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.
6. Na página Identify resources (Identificar recursos), identifique todos os recursos de destino. Para obter mais informações, consulte [Identificadores de recursos](#).
 - a. Em Identifier property (Propriedade do identificador), escolha o tipo de identificador do recurso. Por exemplo, o recurso `AWS::DynamoDB::Table` pode ser identificado usando a propriedade `TableName`.
 - b. Em Identifier value (Valor do identificador), digite o valor real da propriedade. Por exemplo, o `TableName` para o recurso `GamesTable` no modelo de exemplo é *Games*.
 - c. Escolha Próximo.
7. Na página Specify stack details (Especificar detalhes da pilha), atualize os parâmetros e escolha Next (Próximo). Isso cria automaticamente um conjunto de alterações.

 Note

A operação de importação falhará se você modificar os parâmetros existentes que iniciam uma operação de criação, atualização ou exclusão.

8. Na página Review *stack-name* (Revisar nome-pilha), revise os recursos a serem importados e escolha Import resources (Importar recursos). Isso executa automaticamente o conjunto

de alterações criado na última etapa. As tags no nível da pilha são aplicadas aos recursos importados no momento. Para obter mais informações, consulte [Configurar opções da pilha](#).

A página Events (Eventos) da pilha é exibida.

The screenshot shows the AWS CloudFormation console interface for a stack named 'ImportTestStack'. The 'Events' tab is selected, displaying a table of stack events. The table has columns for Timestamp, Logical ID, Status, and Status reason. The events show the progression of the stack import, from 'User Initiated' to 'REVIEW_IN_PROGRESS', 'IMPORT_IN_PROGRESS', 'UPDATE_IN_PROGRESS', and finally 'IMPORT_COMPLETE' and 'UPDATE_COMPLETE'.

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

- (Opcional) Execute a detecção de desvios na pilha para garantir que o modelo e a configuração real dos recursos importados sejam correspondentes. Para obter mais informações sobre como detectar desvios, consulte [Detectar desvio em uma pilha inteira do CloudFormation](#).
- (Opcional) Se os recursos importados não corresponderem às configurações de modelo esperadas, corrija as configurações do modelo ou atualize diretamente os recursos. Para obter mais informações sobre como importar recursos com desvio, consulte [Resolver o desvio com uma operação de importação](#).

Importar um recurso existente para a pilha usando a AWS CLI

- Para saber quais propriedades identificam cada tipo de recurso no modelo, execute o comando `get-template-summary`, especificando o URL do S3 do modelo. Por exemplo, o recurso `AWS::DynamoDB::Table` pode ser identificado usando a propriedade `TableName`. Para o recurso `GameTable` no modelo de exemplo, o valor de `TableName` é `Games`. Você precisará dessas informações na próxima etapa.

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

Para obter mais informações, consulte [Identificadores de recursos](#).

2. Componha uma lista com os recursos reais a serem importados e os identificadores exclusivos no formato de string JSON apresentado a seguir.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

Como alternativa, é possível especificar os parâmetros formatados em JSON em um arquivo de configuração.

Por exemplo, para importar GamesTable, é possível criar um arquivo *ResourcesToImport.txt* que contém a configuração apresentada a seguir.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier": {
      "TableName":"Games"
    }
  }
]
```

3. Para criar um conjunto de alterações, use o comando `create-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. Para a opção `--change-set-type` especifique um valor de **IMPORT**. Para a opção `--resources-to-import`, substitua a string JSON de exemplo pela string JSON real que você acabou de criar.

```
aws cloudformation create-change-set \
  --stack-name TargetStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json \
  --resources-to-
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]'
```

Note

A opção `--resources-to-import` não oferece suporte para a linguagem YAML incorporada. Os requisitos para usar sequências de escape de aspas em uma string

JSON variam de acordo com o terminal utilizado. Para obter mais informações, consulte [Using quotation marks inside strings](#) no Guia do usuário da AWS Command Line Interface.

Como alternativa, é possível usar um URL de arquivo como entrada para a opção `--resources-to-import`, conforme mostrado no exemplo a seguir.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Revise o conjunto de alterações para garantir que os recursos corretos sejam importados.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Para iniciar o conjunto de alterações e importar os recursos, use o comando `execute-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. As tags no nível da pilha são aplicadas aos recursos importados no momento. Para obter mais informações, consulte [Configurar opções da pilha](#). Após a conclusão bem-sucedida da operação (`IMPORT_COMPLETE`), os recursos são importados com êxito.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opcional) Execute a detecção de desvios na pilha `IMPORT_COMPLETE` para garantir que o modelo e a configuração real dos recursos importados sejam correspondentes. Para obter mais informações sobre como detectar desvios, consulte [Detectar desvio em uma pilha inteira do CloudFormation](#).

- a. Execute a detecção de desvio na pilha especificada.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Se a detecção ocorrer com êxito, esse comando retornará o exemplo de saída apresentado a seguir.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Visualize o progresso de uma operação de detecção de desvio para o ID de detecção de desvio de pilha especificado.

```
aws cloudformation describe-stack-drift-detection-status \
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Visualize as informações de desvio para os recursos que foram verificados em busca de desvios na pilha especificada.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opcional) Se os recursos importados não corresponderem às configurações de modelo esperadas, corrija as configurações do modelo ou atualize diretamente os recursos. Para obter mais informações sobre como importar recursos com desvio, consulte [Resolver o desvio com uma operação de importação](#).

Mover recursos entre pilhas

Com o recurso `resource import`, é possível mover recursos entre pilhas ou refatorar pilhas. Primeiro, você precisa adicionar uma política de exclusão `Retain` ao recurso que você deseja mover para garantir que o recurso seja preservado ao removê-lo da pilha de origem e importá-lo na pilha de destino.

Se você não tiver familiaridade com a importação, recomendamos que primeiro revise as informações introdutórias no tópico [Importar recursos da AWS para uma pilha do CloudFormation](#).

Important

Nem todos os recursos são compatíveis com as operações de importação. Consulte [Recursos compatíveis com as operações de importação](#) antes de remover um recurso da pilha. Se você remover um recurso que não é compatível com operações de importação da pilha, não será possível importar o recurso para outra pilha ou trazê-lo de volta para a pilha de origem.

Refatorar uma pilha usando o Console de gerenciamento da AWS

1. No modelo de origem, especifique uma [DeletionPolicy](#) `Retain` para o recurso que deseja mover.

No modelo de origem de exemplo a seguir, Games é o destino desta refatoração.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
```

```
        "AttributeName": "key",
        "KeyType": "HASH"
      },
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

2. Abra o console do CloudFormation para executar uma atualização de pilha para aplicar a política de exclusão.
 - a. Na página Stacks (Pilhas) com a pilha selecionada, escolha Update (Atualizar).
 - b. Em Preparar modelo, escolha Substituir modelo atual.
 - c. Em Specify template (Especificar modelo), forneça o modelo de origem atualizado com o atributo `DeletionPolicy` na `GameTable` e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL para o modelo de origem atualizado na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo de origem atualizado.
 - d. Na página Specify stack details (Especificar detalhes da pilha) nenhuma alteração é necessária. Escolha Próximo.
 - e. Na página Configure stack options (Configurar opções de pilha) nenhuma alteração é necessária. Escolha Próximo.
 - f. Na página Analisar **SourceStackName**, analise as alterações. Se o seu modelo contém recursos do IAM, selecione I acknowledge that this template may create IAM resources (Eu reconheço que este modelo pode criar recursos do IAM) para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações sobre o uso de recursos do IAM em modelos, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#). Depois, atualize sua pilha de origem criando um conjunto de alterações ou atualize sua pilha de origem diretamente.
3. Remova o recurso, os parâmetros relacionados e as saídas do modelo de origem e adicione-os ao modelo de destino.

O modelo de origem agora se parece com o seguinte.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

O modelo de destino de exemplo a seguir tem atualmente o recurso `PlayersTable` e agora também contém `GamesTable`.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
```

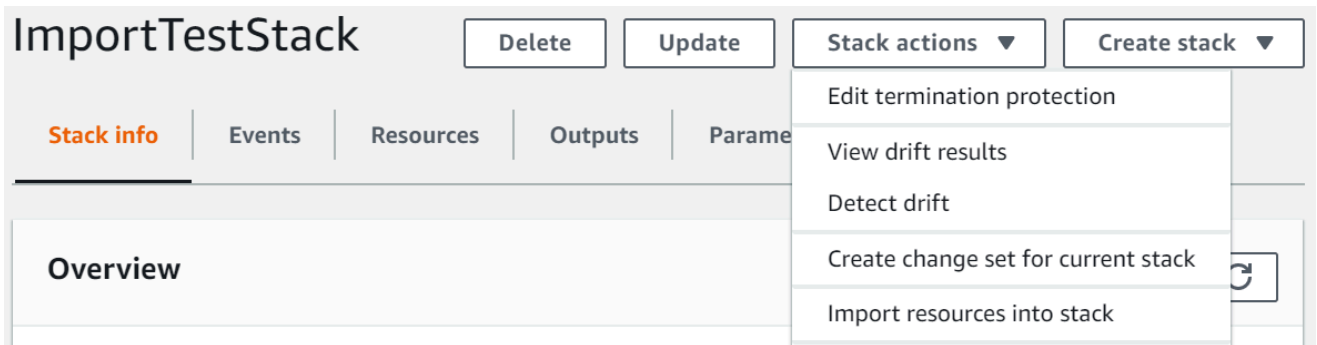
```
"PlayersTable": {
  "Type": "AWS::DynamoDB::Table",
  "Properties": {
    "TableName": "Players",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
},
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

```

    }
  }
}

```

4. Repita as etapas 2 e 3 para atualizar a pilha de origem novamente, desta vez para excluir o recurso de destino da pilha.
5. Execute uma operação de importação para adicionar `GameTable` à pilha de destino.
 - a. Na página Stacks (Pilhas), com a pilha pai selecionada, escolha Stack actions (Ações da pilha) e Import resources into stack (Importar recursos para a pilha).



- b. Leia a página Import overview (Visão geral da importação) para obter uma lista de itens que você deve fornecer durante esta operação. Em seguida, escolha Próximo.
- c. Na página Specify template (Especificar modelo), siga um destes procedimentos e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique um URL na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure um arquivo a ser carregado.
- d. Na página Identify resources (Identificar recursos) identifique o recurso que você está movendo (neste exemplo, `GameTable`). Para obter mais informações, consulte [Identificadores de recursos](#).
 1. Em Identifier property (Propriedade do identificador), escolha o tipo de identificador do recurso. Por exemplo, um recurso `AWS::DynamoDB::Table` pode ser identificado usando a propriedade `TableName`.
 2. Em Identifier value (Valor do identificador), digite o valor real da propriedade. Por exemplo, `.GameTables`
 3. Escolha Próximo.

- e. Na página Specify stack details (Especificar detalhes da pilha), modifique os parâmetros e escolha Next (Próximo). Isso cria automaticamente um conjunto de alterações.

Important

A operação de importação falhará se você modificar os parâmetros existentes que iniciam uma operação de criação, atualização ou exclusão.

- f. Na página Analisar **TargetStackName**, confirme se o recurso correto está sendo importado e, em seguida, escolha Importar recursos. Isso inicia automaticamente o conjunto de alterações criadas na última etapa. As [tags no nível da pilha](#) são aplicadas aos recursos importados no momento.
- g. O painel Events (Eventos) da página Stack details (Detalhes da pilha) de sua pilha pai será exibido.

ImportTestStack Delete Update Stack actions ▼ Create stack ▼

Stack info **Events** Resources Outputs Parameters Template Change sets

Events 🔄

🔍 Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	🟢 IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🟢 UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	🟢 IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	🔄 IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	🔄 IMPORT_IN_PROGRESS	User initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	🔄 REVIEW_IN_PROGRESS	User initiated

Note

Não é necessário executar a detecção de desvio na pilha pai após essa operação de importação, pois o recurso `AWS::CloudFormation::Stack` já é gerenciado pelo CloudFormation.

Refatorar uma pilha usando o AWS CLI

1. No modelo de origem, especifique uma [DeletionPolicy](#) Retain para o recurso que deseja mover.

No modelo de origem de exemplo a seguir, GamesTable é o destino desta refatoração.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
```

```

        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}
}
}

```

2. Atualize a pilha de origem para aplicar a política de exclusão ao recurso.

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. Remova o recurso, os parâmetros relacionados e as saídas do modelo de origem e adicione-os ao modelo de destino.

O modelo de origem agora se parece com o seguinte.

Example JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}

```



```
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
```

O modelo de destino de exemplo a seguir tem atualmente o recurso `PlayersTable` e agora também contém `GamesTable`.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "PlayersTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Players",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
```

```

        "DeletionPolicy": "Retain",
        "Properties": {
            "TableName": "Games",
            "AttributeDefinitions": [
                {
                    "AttributeName": "key",
                    "AttributeType": "S"
                }
            ],
            "KeySchema": [
                {
                    "AttributeName": "key",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 1
            }
        }
    }
}

```

4. Atualize a pilha de origem para excluir o recurso `GamesTable` e os parâmetros relacionados, além das saídas da pilha.

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. Componha uma lista com os recursos reais a serem importados e os identificadores exclusivos no formato de string JSON apresentado a seguir. Para obter mais informações, consulte [Identificadores de recursos](#).

```
[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": "GamesTable", "PhysicalResourceId": "GamesTable", "ImportedResource": {"TableName": "Games"}}]
```

Como alternativa, é possível especificar os parâmetros formatados em JSON em um arquivo de configuração.

Por exemplo, para importar `GamesTable`, é possível criar um arquivo *ResourcesToImport.txt* que contém a configuração apresentada a seguir.

```
[
  {
    "ResourceType": "AWS::DynamoDB::Table",
    "LogicalResourceId": "GamesTable",
    "ResourceIdentifier": {
      "TableName": "Games"
    }
  }
]
```

6. Para criar um conjunto de alterações, use o comando `create-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. Para a opção `--change-set-type` especifique um valor de **IMPORT**. Para a opção `--resources-to-import`, substitua a string JSON de exemplo pela string JSON real que você acabou de criar.

```
aws cloudformation create-change-set \
  --stack-name TargetStackName --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-import
  "'[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": {"TableName": "Games"}}]'"
```

Note

A opção `--resources-to-import` não oferece suporte para a linguagem YAML incorporada. Os requisitos para usar sequências de escape de aspas em uma string JSON variam de acordo com o terminal utilizado. Para obter mais informações, consulte [Using quotation marks inside strings](#) no Guia do usuário da AWS Command Line Interface.

Como alternativa, é possível usar um URL de arquivo como entrada para a opção `--resources-to-import`, conforme mostrado no exemplo a seguir.

```
--resources-to-import file://ResourcesToImport.txt
```

7. Revise o conjunto de alterações para certificar-se de que o recurso correto está sendo importado para a pilha de destino.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet
```

8. Para iniciar o conjunto de alterações e importar o recurso, use o comando `execute-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. As tags no nível da pilha são aplicadas aos recursos importados no momento. Após a conclusão com êxito da operação (`IMPORT_COMPLETE`), o recurso será importado com êxito.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStackName
```

 Note

Não é necessário executar a detecção de desvio na pilha de destino após a operação de importação porque o recurso já é gerenciado pelo CloudFormation.

Aninhar uma pilha existente

Use o recurso `resource import` para aninhar uma pilha existente em outra pilha existente. As pilhas aninhadas são componentes comuns que você declara e às quais faz referência a partir de outros modelos. Dessa forma, você pode evitar copiar e colar as mesmas configurações nos modelos e simplificar as atualizações de pilha. Se tiver um modelo para um componente comum, você poderá usar o recurso `AWS::CloudFormation::Stack` para fazer referência a este modelo a partir de outro modelo. Para saber mais sobre pilhas aninhadas, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).

CloudFormation O oferece suporte a apenas um nível de aninhamento usando `resource import`. Isso significa que você não pode importar uma pilha em uma pilha filho ou importar uma pilha com filhos.

Se você não tiver familiaridade com a importação, recomendamos que primeiro revise as informações introdutórias no tópico [Importar recursos da AWS para uma pilha do CloudFormation manualmente](#).

Validação da importação da pilha aninhada

Durante uma operação de importação de pilha aninhada, o CloudFormation realiza as validações a seguir.

- A definição aninhada de `AWS::CloudFormation::Stack` no modelo de pilha pai corresponde ao modelo da pilha real aninhada.
- As tags da definição aninhada de `AWS::CloudFormation::Stack` no modelo de pilha pai correspondem às tags do recurso de pilha real aninhada.

Aninhar uma pilha existente usando o . Console de gerenciamento da AWS

1. Adicione o recurso `AWS::CloudFormation::Stack` ao modelo de pilha pai com uma [DeletionPolicy](#) Retain. No modelo de exemplo de pilha principal, apresentado a seguir, `MyNestedStack` corresponde ao destino da importação.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "KeySchema":[
          {
            "AttributeName":"key",
            "KeyType":"HASH"
          }
        ],
        "ProvisionedThroughput":{
          "ReadCapacityUnits":5,
          "WriteCapacityUnits":1
        }
      }
    }
  }
}
```

```

    }
  },
  "MyNestedStack" : {
    "Type" : "AWS::CloudFormation::Stack",
    "DeletionPolicy": "Retain",
    "Properties" : {
      "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
      "Parameters" : {
        "InstanceType" : "t1.micro",
        "KeyName" : "mykey"
      }
    }
  }
}
}
}
}

```

YAML

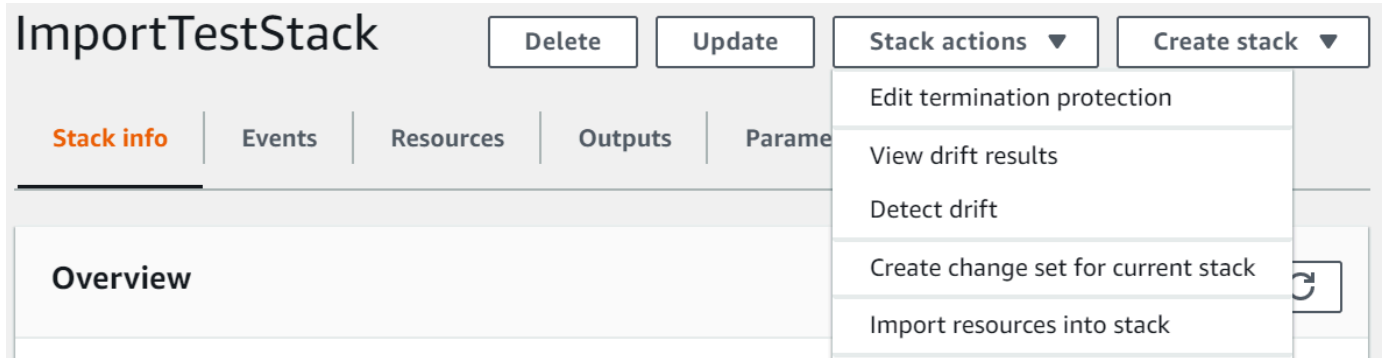
```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  MyNestedStack:
    Type: AWS::CloudFormation::Stack
    DeletionPolicy: Retain
    Properties:
      TemplateURL: >-
        https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template
      Parameters:
        InstanceType: t1.micro

```

KeyName: mykey

2. Abra o CloudFormation Console.
3. Na página Stacks (Pilhas), com a pilha pai selecionada, escolha Stack actions (Ações da pilha) e Import resources into stack (Importar recursos para a pilha).



4. Leia a página Import overview (Visão geral da importação) para obter uma lista de itens que você deve fornecer durante esta operação. Em seguida, escolha Próximo.
5. Na página Specify template (Especificar modelo), forneça o modelo pai atualizado usando um dos métodos a seguir e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.
6. Na página Identify resources (Identificar recursos), identifique o recurso `AWS::CloudFormation::Stack`.
 - a. Em Identifier property (Propriedade do identificador), escolha o tipo de identificador do recurso. Por exemplo, um recurso `AWS::CloudFormation::Stack` pode ser identificado usando a propriedade `StackId`.
 - b. Em Valor do identificador, digite o ARN da pilha que você está importando. Por exemplo, *`arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10`*.

Resources to import (1)

The following logical IDs are new to your template. Provide the identifier values for the resource you are importing.

NestedStack

AWS::CloudFormation::Stack

Identifier property

StackId

Identifier value

Enter StackId

c. Escolha Próximo.

7. Na página Specify stack details (Especificar detalhes da pilha), modifique os parâmetros e escolha Next (Próximo). Isso cria automaticamente um conjunto de alterações.

Important

A operação de importação falhará se você modificar os parâmetros existentes que iniciam uma operação de criação, atualização ou exclusão.

8. Na página Analisar **MyParentStack**, confirme se o recurso correto está sendo importado e, em seguida, escolha Importar recursos. Isso executa automaticamente o conjunto de alterações criado na última etapa. As tags no nível da pilha são aplicadas aos recursos importados no momento.
9. O painel Events (Eventos) da página Stack details (Detalhes da pilha) de sua pilha pai será exibido.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

🔍 Search events

🔄

⚙️

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	🟢 IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🟢 UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	🟢 IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	🔄 IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	🔄 IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	🔄 REVIEW_IN_PROGRESS	User Initiated

Note

Não é necessário executar a detecção de desvios na pilha pai após a operação de importação porque o recurso `AWS::CloudFormation::Stack` já foi gerenciado pelo CloudFormation.

Aninhar uma pilha existente usando o . AWS CLI

- Adicione o recurso `AWS::CloudFormation::Stack` ao modelo de pilha pai com uma [DeletionPolicy](#) Retain. No modelo pai do exemplo a seguir, MyNestedStack é o destino da importação.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ]
      }
    }
  }
}
```

```

    ],
    "KeySchema":[
        {
            "AttributeName":"key",
            "KeyType":"HASH"
        }
    ],
    "ProvisionedThroughput":{
        "ReadCapacityUnits":5,
        "WriteCapacityUnits":1
    }
},
"MyNestedStack" : {
    "Type" : "AWS::CloudFormation::Stack",
    "DeletionPolicy": "Retain",
    "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
        "Parameters" : {
            "InstanceType" : "t1.micro",
            "KeyName" : "mykey"
        }
    }
}
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:

```

```

    ReadCapacityUnits: 5
    WriteCapacityUnits: 1
    MyNestedStack:
      Type: AWS::CloudFormation::Stack
      DeletionPolicy: Retain
      Properties:
        TemplateURL: >-
          https://s3.amazonaws.com/cloudformation-templates-us-east-2/
          EC2ChooseAMI.template
      Parameters:
        InstanceType: t1.micro
        KeyName: mykey

```

2. Componha uma string JSON, conforme mostrado no exemplo a seguir, com estas modificações:

- Substitua *MyNestedStack* pelo ID lógico do recurso de destino, conforme especificado no modelo.
- Substitua *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10* pelo ARN da pilha que você deseja importar.

```

[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","Resource
{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}]}]

```

Como alternativa, é possível especificar os parâmetros em um arquivo de configuração.

Por exemplo, para importar *MyNestedStack*, é possível criar um arquivo *ResourcesToImport.txt* que contenha a configuração apresentada a seguir.

JSON

```

[
  {
    "ResourceType":"AWS::CloudFormation::Stack",
    "LogicalResourceId":"MyNestedStack",
    "ResourceIdentifier": {
      "StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/
mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"
    }
  }
]

```

]

YAML

```

ResourceType: AWS::CloudFormation::Stack
  LogicalResourceId: MyNestedStack
  ResourceIdentifier:
    StackId: >-
      arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10

```

3. Para criar um conjunto de alterações, use o comando `create-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. Para a opção, `--change-set-type` especifique um valor de **IMPORT**. Para a opção `--resources-to-import`, substitua a string JSON de exemplo pela string JSON real que você acabou de criar.

```

aws cloudformation create-change-set \
  --stack-name MyParentStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-
import '[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack",
{"StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]'

```

 Note

A opção `--resources-to-import` não oferece suporte para a linguagem YAML incorporada. Os requisitos para usar sequências de escape de aspas em uma string JSON variam de acordo com o terminal utilizado. Para obter mais informações, consulte [Using quotation marks inside strings](#) no Guia do usuário da AWS Command Line Interface.

Como alternativa, é possível usar um URL de arquivo como entrada para a opção `--resources-to-import`, conforme mostrado no exemplo a seguir.

```

--resources-to-import file://ResourcesToImport.txt

```

Se a detecção ocorrer com êxito, esse comando retornará o exemplo de saída apresentado a seguir.

```
{
  "Id": "arn:aws:cloudformation:us-west-2:12345678910:changeSet/
ImportChangeSet/8ad75b3f-665f-46f6-a200-0b4727a9442e",
  "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
MyParentStack/4e345b70-1281-11ef-b027-027366d8e82b"
}
```

4. Revise o conjunto de alterações para garantir que a pilha correta está sendo importada.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Para iniciar o conjunto de alterações e importar a pilha para a pilha principal de origem, use o comando `execute-change-set`, apresentado a seguir, e substitua o texto do espaço reservado. As [tags no nível da pilha](#) são aplicadas aos recursos importados no momento. Após concluir a operação de importação (`IMPORT_COMPLETE`), a pilha será aninhada com êxito.

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

Note

Não é necessário executar a detecção de desvio na pilha pai após essa operação de importação, pois o recurso `AWS::CloudFormation::Stack` já é gerenciado pelo CloudFormation.

Importar recursos da AWS para uma pilha do CloudFormation automaticamente

Agora, é possível importar recursos nomeados automaticamente ao criar ou atualizar pilhas do CloudFormation. Um recurso nomeado é um recurso com nome personalizado. Para obter mais informações, consulte [Tipo de nome](#) na Referência de modelos do CloudFormation.

Quando a importação automática é iniciada, o CloudFormation verifica os recursos existentes que correspondem ao modelo e os importa durante a implantação. No caso de pilhas aninhadas, crie o conjunto de alterações a partir da pilha raiz.

Após a importação ter sido concluída, e antes de realizar as operações de pilhas subsequentes, recomendamos executar a detecção de desvios nos recursos importados. A detecção de desvios garante que a configuração do modelo corresponda à configuração real. Para obter mais informações, consulte [Detectar desvio em uma pilha inteira do CloudFormation](#).

Para importar um recurso, é necessário que ele atenda aos seguintes requisitos:

- O recurso deve ter um nome personalizado estático definido no modelo. No momento, não há suporte para nomes dinâmicos (usando `!Ref` ou outras funções).
- O recurso deve ter `DeletionPolicy` de `Retain` ou `RetainExceptOnCreate`.
- O recurso não deve pertencer a outra pilha do CloudFormation.
- O tipo de recurso deve ser compatível com operações de importação do CloudFormation. Para obter mais informações, consulte [Suporte a tipos de recursos](#).
- O ID primário do tipo de recurso deve estar no modelo. Não há suporte a IDs primários com propriedades somente leitura. Para descobrir qual é o ID primário de um tipo, procure a propriedade `primaryIdentifier` no esquema do recurso. Para obter mais informações sobre a propriedade, consulte [primaryIdentifier](#).

Exemplo Exemplo de importação automática

O exemplo a seguir usa um conjunto de alterações, `CreateChangeSet`, para criar uma pilha chamada `my-stack` com base em um arquivo de modelo, `template.yaml`, e importa os recursos correspondentes de maneira automática.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name CreateChangeSet \  
  --change-set-type CREATE \  
  --template-body file:///template.yaml \  
  --import-existing-resources
```

Solução de problemas

Se a importação automática falhar, faça o seguinte para solucionar o problema:

- Verifique se o nome do recurso no modelo corresponde exatamente ao nome do recurso
- Verifique se o recurso não está sendo gerenciado por outra pilha
- Certifique-se de que o tipo de recurso ofereça suporte às operações de importação

- Verifique se o modelo inclui todas as propriedades necessárias para o tipo de recurso

Reverter uma operação de importação

Para reverter uma operação de importação, especifique uma política de exclusão `Retain` para o recurso que você deseja remover do modelo para garantir que ele seja preservado ao excluí-lo da pilha.

Reverter uma operação de importação usando a Console de gerenciamento da AWS

1. Especifique uma [DeletionPolicy](#) `Retain` para os recursos que você deseja remover da pilha. No modelo de exemplo a seguir, `GameTable` é o destino desta operação de reversão.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GameTable": {
```

```
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
      "TableName": "Games",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}
```

2. Abra o console do CloudFormation para executar uma atualização de pilha para aplicar a política de exclusão.
 - a. Na página Stacks (Pilhas) com a pilha selecionada, escolha Update (Atualizar) e selecione Update stack (standard) (Atualizar pilha (padrão)).
 - b. Em Preparar modelo, escolha Substituir modelo atual.
 - c. Em Specify template (Especificar modelo), forneça o modelo de origem atualizado com o atributo DeletionPolicy na GamesTable e escolha Next (Próximo).
 - Escolha Amazon S3 URL (URL do Amazon S3) e especifique o URL para o modelo de origem atualizado na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo de origem atualizado.
 - d. Na página Specify stack details (Especificar detalhes da pilha) nenhuma alteração é necessária. Escolha Próximo.

- e. Na página Configure stack options (Configurar opções de pilha) nenhuma alteração é necessária. Escolha Próximo.
 - f. Na página Analisar **MyStack**, analise as alterações. Se o seu modelo contém recursos do IAM, selecione I acknowledge that this template may create IAM resources (Eu reconheço que este modelo pode criar recursos do IAM) para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#). Depois, atualize sua pilha de origem criando um conjunto de alterações ou atualize sua pilha de origem diretamente.
3. Remova o recurso, os parâmetros relacionados e as saídas do modelo de pilha. Neste exemplo, o modelo agora se parece com o seguinte.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

```
}
```

4. Repita a etapa 2 para excluir o recurso (GameSTable) e os parâmetros relacionados e as saídas da pilha.

Reverter uma operação de importação usando a AWS CLI

1. Especifique uma [DeletionPolicy](#) Retain para os recursos que você deseja remover da pilha. No modelo de exemplo a seguir, GameSTable é o destino desta operação de reversão.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GameSTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
```

```

        "TableName": "Games",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}
}
}
}

```

2. Atualize a pilha para aplicar a política de exclusão ao recurso.

```
aws cloudformation update-stack --stack-name MyStack
```

3. Remova o recurso, os parâmetros relacionados e as saídas do modelo de pilha. Neste exemplo, o modelo agora se parece com o seguinte.

Example JSON

```

{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Import test",
    "Resources": {
        "ServiceTable": {
            "Type": "AWS::DynamoDB::Table",
            "Properties": {
                "TableName": "Service",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ]
            }
        }
    }
}

```

```
    }
  ],
  "KeySchema":[
    {
      "AttributeName":"key",
      "KeyType":"HASH"
    }
  ],
  "ProvisionedThroughput":{
    "ReadCapacityUnits":5,
    "WriteCapacityUnits":1
  }
}
}
```

4. Atualize a pilha para excluir o recurso (GameTable) e os parâmetros relacionados, além das saídas da pilha.

```
aws cloudformation update-stack --stack-name MyStack
```

Refatoração de pilhas

Com a refatoração de pilhas, é possível reorganizar os recursos em suas pilhas do CloudFormation enquanto preserva as propriedades e os dados dos recursos existentes. É possível mover recursos entre pilhas, dividir pilhas grandes em pilhas menores ou combinar várias pilhas em uma única pilha.

Tópicos

- [Como a refatoração de pilhas funciona](#)
- [Considerações sobre refatoração de pilhas](#)
- [Pré-requisitos](#)
- [Refatorar pilhas \(console\)](#)
- [Refatorar pilhas \(AWS CLI\)](#)
- [Limitações do recurso](#)

Como a refatoração de pilhas funciona

A refatoração de pilhas envolve as seguintes fases:

1. Avaliação da infraestrutura atual: analise suas pilhas e recursos existentes do CloudFormation para identificar oportunidades de refatoração de pilhas.
2. Planejamento da refatoração: defina como os recursos devem ser organizados. Considere suas dependências, convenções de nomenclatura e limites operacionais. Isso pode afetar a validação do CloudFormation posteriormente.
3. Determinação das pilhas de destino: decida em quais pilhas você refatorará os recursos. Você pode mover recursos entre pelo menos duas pilhas (usando o console) e no máximo cinco pilhas (usando a AWS CLI). Os recursos podem ser movidos entre pilhas aninhadas.
4. Atualização de modelos: altere seus modelos do CloudFormation para refletir a alteração planejada, como mover definições de recursos entre modelos. É possível renomear IDs lógicos durante esse processo.
5. Criação da refatoração de pilhas: forneça uma lista dos nomes e modelos da pilha que você deseja refatorar.
6. Análise do impacto da refatoração em sua infraestrutura e resolução de quaisquer conflitos: o CloudFormation valida os modelos fornecidos por você e verifica dependências entre pilhas, tipos de recursos com problemas de atualização de tags e conflitos de ID lógico de recursos.

Se a validação for bem-sucedida, o CloudFormation gerará uma prévia das ações de refatoração que ocorrerão após a execução.

Se a validação falhar, resolva os problemas identificados e tente novamente. Em caso de conflitos, forneça um mapeamento de ID lógico de recurso que mostre a origem e o destino dos recursos conflitantes.

7. Execução da refatoração: após confirmar se as alterações estão alinhadas com suas metas de refatoração, conclua a refatoração da pilha.
8. Monitoramento: acompanhe o status da execução para garantir que a operação seja concluída com êxito.

Considerações sobre refatoração de pilhas

Ao refatorar suas pilhas, lembre-se do seguinte:

- A refatoração da pilha se limita à reorganização dos recursos existentes. Você não pode criar ou excluir recursos, modificar configurações de recursos ou alterar ou adicionar novos parâmetros, condições ou mapeamentos durante a refatoração. Para fazer essas alterações, primeiro atualize suas pilhas e, em seguida, realize a refatoração da pilha.
- Não é possível refatorar o mesmo recurso em várias pilhas.
- Você não pode refatorar recursos que se referem a pseudoparâmetros cujos valores diferem entre as pilhas de origem e destino, como `AWS::StackName`.
- O CloudFormation não oferece suporte a pilhas vazias. Se a refatoração deixar uma pilha sem recursos, você deve primeiro adicionar pelo menos um recurso a essa pilha antes de executar o [create-stack-refactor](#). Isso pode ser um recurso simples, como `AWS::SNS::Topic` ou `AWS::CloudFormation::WaitCondition`. Por exemplo:

```
Resources:
  MySimpleSNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: MySimpleTopic
```

- A refatoração de pilha não é compatível com pilhas que tenham políticas de pilha anexadas, independentemente do que as políticas permitam ou neguem.

Pré-requisitos

Para refatoração de pilhas, você já deve ter criado os modelos revisados.

Use o comando [get-template](#) para recuperar os modelos do CloudFormation para as pilhas que você deseja refatorar.

```
aws cloudformation get-template --stack-name Stack1
```

Quando tiver os modelos, use o ambiente de desenvolvimento integrado (IDE) de sua escolha para atualizá-los para usar a estrutura e a organização de recursos desejadas.

Refatorar pilhas (console)

Use o procedimento a seguir para refatorar uma pilha usando o console.

Para refatorar pilhas

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que as pilhas estão localizadas.
3. No painel de navegação à esquerda, escolha Refatorações de pilha.
4. Na página Refatorações de pilha, escolha Iniciar refatoração de pilha.
5. Em Descrição, forneça uma descrição para ajudar a identificar a refatoração da pilha. Em seguida, escolha Próximo.
6. Na Pilha 1, faça o seguinte:

- a. Escolha Atualizar o modelo para uma pilha existente ou Criar nova pilha.

Se você escolher Atualizar o modelo para uma pilha existente, selecione uma pilha existente na lista. Ou escolha Inserir um ARN de pilha para inserir o ARN de uma pilha existente.

Se você escolher Criar nova pilha, em Nome da pilha, forneça um nome para a nova pilha.

- b. Em Substituir modelo existente por modelo refatorado, escolha URL do Amazon S3 ou Fazer upload de um arquivo de modelo para carregar o modelo desejado para a Pilha 1.
 - c. Escolha Próximo.
7. Na Pilha 2, faça o seguinte:

- a. Escolha Atualizar o modelo para uma pilha existente ou Criar nova pilha.

Se você escolher Atualizar o modelo para uma pilha existente, selecione uma pilha existente na lista. Ou escolha Inserir um ARN de pilha para inserir o ARN de uma pilha existente.

Se você escolher Criar nova pilha, em Nome da pilha, forneça um nome para a nova pilha.

- b. Em Substituir modelo existente por modelo refatorado, escolha URL do Amazon S3 ou Fazer upload de um arquivo de modelo para carregar o modelo desejado para a Pilha 2.
 - c. Escolha Próximo.
8. Na página Especificar renomeações de IDs de recursos lógicos, certifique-se de que o CloudFormation saiba como refatorar pilhas mapeando todos os recursos mostrados para seus IDs lógicos corretos. Como parte da refatoração da pilha, se os IDs lógicos de algum recurso

forem alterados, você precisará especificar como ele foi renomeado fornecendo o nome da pilha de origem, o ID lógico original, o nome da pilha de destino e o ID lógico renomeado. Em alguns casos, o console do CloudFormation pode detectar automaticamente o mapeamento de recursos, e você pode simplesmente verificar se o mapeamento de recursos pré-preenchido está correto antes de continuar.

9. Escolha Próximo.
10. Na página Revisar e executar, revise todas as seleções das etapas anteriores e confirme que tudo está configurado corretamente.
11. Quando tudo estiver pronto para refatorar as pilhas, escolha Executar refatoração da pilha.

Refatorar pilhas (AWS CLI)

Os comandos da AWS CLI para refatoração de pilha incluem:

- [create-stack-refactor](#) para validar e gerar uma prévia das alterações planejadas.
- [describe-stack-refactor](#) para recuperar o status e os detalhes de uma operação de refatoração de pilha.
- [execute-stack-refactor](#) para concluir a operação validada de refatoração da pilha.
- [list-stack-refactors](#) para listar todas as operações de refatoração de pilha em sua conta com seu status atual e informações básicas.
- [list-stack-refactor-actions](#) para mostrar uma prévia das ações específicas que o CloudFormation executará em cada pilha e recurso durante a execução da refatoração.

Use o procedimento a seguir para refatorar uma pilha usando a AWS CLI.

Para refatorar pilhas

1. Use o comando [create-stack-refactor](#) e forneça os nomes das pilhas e os modelos atualizados para que as pilhas sejam refatoradas. Inclua a opção `--enable-stack-creation` para permitir que o CloudFormation crie novas pilhas se elas ainda não existirem.

```
aws cloudformation create-stack-refactor \
  --stack-definitions \
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \
  --enable-stack-creation
```


O comando retorna um `StackRefactorId` que você usará em etapas posteriores.

```
{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841"
}
```

Se forem detectados conflitos durante a validação do modelo (o que você pode confirmar na próxima etapa), use o comando [create-stack-refactor](#) com a opção `--resource-mappings`.

```
aws cloudformation create-stack-refactor \
  --stack-definitions \
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \
  --enable-stack-creation \
  --resource-mappings file://resource-mapping.json
```

Veja a seguir um exemplo de arquivo `resource-mapping.json`.

```
[
  {
    "Source": {
      "StackName": "Stack1",
      "LogicalResourceId": "MySNSTopic"
    },
    "Destination": {
      "StackName": "Stack2",
      "LogicalResourceId": "MyLambdaSNSTopic"
    }
  }
]
```

2. Use o comando [describe-stack-refactor](#) para verificar se `Status` é `CREATE_COMPLETE`. Isso verifica se a validação foi concluída.

```
aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Resultado do exemplo:

```
{
```

```

    "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
    "StackIds": [
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
    ],
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE"
}

```

3. Use o comando [list-stack-refactor-actions](#) para visualizar as ações específicas que serão executadas.

```

aws cloudformation list-stack-refactor-actions \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

Resultado do exemplo:

```

{
  "StackRefactorActions": [
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestLambdaRole",
      "Description": "No configuration changes detected.",
      "Detection": "AUTO",
      "TagResources": [],
      "UntagResources": [],
      "ResourceMapping": {
        "Source": {
          "StackName": "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
          "LogicalResourceId": "MyLambdaRole"
        },
        "Destination": {
          "StackName": "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
          "LogicalResourceId": "MyLambdaRole"
        }
      }
    },
    {

```

```

        "Action": "MOVE",
        "Entity": "RESOURCE",
        "PhysicalResourceId": "MyTestFunction",
        "Description": "Resource configuration changes will be validated during
refactor execution.",
        "Detection": "AUTO",
        "TagResources": [
            {
                "Key": "aws:cloudformation:stack-name",
                "Value": "Stack2"
            },
            {
                "Key": "aws:cloudformation:logical-id",
                "Value": "MyFunction"
            },
            {
                "Key": "aws:cloudformation:stack-id",
                "Value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
            }
        ],
        "UntagResources": [
            "aws:cloudformation:stack-name",
            "aws:cloudformation:logical-id",
            "aws:cloudformation:stack-id"
        ],
        "ResourceMapping": {
            "Source": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
                "LogicalResourceId": "MyFunction"
            },
            "Destination": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
                "LogicalResourceId": "MyFunction"
            }
        }
    }
}
]
}

```

4. Depois de revisar e confirmar suas alterações, use o comando [execute-stack-refactor](#) para concluir a operação de refatoração da pilha.

```
aws cloudformation execute-stack-refactor \  
--stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

5. Use o comando [describe-stack-refactor](#) para monitorar o status da execução.

```
aws cloudformation describe-stack-refactor \  
--stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Resultado do exemplo:

```
{  
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",  
  "StackIds": [  
    "arn:aws:cloudformation:us-east-1:123456789012:stack/  
Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",  
    "arn:aws:cloudformation:us-east-1:123456789012:stack/  
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"  
  ],  
  "ExecutionStatus": "SUCCEEDED",  
  "Status": "COMPLETE"  
}
```

Limitações do recurso

- A refatoração de pilha só suporta tipos de recursos com um provisioningType de FULLY_MUTABLE, que você pode verificar usando o comando [describe-type](#).
- O CloudFormation validará a qualificação do recurso durante a criação da refatoração e relatará quaisquer recursos não suportados na saída do comando [describe-stack-refactor](#).
- Os seguintes recursos não são compatíveis com refatoração de pilhas:
 - AWS::ACMPCA::Certificate
 - AWS::ACMPCA::CertificateAuthority
 - AWS::ACMPCA::CertificateAuthorityActivation
 - AWS::ApiGateway::BasePathMapping
 - AWS::ApiGateway::Method
 - AWS::AppConfig::ConfigurationProfile

- `AWS::AppConfig::Deployment`
- `AWS::AppConfig::Environment`
- `AWS::AppConfig::Extension`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::DirectoryConfig`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::BackupGateway::Hypervisor`
- `AWS::CertificateManager::Certificate`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodePipeline::CustomActionType`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::ConfigRule`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::DataBrew::Dataset`
- `AWS::DataBrew::Job`
- `AWS::DataBrew::Project`
- `AWS::DataBrew::Recipe`
- `AWS::DataBrew::Ruleset`
- `AWS::DataBrew::Schedule`
- `AWS::DataZone::DataSource`
- `AWS::DataZone::Environment`

- `AWS::DataZone::EnvironmentBlueprintConfiguration`
- `AWS::DataZone::EnvironmentProfile`
- `AWS::DataZone::Project`
- `AWS::DataZone::SubscriptionTarget`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DynamoDB::GlobalTable`
- `AWS::EC2::CustomerGateway`
- `AWS::EC2::EIP`
- `AWS::EC2::LaunchTemplate`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::EC2::PlacementGroup`
- `AWS::EC2::SpotFleet`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VolumeAttachment`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticBeanstalk::ConfigurationTemplate`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::FIS::ExperimentTemplate`
- `AWS::Glue::Schema`
- `AWS::GuardDuty::IPSet`
- `AWS::GuardDuty::PublishingDestination`
- `AWS::GuardDuty::ThreatIntelSet`

- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::ImageBuilder::Component`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTFleetWise::Campaign`
- `AWS::IoTWireless::WirelessDeviceImportTask`
- `AWS::Lambda::EventInvokeConfig`
- `AWS::Lex::BotVersion`
- `AWS::M2::Application`
- `AWS::MSK::Configuration`
- `AWS::MSK::ServerlessCluster`
- `AWS::Maester::DocumentType`
- `AWS::MediaTailor::Channel`
- `AWS::NeptuneGraph::PrivateGraphEndpoint`
- `AWS::Omics::AnnotationStore`
- `AWS::Omics::ReferenceStore`
- `AWS::Omics::SequenceStore`
- `AWS::OpenSearchServerless::Collection`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`

- `AWS::Panorama::PackageVersion`
- `AWS::QuickSight::Theme`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::RefactorSpaces::Environment`
- `AWS::RefactorSpaces::Route`
- `AWS::RefactorSpaces::Service`
- `AWS::RoboMaker::RobotApplication`
- `AWS::RoboMaker::SimulationApplication`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SDB::Domain`
- `AWS::SageMaker::InferenceComponent`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::PortfolioShare`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::StepFunctions::StateMachineVersion`
- `AWS::Synthetics::Canary`
- `AWS::VoiceID::Domain`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`

- `AWS::WAF::XssMatchSet`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`
- `AWS::WorkSpaces::Workspace`

Suporte a tipos de recursos

A tabela a seguir lista os tipos de recursos da AWS que são compatíveis com importação, detecção de desvios e operações do gerador de infraestrutura como código (IaC). Cada nome de tipo de recurso está vinculado ao tópico de referência correspondente no [Guia de referência de modelos do CloudFormation](#).

Um recurso que é compatível com importação de recursos também pode ser compatível com importação automática. Para obter mais informações, consulte [Importar recursos da AWS para uma pilha do CloudFormation](#).

Essa lista não esgota todos os tipos de recursos da AWS. Se um tipo de recurso específico não estiver listado, provavelmente significa que ele não pode ser acessado por meio do AWS API Cloud Control. Para obter mais informações, consulte [Resource types that support Cloud Control API](#) no Cloud Control API User Guide. Cada serviço da AWS determina de forma independente quais tipos de recursos devem ser disponibilizados por meio da API do Cloud Control.

O CloudFormation também oferece suporte a operações de importação e detecção de desvios para tipos de recursos privados que são provisionáveis (aqueles com tipos de provisionamento `FULLY_MUTABLE` ou `IMMUTABLE`). Para importar ou realizar a detecção de desvios em um tipo de recurso privado, é necessário primeiro registrar o versionamento padrão desse tipo de recurso na sua conta e garantir que ele seja provisionável. Para obter mais informações, consulte [Use extensões privadas de terceiros que foram compartilhadas com você](#).

Observe que o gerador de IaC somente oferece suporte a recursos da AWS compatíveis com a API do Cloud Control na sua região.

Para acessar programaticamente as informações sobre os tipos de recursos provisionáveis públicos e privados, é possível usar o AWS API Cloud Control. Para obter mais informações, consulte [Como](#)

determinar se um tipo de recurso é compatível com o Cloud Control API no Guia do usuário do Cloud Control API

Aqui estão alguns tópicos introdutórios úteis sobre importação, detecção de desvios ou gerador de IaC:

- [Importar recursos da AWS para uma pilha do CloudFormation](#)
- [Detectar alterações de configuração não gerenciadas em pilhas e recursos com detecção de desvios](#)
- [Gerar modelos a partir de recursos existentes com o gerador de IaC](#)

Recurso	Importar	Detecção de desvios	Gerador de IaC
AWS::ACMPCA::Certificate	 Sim	 No (Não)	 No (Não)
AWS::ACMPCA::CertificateAuthority	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::ACMPCA::CertificateAuthorityActivation	 Yes (Sim)	 No (Não)	 No (Não)
AWS::ACMPCA::Permission	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AIOps::InvestigationGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::APS::AnomalyDetector</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::APS::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::APS::RuleGroupsNamespace</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::APS::Scraper</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::APS::Workspace</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ARCRegionSwitch::Plan</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ARCZonalShift::AutoshiftObserveNotificationStatus</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ARCZonalShift::ZonalAutoshiftConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AccessAnalyzer::Analyzer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AmazonMQ::Broker</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AmazonMQ::Configuration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Amplify::App</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Amplify::Branch</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Amplify::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AmplifyUIBuilder::Component</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AmplifyUIBuilder::Form</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AmplifyUIBuilder::Theme</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::Account</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::ApiKey</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::Authorizer</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApiGateway::BasePathMapping</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::BasePathMappingV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::ClientCertificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::DocumentationPart</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::DocumentationVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::DomainName</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::DomainNameV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::GatewayResponse</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::Method</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::Model</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::RequestValidator</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGateway::Resource</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApiGateway::RestApi</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::Stage</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::ApiGateway::UsagePlan</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGateway::UsagePlanKey</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::ApiGateway::VpcLink</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::Api</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGatewayV2::ApiMapping</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApiGatewayV2::Authorizer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGatewayV2::DomainName</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApiGatewayV2::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::Model</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::Route</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApiGatewayV2::RouteResponse</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::RoutingRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApiGatewayV2::VpcLink</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppConfig::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppConfig::ConfigurationProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppConfig::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppConfig::DeploymentStrategy</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AppConfig::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppConfig::Extension</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppConfig::ExtensionAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppFlow::Connector</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppFlow::ConnectorProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppFlow::Flow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AppIntegrations::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppIntegrations::DataIntegration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppIntegrations::EventIntegration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppRunner::Service</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppRunner::VpcConnector</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AppRunner::VpcIngressConnection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::AppBlock</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::AppBlockBuilder</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::DirectoryConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AppStream::Entitlement</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppStream::ImageBuilder</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppSync::Api</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppSync::ChannelNamespace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppSync::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppSync::DomainName</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppSync::DomainNameApiAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AppSync::FunctionConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 No (Não)
<u>AWS::AppSync::GraphQLApi</u>	 No (Não)	 Sim	 Não
<u>AWS::AppSync::Resolver</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::AppSync::SourceApiAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AppTest::TestCase</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 Yes (Sim)	 No (Não)	 No (Não)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ApplicationInsights::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApplicationSignals::Discovery</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApplicationSignals::GroupingCon</u> <u>figuration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ApplicationSignals::ServiceLeve</u> <u>lObjective</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Athena::CapacityReservation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Athena::DataCatalog</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Athena::NamedQuery</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Athena::PreparedStatement</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Athena::WorkGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AuditManager::Assessment</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::AutoScaling::AutoScalingGroup</u>	 Sim	 Não	 Yes (Sim)
<u>AWS::AutoScaling::LaunchConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AutoScaling::LifecycleHook</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AutoScaling::ScalingPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::AutoScaling::ScheduledAction</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::AutoScaling::WarmPool</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::B2BI::Capability</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::B2BI::Partnership</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::B2BI::Profile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::B2BI::Transformer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BCMDDataExports::Export</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Backup::BackupPlan</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Backup::BackupSelection</u>	 Sim	 Não	 Yes (Sim)
<u>AWS::Backup::BackupVault</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Backup::Framework</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Backup::ReportPlan</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Backup::RestoreTestingPlan</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Backup::RestoreTestingSelection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BackupGateway::Hypervisor</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Batch::ComputeEnvironment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Batch::ConsumableResource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Batch::JobDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Batch::JobQueue</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Batch::SchedulingPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Batch::ServiceEnvironment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Bedrock::Agent</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::AgentAlias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::AutomatedReasoningPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Bedrock::AutomatedReasoningPolicyVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Bedrock::Blueprint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Bedrock::DataAutomationProject</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::Flow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::FlowAlias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::FlowVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::Guardrail</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::GuardrailVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Bedrock::IntelligentPromptRouter</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::KnowledgeBase</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::Bedrock::Prompt</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Bedrock::PromptVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::BedrockAgentCore::BrowserCustom</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::CodeInterpreterCustom</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::Gateway</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::BedrockAgentCore::GatewayTarget</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::Memory</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::Runtime</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::RuntimeEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BedrockAgentCore::WorkloadIdentity</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Billing::BillingView</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::BillingConductor::BillingGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::BillingConductor::CustomLineItem</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::BillingConductor::PricingPlan</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::BillingConductor::PricingRule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Budgets::BudgetsAction</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CE::AnomalyMonitor</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CE::AnomalySubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CE::CostCategory</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CUR::ReportDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cases::CaseRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cases::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cases::Field</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cases::Layout</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cases::Template</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cassandra::Keyspace</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Cassandra::Table</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Cassandra::Type</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CertificateManager::Account</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Chatbot::CustomAction</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Chatbot::MicrosoftTeamsChannelConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Chatbot::SlackChannelConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CleanRooms::AnalysisTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CleanRooms::Collaboration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CleanRooms::ConfiguredTable</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CleanRooms::IdMappingTable</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CleanRooms::Membership</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CleanRoomsML::TrainingDataset</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::GuardHook</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::HookDefaultVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFormation::HookTypeConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFormation::HookVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFormation::LambdaHook</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudFormation::ModuleVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::PublicTypeVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::Publisher</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::ResourceVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::Stack</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFormation::StackSet</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudFormation::TypeActivation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::AnycastIpList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::CachePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::ConnectionFunction</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::ConnectionGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::ContinuousDeploymentPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudFront::Distribution</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::DistributionTenant</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::Function</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::KeyGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::KeyValueStore</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::MonitoringSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::OriginAccessControl</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudFront::OriginRequestPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::PublicKey</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::RealtimeLogConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::TrustStore</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudFront::VpcOrigin</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudTrail::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudTrail::Dashboard</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudTrail::EventDataStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudTrail::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CloudTrail::Trail</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudWatch::Alarm</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudWatch::CompositeAlarm</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CloudWatch::Dashboard</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CloudWatch::MetricStream</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeArtifact::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CodeArtifact::PackageGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeArtifact::Repository</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CodeBuild::Fleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeConnections::Connection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeDeploy::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CodeDeploy::DeploymentConfig</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodePipeline::CustomActionType</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodePipeline::Pipeline</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodePipeline::Webhook</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CodeStarConnections::Connection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
AWS::CodeStarConnections::RepositoryLink	 Yes (Sim)	 Yes (Sim)	 Não
AWS::CodeStarConnections::SyncConfiguration	 Yes (Sim)	 Yes (Sim)	 Não
AWS::CodeStarNotifications::NotificationRule	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::Cognito::IdentityPool	 Yes (Sim)	 Yes (Sim)	 Não
AWS::Cognito::IdentityPoolPrincipalTag	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::Cognito::IdentityPoolRoleAttachment	 Yes (Sim)	 Yes (Sim)	 Não
AWS::Cognito::LogDeliveryConfiguration	 Sim	 No (Não)	 No (Não)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Cognito::ManagedLoginBranding</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::Terms</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPool</u>	 Sim	 No (Não)	 No (Não)
<u>AWS::Cognito::UserPoolClient</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPoolDomain</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPoolGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Cognito::UserPoolResourceServer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Cognito::UserPoolUser</u>	 Sim	 No (Não)	 No (Não)
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Comprehend::DocumentClassifier</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Comprehend::Flywheel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Config::AggregationAuthorization</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Config::ConfigRule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Config::ConfigurationAggregator</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Config::ConformancePack</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Config::OrganizationConformancePack</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Config::StoredQuery</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::AgentStatus</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Connect::ApprovedOrigin</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::ContactFlow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::ContactFlowModule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::ContactFlowVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::DataTable</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Connect::DataTableAttribute</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Connect::DataTableRecord</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Connect::EmailAddress</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::EvaluationForm</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::HoursOfOperation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::Instance</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::InstanceStorageConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::IntegrationAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::PhoneNumber</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Connect::PredefinedAttribute</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::Prompt</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::Queue</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::QuickConnect</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::RoutingProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::Rule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Connect::SecurityKey</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Connect::SecurityProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::TaskTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::TrafficDistributionGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::User</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::UserHierarchyGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::UserHierarchyStructure</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Connect::View</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Connect::ViewVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Connect::Workspace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ConnectCampaigns::Campaign</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ConnectCampaignsV2::Campaign</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ControlTower::EnabledBaseline</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ControlTower::EnabledControl</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ControlTower::LandingZone</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CustomerProfiles::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CustomerProfiles::EventStream</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CustomerProfiles::EventTrigger</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::CustomerProfiles::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CustomerProfiles::ObjectType</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DMS::DataMigration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DMS::DataProvider</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DMS::InstanceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DMS::MigrationProject</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DMS::ReplicationConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DSQL::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataBrew::Dataset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataBrew::Job</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataBrew::Project</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataBrew::Recipe</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataBrew::Ruleset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataBrew::Schedule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataPipeline::Pipeline</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::Agent</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataSync::LocationAzureBlob</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationEFS</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationFSxLustre</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationFSxONTAP</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationFSxWindows</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationHDFS</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataSync::LocationNFS</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationObjectStorage</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationS3</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::LocationSMB</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataSync::Task</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DataZone::Connection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataZone::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::DomainUnit</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::EnvironmentActions</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::EnvironmentProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::FormType</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataZone::GroupProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::Owner</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::PolicyGrant</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::Project</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::ProjectMembership</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::ProjectProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DataZone::SubscriptionTarget</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DataZone::UserProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::Farm</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::Fleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::LicenseEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::Limit</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::MeteredProduct</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::Monitor</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Deadline::Queue</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::QueueEnvironment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::QueueFleetAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::QueueLimitAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Deadline::StorageProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Detective::Graph</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Detective::MemberInvitation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Detective::OrganizationAdmin</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DevOpsAgent::AgentSpace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DevOpsAgent::Association</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DevOpsGuru::LogAnomalyDetectionIntegration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DevOpsGuru::NotificationChannel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DevOpsGuru::ResourceCollection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DeviceFarm::DevicePool</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DeviceFarm::InstanceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DeviceFarm::NetworkProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DeviceFarm::Project</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DeviceFarm::TestGridProject</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DeviceFarm::VPCEConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DirectoryService::SimpleAD</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::DocDBElastic::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::DynamoDB::GlobalTable</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::DynamoDB::Table</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::CapacityManagerDataExport</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::CapacityReservation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::CapacityReservationFleet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::CarrierGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::CustomerGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::DHCPOptions</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::EC2Fleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::EIP</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::EIPAssociation</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::FlowLog</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::GatewayRouteTableAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::Host</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAM</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAMAllocation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAMPool</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAMPoolCidr</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAMResourceDiscovery</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::IPAMResourceDiscoveryAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IPAMScope</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::Instance</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::InstanceConnectEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::InternetGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::IpPoolRouteTableAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::KeyPair</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::LaunchTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::LocalGatewayRoute</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::LocalGatewayRouteTable</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::LocalGatewayVirtualInterface</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::LocalGatewayVirtualInterfaceGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::NatGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::NetworkAcl</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::NetworkInsightsAccessScopeAnalysis</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::NetworkInsightsPath</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::NetworkInterface</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::NetworkPerformanceMetricSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::PlacementGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::PrefixList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::Route</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::RouteServer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::RouteServerAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::RouteServerEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::RouteServerPeer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::RouteServerPropagation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::RouteTable</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::SecurityGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::SecurityGroupEgress</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::SecurityGroupIngress</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::SecurityGroupVpcAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::SpotFleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::Subnet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::SubnetCidrBlock</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::SubnetNetworkAclAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 Yes (Sim)	 Não	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::TrafficMirrorFilter</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TrafficMirrorSession</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TrafficMirrorTarget</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayConnect</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::TransitGatewayConnectPeer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayMeteringPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayMeteringPolicyEntry</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayMulticastGroupSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::TransitGatewayPeeringAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayRoute</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayRouteTable</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::TransitGatewayRouteTableAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayRouteTablePropagation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::TransitGatewayVpcAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPC</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::VPCLockPublicAccessExclusion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCLockPublicAccessOptions</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCCIDRBlock</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCDHCPOptionsAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPCEncryptionControl</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPCEndpointConnectionNotification</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EC2::VPCEndpointService</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPCEndpointServicePermissions</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCGatewayAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPCPeeringConnection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPNConcentrator</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EC2::VPNConnection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EC2::VPNConnectionRoute</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
AWS::EC2::VPNGateway	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::VerifiedAccessEndpoint	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::VerifiedAccessGroup	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::VerifiedAccessInstance	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::VerifiedAccessTrustProvider	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::Volume	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::EC2::VolumeAttachment	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ECR::PublicRepository</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECR::PullThroughCacheRule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECR::PullTimeUpdateExclusion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECR::RegistryPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECR::RegistryScanningConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECR::ReplicationConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECR::Repository</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ECR::RepositoryCreationTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECR::SigningConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECS::CapacityProvider</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECS::Cluster</u>	 Sim	 Não	 Yes (Sim)
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::ECS::ExpressGatewayService</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ECS::PrimaryTaskSet</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ECS::Service</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECS::TaskDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ECS::TaskSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EFS::AccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EFS::FileSystem</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EFS::MountTarget</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EKS::AccessEntry</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EKS::Addon</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EKS::Capability</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EKS::Cluster</u>	 Sim	 Não	 Yes (Sim)
<u>AWS::EKS::FargateProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EKS::IdentityProviderConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EKS::Nodegroup</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::EKS::PodIdentityAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EMR::SecurityConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EMR::Step</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::EMR::Studio</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EMR::StudioSessionMapping</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EMR::WALWorkspace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EMRContainers::VirtualCluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EMRServerless::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EVS::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ElastiCache::ParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ElastiCache::ServerlessCache</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ElastiCache::SubnetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElastiCache::User</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElastiCache::UserGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ElasticBeanstalk::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElasticBeanstalk::Environment</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EntityResolution::IdNamespace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EntityResolution::MatchingWorkflow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EntityResolution::PolicyStatement</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::EntityResolution::SchemaMapping</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::EventSchemas::Discoverer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EventSchemas::Registry</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EventSchemas::RegistryPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::EventSchemas::Schema</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Events::ApiDestination</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Events::Archive</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Events::Connection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Events::Endpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Events::EventBus</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Events::EventBusPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Events::Rule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Evidently::Experiment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Evidently::Feature</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Evidently::Launch</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Evidently::Project</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Evidently::Segment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FIS::ExperimentTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FIS::TargetAccountConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FMS::NotificationChannel</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FMS::Policy</u>	 Sim	 No (Não)	 No (Não)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::FMS::ResourceSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FSx::DataRepositoryAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FSx::S3AccessPointAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FinSpace::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Forecast::Dataset</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Forecast::DatasetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::FraudDetector::Detector</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::FraudDetector::EntityType</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FraudDetector::EventType</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FraudDetector::Label</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FraudDetector::List</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FraudDetector::Outcome</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::FraudDetector::Variable</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GameLift::Alias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::GameLift::Build</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GameLift::ContainerFleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::ContainerGroupDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::Fleet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GameLift::GameServerGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::GameSessionQueue</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::Location</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::GameLift::MatchmakingConfigurat ion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::MatchmakingRuleSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLift::Script</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLiftStreams::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GameLiftStreams::StreamGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GlobalAccelerator::Accelerator</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GlobalAccelerator::CrossAccount Attachment</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GlobalAccelerator::Listener</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Glue::Crawler</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::Database</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::IdentityCenterConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::IntegrationResourceProperty</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Glue::Job</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::Registry</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Glue::Schema</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Glue::SchemaVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Glue::SchemaVersionMetadata</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Glue::Trigger</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Glue::UsageProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Grafana::Workspace</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GreengrassV2::ComponentVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GreengrassV2::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GroundStation::Config</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GroundStation::DataflowEndpoint Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GroundStation::DataflowEndpoint GroupV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GroundStation::MissionProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::GuardDuty::Detector</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::GuardDuty::Filter</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::IPSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::Master</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::Member</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::PublishingDestination</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::GuardDuty::ThreatEntitySet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::ThreatIntelSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::GuardDuty::TrustedEntitySet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::HealthImaging::Datastore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::HealthLake::FHIRDatastore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::GroupPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IAM::InstanceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::ManagedPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::OIDCProvider</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::Role</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::RolePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IAM::SAMLProvider</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::ServerCertificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IAM::ServiceLinkedRole</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IAM::User</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IAM::UserPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IAM::VirtualMFADevice</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IVS::EncoderConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::IngestConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IVS::PlaybackKeyPair</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::PublicKey</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::RecordingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IVS::Stage</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::StorageConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IVS::StreamKey</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IVSChat::LoggingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IVSChat::Room</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IdentityStore::Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IdentityStore::GroupMembership</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::Component</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::ContainerRecipe</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ImageBuilder::DistributionConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ImageBuilder::Image</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::ImagePipeline</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::ImageRecipe</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ImageBuilder::Workflow</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Inspector::AssessmentTarget</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Inspector::AssessmentTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Inspector::ResourceGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::InspectorV2::CisScanConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::InspectorV2::CodeSecurityIntegration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::InspectorV2::CodeSecurityScanConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::InspectorV2::Filter</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::InternetMonitor::Monitor</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Invoicing::InvoiceUnit</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT::AccountAuditConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::Authorizer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::BillingGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::CACertificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::Certificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::CertificateProvider</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoT::Command</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT::CustomMetric</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::Dimension</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::DomainConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT::EncryptionConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT::FleetMetric</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT::JobTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoT::Logging</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::MitigationAction</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::Policy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::ProvisioningTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::ResourceSpecificLogging</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::RoleAlias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT::ScheduledAudit</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteccão de desvios	Gerador de IaC
AWS::IoT::SecurityProfile	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::IoT::SoftwarePackage	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::IoT::SoftwarePackageVersion	 Yes (Sim)	 Yes (Sim)	 Não
AWS::IoT::Thing	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::IoT::ThingGroup	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::IoT::ThingType	 Yes (Sim)	 Yes (Sim)	 Não
AWS::IoT::TopicRule	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoT::TopicRuleDestination</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTAnalytics::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTAnalytics::Dataset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTAnalytics::Datastore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTAnalytics::Pipeline</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTEvents::AlarmModel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoTEvents::DetectorModel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTEvents::Input</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTFleetWise::Campaign</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTFleetWise::DecoderManifest</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTFleetWise::Fleet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTFleetWise::ModelManifest</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTFleetWise::SignalCatalog</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoT FleetWise::StateTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT FleetWise::Vehicle</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT SiteWise::AccessPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT SiteWise::Asset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT SiteWise::AssetModel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoT SiteWise::ComputationModel</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoT SiteWise::Dashboard</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoTSiteWise::Dataset</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoTSiteWise::Gateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTSiteWise::Portal</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoTSiteWise::Project</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::IoTTwinMaker::ComponentType</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTTwinMaker::Entity</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTTwinMaker::Scene</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoTTwinMaker::SyncJob</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTTwinMaker::Workspace</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::Destination</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::DeviceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::FuotaTask</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::MulticastGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::NetworkAnalyzerCon figuration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::IoTWireless::PartnerAccount</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::ServiceProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::TaskDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::WirelessDevice</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::WirelessDeviceImportTask</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::IoTWireless::WirelessGateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::KMS::Alias</u>	 Yes (Sim)	 Não	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::KMS::Key</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::KMS::ReplicaKey</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::KafkaConnect::Connector</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::KafkaConnect::CustomPlugin</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Kendra::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Kendra::Faq</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Kendra::Index</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::KendraRanking::ExecutionPlan</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Kinesis::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Kinesis::Stream</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Kinesis::StreamConsumer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::KinesisAnalyticsV2::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::KinesisFirehose::DeliveryStream</u>	 Sim	 No (Não)	 No (Não)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::KinesisVideo::SignalingChannel</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::KinesisVideo::Stream</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LakeFormation::DataCellsFilter</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LakeFormation::PrincipalPermissions</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LakeFormation::Tag</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LakeFormation::TagAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lambda::Alias</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Lambda::CapacityProvider</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lambda::CodeSigningConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lambda::EventInvokeConfig</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lambda::EventSourceMapping</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lambda::Function</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lambda::LayerVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lambda::LayerVersionPermission</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Lambda::Permission</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lambda::Url</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lambda::Version</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::LaunchWizard::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lex::Bot</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lex::BotAlias</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lex::BotVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Lex::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LicenseManager::Grant</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LicenseManager::License</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lightsail::Alarm</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::Bucket</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::Certificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::Container</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Lightsail::Database</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lightsail::Disk</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::DiskSnapshot</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lightsail::Distribution</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Lightsail::Instance</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::InstanceSnapshot</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Lightsail::LoadBalancer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Lightsail::StaticIp</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Location::APIKey</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Location::GeofenceCollection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Location::Map</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Location::PlaceIndex</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Location::RouteCalculator</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Location::Tracker</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Location::TrackerConsumer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::AccountPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::Delivery</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Logs::DeliveryDestination</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Logs::DeliverySource</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Logs::Destination</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Logs::LogAnomalyDetector</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Logs::LogGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::LogStream</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::MetricFilter</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::QueryDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Logs::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::SubscriptionFilter</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Logs::Transformer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LookoutEquipment::InferenceScheduler</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::LookoutVision::Project</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::M2::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::M2::Deployment</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::M2::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MPA::ApprovalTeam</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MPA::IdentitySource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MSK::BatchScramSecret</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MSK::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MSK::ClusterPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MSK::Configuration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MSK::Replicator</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MSK::ServerlessCluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MSK::VpcConnection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MWAA::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Macie::AllowList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Macie::CustomDataIdentifier</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Macie::FindingsFilter</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Macie::Session</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ManagedBlockchain::Accessor</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaConnect::Bridge</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaConnect::BridgeOutput</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::BridgeSource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::Flow</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::FlowEntitlement</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MediaConnect::FlowOutput</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::FlowSource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::FlowVpcInterface</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaConnect::Gateway</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaConnect::RouterInput</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::RouterNetworkInterface</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaConnect::RouterOutput</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MediaLive::ChannelPlacementGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::CloudWatchAlarmTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::CloudWatchAlarmTemplateGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::EventBridgeRuleTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::EventBridgeRuleTemplateGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::Multiplex</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MediaLive::Multiplexprogram</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::Network</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::SdiSource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaLive::SignalMap</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaPackage::Asset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackage::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackage::OriginEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MediaPackage::PackagingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackage::PackagingGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackageV2::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackageV2::ChannelGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MediaTailor::Channel</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaTailor::ChannelPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaTailor::LiveSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MediaTailor::SourceLocation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MediaTailor::VodSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MemoryDB::ACL</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::MemoryDB::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MemoryDB::MultiRegionCluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::MemoryDB::ParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MemoryDB::SubnetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::MemoryDB::User</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Neptune::DBCluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Neptune::DBClusterParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Neptune::DBInstance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Neptune::DBParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Neptune::DBSubnetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Neptune::EventSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NeptuneGraph::Graph</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkFirewall::Firewall</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkFirewall::RuleGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkManager::ConnectAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::ConnectPeer</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::NetworkManager::CoreNetwork</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::CoreNetworkPrefixListAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::Device</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::NetworkManager::GlobalNetwork</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::Link</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::NetworkManager::LinkAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::Site</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::NetworkManager::VpcAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Notifications::ChannelAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::EventRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::ManagedNotificationAccountContactAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::ManagedNotificationAdditionalChannelAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::NotificationConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::NotificationHub</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Notifications::OrganizationalUnitAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::NotificationsContacts::EmailContact</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ODB::CloudAutonomousVmCluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ODB::CloudExadataInfrastructure</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ODB::CloudVmCluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ODB::OdbNetwork</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ODB::OdbPeeringConnection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::OSIS::Pipeline</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Oam::Link</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Oam::Sink</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ObservabilityAdmin::OrganizationCentralizationRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ObservabilityAdmin::OrganizationTelemetryRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ObservabilityAdmin::S3TableIntegration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ObservabilityAdmin::TelemetryPipelines</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ObservabilityAdmin::TelemetryRule</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Omics::AnnotationStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::ReferenceStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::RunGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::SequenceStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::VariantStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::Workflow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Omics::WorkflowVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
AWS::OpenSearchServerless::AccessPolicy	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::OpenSearchServerless::Collection	 Yes (Sim)	 Yes (Sim)	 Não
AWS::OpenSearchServerless::CollectionGroup	 Yes (Sim)	 Yes (Sim)	 Não
AWS::OpenSearchServerless::Index	 Yes (Sim)	 Yes (Sim)	 Não
AWS::OpenSearchServerless::LifecyclePolicy	 Yes (Sim)	 Yes (Sim)	 Não
AWS::OpenSearchServerless::SecurityConfig	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::OpenSearchServerless::SecurityPolicy	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::OpenSearchService::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::OpenSearchService::Domain</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Organizations::Account</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Organizations::Organization</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Organizations::OrganizationalUnit</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Organizations::Policy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Organizations::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorAD::Connector</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorAD::DirectoryRegistration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorAD::ServicePrincipalName</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorAD::Template</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorAD::TemplateGroupAccessControlEntry</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::PCAConectorSCEP::Challenge</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::PCAConnectorSCEP::Connector</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::PCS::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::PCS::ComputeNodeGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::PCS::Queue</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Panorama::ApplicationInstance</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Panorama::Package</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Panorama::PackageVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::PaymentCryptography::Alias</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::PaymentCryptography::Key</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Personalize::Dataset</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Personalize::DatasetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Personalize::Schema</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Personalize::Solution</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Pinpoint::InAppTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Pipes::Pipe</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Proton::EnvironmentAccountConnection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Proton::EnvironmentTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Proton::ServiceTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QBusiness::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::DataAccessor</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::QBusiness::Index</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::Permission</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::Plugin</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::Retriever</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QBusiness::WebExperience</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QLDB::Stream</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QuickSight::Analysis</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::QuickSight::CustomPermissions</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QuickSight::Dashboard</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QuickSight::DataSet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QuickSight::DataSource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QuickSight::Folder</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QuickSight::RefreshSchedule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QuickSight::Template</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::QuickSight::Theme</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::QuickSight::Topic</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::QuickSight::VPCConnection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RAM::Permission</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RAM::ResourceShare</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RDS::CustomDBEngineVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RDS::DBCluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::RDS::DBClusterParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RDS::DBInstance</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::DBParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RDS::DBProxy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::DBProxyEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::DBProxyTargetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::DBShardGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::RDS::DBSubnetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::EventSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::GlobalCluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RDS::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RDS::OptionGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RTBFabric::InboundExternalLink</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RTBFabric::Link</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::RTBFabric::OutboundExternalLink</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RTBFabric::RequesterGateway</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RTBFabric::ResponderGateway</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RUM::AppMonitor</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Rbin::Rule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Redshift::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Redshift::ClusterParameterGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Redshift::ClusterSubnetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Redshift::EndpointAccess</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Redshift::EndpointAuthorization</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Redshift::EventSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Redshift::Integration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Redshift::ScheduledAction</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RedshiftServerless::Namespace</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::RedshiftServerless::Snapshot</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RedshiftServerless::Workgroup</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RefactorSpaces::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RefactorSpaces::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RefactorSpaces::Route</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RefactorSpaces::Service</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Rekognition::Collection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Rekognition::Project</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Rekognition::StreamProcessor</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ResilienceHub::App</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ResourceExplorer2::DefaultViewAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ResourceExplorer2::Index</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ResourceExplorer2::View</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ResourceGroups::Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ResourceGroups::TagSyncTask</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::RolesAnywhere::CRL</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RolesAnywhere::Profile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::RolesAnywhere::TrustAnchor</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53::CidrCollection</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53::DNSSEC</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Route53::HealthCheck</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53::HostedZone</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53::KeySigningKey</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Profiles::Profile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53Profiles::ProfileAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53RecoveryControl::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryControl::SafetyRule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryReadiness::Cell</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryReadiness::ReadinessCheck</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryReadiness::RecoveryGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53RecoveryReadiness::ResourceSet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Route53Resolver::FirewallDomainList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53Resolver::FirewallRuleGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::FirewallRuleGroupAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::OutpostResolver</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::ResolverConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::ResolverDNSSECConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::ResolverEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Route53Resolver::ResolverRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3::AccessGrant</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3::AccessGrantsInstance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3::AccessGrantsLocation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::S3::AccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3::Bucket</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3::BucketPolicy</u>	 Yes (Sim)	 Não	 Yes (Sim)
<u>AWS::S3::MultiRegionAccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::S3::StorageLens</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3::StorageLensGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::S3Express::AccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Express::BucketPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Express::DirectoryBucket</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3ObjectLambda::AccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3ObjectLambda::AccessPointPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Outposts::AccessPoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3Outposts::Bucket</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::S3Outposts::BucketPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Outposts::Endpoint</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::S3Tables::Namespace</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Tables::Table</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Tables::TableBucket</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Tables::TableBucketPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Tables::TablePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::S3Vectors::Index</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Vectors::VectorBucket</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::S3Vectors::VectorBucketPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::ConfigurationSet</u>	 Yes (Sim)	 Yes (Sim)	 No (Não)
<u>AWS::SES::ConfigurationSetEventDestination</u>	 No (Não)	 Sim	 Não
<u>AWS::SES::ContactList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::DedicatedIpPool</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SES::EmailIdentity</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SES::MailManagerAddonInstance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerAddonSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerAddressList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerArchive</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerIngressPoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerRelay</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SES::MailManagerRuleSet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MailManagerTrafficPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::MultiRegionEndpoint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::Template</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::Tenant</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SES::VdmAttributes</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::ConfigurationSet</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SMSVOICE::OptOutList</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::PhoneNumber</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::Pool</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::ProtectConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SMSVOICE::SenderId</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SNS::Subscription</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SNS::Topic</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SNS::TopicInlinePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SQS::Queue</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SQS::QueueInlinePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSM::Association</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SSM::Document</u>	 Yes (Sim)	 No (Não)	 No (Não)
<u>AWS::SSM::MaintenanceWindowTarget</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SSM::MaintenanceWindowTask</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSM::Parameter</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSM::PatchBaseline</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSM::ResourceDataSync</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSM::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SSMContacts::Contact</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SSMContacts::ContactChannel</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SSMContacts::Plan</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSMContacts::Rotation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSMGuiConnect::Preferences</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSMIncidents::ReplicationSet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SSMIncidents::ResponsePlan</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSMQuickSetup::LifecycleAutomation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SSO::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSO::ApplicationAssignment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSO::Assignment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSO::Instance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::App</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::AppImageConfig</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SageMaker::Cluster</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::Device</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::DeviceFleet</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::Domain</u>	 Yes (Sim)	 Yes (Sim)	 No (Não)
<u>AWS::SageMaker::Endpoint</u>	 No (Não)	 Sim	 Não
<u>AWS::SageMaker::FeatureGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SageMaker::Image</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::ImageVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::InferenceComponent</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::InferenceExperiment</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::MlflowTrackingServer</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::ModelCard</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::ModelPackage</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::ModelPackageGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::MonitoringSchedule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::PartnerApp</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::Pipeline</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SageMaker::ProcessingJob</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::Project</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SageMaker::Space</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::StudioLifecycleConfig</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SageMaker::UserProfile</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Scheduler::Schedule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Scheduler::ScheduleGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SecretsManager::ResourcePolicy</u>	 Sim	 No (Não)	 No (Não)
<u>AWS::SecretsManager::RotationSchedule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecretsManager::Secret</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::AggregatorV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::AutomationRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::AutomationRuleV2</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::ConnectorV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::DelegatedAdmin</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::FindingAggregator</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::Hub</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::HubV2</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::Insight</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SecurityHub::OrganizationConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::PolicyAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::ProductSubscription</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::SecurityControl</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityHub::Standard</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityLake::AwsLogSource</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityLake::DataLake</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SecurityLake::Subscriber</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::SecurityLake::SubscriberNotification</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 Sim	 No (Não)	 No (Não)
<u>AWS::ServiceCatalog::LaunchNotificationConstraint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::LaunchTemplateConstraint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::PortfolioPrincipalAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::PortfolioProductAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ServiceCatalog::PortfolioShare</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::ResourceUpdateConstraint</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::ServiceAction</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ServiceCatalog::ServiceActionAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::TagOption</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalog::TagOptionAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 Sim	 No (Não)	 No (Não)
<u>AWS::Shield::DRTAccess</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Shield::ProactiveEngagement</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Shield::Protection</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Shield::ProtectionGroup</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Signer::ProfilePermission</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Signer::SigningProfile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SimSpaceWeaver::Simulation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::StepFunctions::Activity</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::StepFunctions::StateMachine</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::StepFunctions::StateMachineAlias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::StepFunctions::StateMachineVersion</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::SupportApp::AccountAlias</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SupportApp::SlackChannelConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SupportApp::SlackWorkspaceConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Synthetics::Canary</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Synthetics::Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::SystemsManagerSAP::Application</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Timestream::Database</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Timestream::InfluxDBInstance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Timestream::ScheduledQuery</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Timestream::Table</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Transfer::Agreement</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Transfer::Certificate</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Transfer::Connector</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Transfer::Profile</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Transfer::Server</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Transfer::User</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Transfer::WebApp</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Transfer::Workflow</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VerifiedPermissions::IdentitySource</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VerifiedPermissions::Policy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VerifiedPermissions::PolicyStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
AWS::VerifiedPermissions::PolicyTemplate	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::VoiceID::Domain	 Yes (Sim)	 Yes (Sim)	 Não
AWS::VpcLattice::AccessLogSubscription	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::VpcLattice::AuthPolicy	 Yes (Sim)	 Yes (Sim)	 Não
AWS::VpcLattice::DomainVerification	 Yes (Sim)	 Yes (Sim)	 Não
AWS::VpcLattice::Listener	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
AWS::VpcLattice::ResourceConfiguration	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::VpcLattice::ResourceGateway</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::VpcLattice::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::VpcLattice::Rule</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VpcLattice::Service</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VpcLattice::ServiceNetwork</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::VpcLattice::TargetGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WAFv2::IPSet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WAFv2::LoggingConfiguration</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WAFv2::RegexPatternSet</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WAFv2::RuleGroup</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WAFv2::WebACL</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::WAFv2::WebACLAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIAgent</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIAgentVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIGuardrail</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIGuardrailVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIPrompt</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::AIPromptVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::Wisdom::Assistant</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Wisdom::AssistantAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Wisdom::KnowledgeBase</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::Wisdom::MessageTemplate</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::MessageTemplateVersion</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::Wisdom::QuickResponse</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkSpaces::ConnectionAlias</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::WorkSpaces::WorkspacesPool</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkSpacesThinClient::Environment</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::WorkSpacesWeb::Portal</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::SessionLogger</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkSpacesWeb::TrustStore</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkSpacesWeb::UserSettings</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::WorkspacesInstances::Volume</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::WorkspacesInstances::VolumeAssociation</u>	 Yes (Sim)	 Yes (Sim)	 Não

Recurso	Importar	Deteção de desvios	Gerador de IaC
<u>AWS::WorkspacesInstances::WorkspaceInstance</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::XRay::Group</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::XRay::ResourcePolicy</u>	 Yes (Sim)	 Yes (Sim)	 Yes (Sim)
<u>AWS::XRay::SamplingRule</u>	 Yes (Sim)	 Yes (Sim)	 Não
<u>AWS::XRay::TransactionSearchConfig</u>	 Yes (Sim)	 Yes (Sim)	 No (Não)

Use links de criação rápida para criar pilhas do CloudFormation

Os links de criação rápida fornecem um método simplificado para iniciar pilhas do CloudFormation diretamente de URLs no console do CloudFormation. Ao especificar o modelo de URL, o nome da pilha e parâmetros do modelo como parâmetros de consulta de URL, é possível preencher previamente uma única página de Criar pilha para acelerar a criação de uma pilha. Isso simplifica o processo de criação de pilhas, reduzindo tanto o número de páginas do assistente quanto a

quantidade de entrada do usuário necessária. Isso também otimiza a reutilização de modelos, pois você pode criar vários URLs que especificam valores diferentes para o mesmo modelo.

Formato do URL

O link de criação rápida segue este formato de URL:

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/  
stacks/create/review  
?templateURL=TemplateURL  
&stackName=StackName  
&param_parameterName=parameterValue
```

O CloudFormation é compatível com os seguintes parâmetros de consulta de URL:

URL do modelo

Obrigatório. O parâmetro `templateURL` especifica o URL do modelo de pilha localizado em um bucket do Amazon S3. Para evitar problemas de acesso com um URL do S3 pré-assinado, certifique-se de codificar o URL.

Formatos de URL do S3 válidos:

- `https://s3.region-code.amazonaws.com/bucket-name/template-name`
- `https://bucket-name.s3.region-code.amazonaws.com/template-name`
- `https://s3-region-code.amazonaws.com/bucket-name/template-name` (formato legado)

Nome da pilha

Opcional. Use o parâmetro `stackName` para especificar o nome da pilha do CloudFormation a ser criada. Um nome de pilha pode conter apenas caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 128 caracteres.

Parâmetros do modelo

Opcional. Para parâmetros no modelo de pilha que não são do tipo de parâmetro `NoEcho`, use o formato `param_parameterName` na string de consulta do URL. O parâmetro de URL deve incluir o prefixo `param_` e o segmento do nome do parâmetro deve corresponder exatamente ao nome do parâmetro no modelo. Por exemplo: `param_DBName`.

O CloudFormation ignora os parâmetros que não existem no modelo e quaisquer parâmetros definidos com a propriedade NoEcho definida como tipos `true` (geralmente, nomes de usuário e senhas). Os parâmetros de URL substituem os valores padrão que são especificados no modelo. Inclua quantos parâmetros forem necessários.

Important

Em vez de incorporar informações confidenciais diretamente em modelos do CloudFormation, recomendamos usar os parâmetros dinâmicos no modelo da pilha para fazer referência a informações confidenciais que são armazenadas e gerenciadas de forma externa ao CloudFormation, como no AWS Systems Manager Parameter Store ou no AWS Secrets Manager.

Para obter mais informações, consulte a prática recomendada [Não incorporar credenciais nos seus modelos](#).

Todos os nomes de parâmetro de consulta diferenciam maiúsculas de minúsculas. Os usuários podem substituir esses valores no console antes de criar a pilha.

Exemplo

O exemplo a seguir é baseado no modelo de exemplo da [Instância básica única do WordPress](#). A string de consulta inclui o parâmetro `templateURL` obrigatório e os parâmetros `stackName`, `DBName`, `InstanceType` e `KeyName`.

O URL a seguir tem quebras de linha adicionadas para oferecer clareza.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review
?templateURL=https://s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/
WordPress_Single_Instance.template
&stackName=MyWPBlog
&param_DBName=mywpblog
&param_InstanceType=t2.medium
```

O URL a seguir inclui os mesmos parâmetros do exemplo anterior, mas as quebras de linha foram removidas. Este é o formato real do URL.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?
region=us-east-2#/stacks/create/review?templateURL=https://
```

```
s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/WordPress_Single_Instance.template&stackName=MyWPBlog&param_DBName=mywpblog&param_InstanceType=
```

Criando uma pilha usando um link de criação rápida

Ao abrir um link de criação rápida, você é direcionado para o console do CloudFormation. O console abre diretamente na página Criar pilha rapidamente, com os valores fornecidos usados automaticamente para os parâmetros.

Para criar uma pilha usando um link de criação rápida (console)

1. Na página Criar pilha rapidamente, em Modelo, URL do modelo, confirme se o URL do modelo está correto.
2. Expanda a seção Visualizar modelo para verificar o modelo.
3. Em Nome da pilha, verifique o nome da pilha pré-preenchido.
4. Revise a seção Parâmetros. Verifique se os valores dos parâmetros pré-preenchidos estão corretos. Preencha todos os parâmetros obrigatórios que não foram especificados no URL. Modifique quaisquer valores pré-preenchidos, se necessário.
5. Agora, é possível definir as seguintes configurações:
 - Tags — Organize recursos com pares de chave-valor.
 - Permissões: escolha o perfil de serviço do IAM para operações de pilha.
 - Opções de falha na pilha — Escolha reverter (padrão) ou preservar recursos.
 - Política de pilha — Controle as permissões de atualização de recursos.
 - Configuração de reversão — Configure a reversão baseada em alarme do CloudWatch.
 - Opções de notificação — Configure notificações do Amazon SNS para eventos de pilha.
 - Opções de criação de pilha — Defina o tempo máximo de criação da pilha e ative a proteção contra encerramento para evitar exclusões acidentais.

Para obter mais informações, consulte [Configurar opções da pilha](#).

6. Para Capacidades, preencha todas as confirmações necessárias. Se o seu modelo contém recursos do IAM, selecione I acknowledge that this template may create IAM resources (Eu reconheço que este modelo pode criar recursos do IAM) para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).

7. (Opcional) É possível criar um conjunto de alterações para visualizar a configuração da pilha do antes de criá-la. Escolha Criar conjunto de alterações e siga as instruções. Para obter mais informações, consulte [Visualizar a configuração da sua pilha](#).
8. Quando estiver pronto, escolha Criar pilha para iniciar a pilha e, em seguida, monitore o progresso da criação da pilha na guia Eventos. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Exemplos de comandos de operação de pilha do CloudFormation para o AWS CLI e o PowerShell

Os exemplos de linha de comando a seguir demonstram como realizar ações individuais do CloudFormation com a AWS CLI e o PowerShell. Estes exemplos incluem apenas as ações mais usadas. Para obter uma lista completa, consulte [cloudformation](#) na AWS CLI Command Reference.

Os exemplos deste guia usam a convenção de uma barra invertida (\) para indicar que uma linha de comando longa continua na próxima linha.

Tópicos

- [Cancelar uma atualização de pilha](#)
- [Continuar revertendo uma atualização](#)
- [Criar uma pilha](#)
- [Criar uma pilha que inclua transformações](#)
- [Excluir uma pilha](#)
- [Descrever os eventos da pilha](#)
- [Descrever um recurso de pilha](#)
- [Descrever recurso de pilha](#)
- [Descrever pilhas](#)
- [Obter um modelo](#)
- [Listar recursos de pilha](#)
- [Listar pilhas](#)
- [Atualizar um stack](#)
- [Validar seu modelo](#)

- [Fazer upload de artefatos locais em um bucket do S3 com a AWS CLI](#)

Cancelar uma atualização de pilha

Use o comando [cancel-update-stack](#) para cancelar uma atualização de pilha. Para obter mais informações, consulte [Cancelar uma atualização de pilha](#).

CLI

AWS CLI

Para cancelar uma atualização de pilha que está em andamento

O comando `cancel-update-stack` a seguir cancela uma atualização de pilha na pilha `myteststack`:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Consulte detalhes da API em [CancelUpdateStack](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: cancela uma atualização na pilha especificada.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Consulte detalhes da API em [CancelUpdateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: cancela uma atualização na pilha especificada.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Para obter detalhes da API, consulte [CancelUpdateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Continuar revertendo uma atualização

Use o comando [continue-update-rollback](#) para continuar revertendo uma atualização. Para obter mais informações, consulte [Continuar revertendo uma atualização](#).

CLI

AWS CLI

Para fazer nova tentativa de reversão de atualização

O exemplo `continue-update-rollback` a seguir retoma uma operação de reversão de uma atualização de pilha anterior em que houve falha.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Este comando não produz saída.

- Consulte detalhes da API em [ContinueUpdateRollback](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: continua a reversão da pilha nomeada, que deve estar no estado "UPDATE_ROLLBACK_FAILED". Se a reversão contínua for bem-sucedida, a pilha entrará no estado "UPDATE_ROLLBACK_COMPLETE".

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Consulte detalhes da API em [ContinueUpdateRollback](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: continua a reversão da pilha nomeada, que deve estar no estado "UPDATE_ROLLBACK_FAILED". Se a reversão contínua for bem-sucedida, a pilha entrará no estado "UPDATE_ROLLBACK_COMPLETE".

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Para obter detalhes da API, consulte [ContinueUpdateRollback](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Criar uma pilha

Use o comando [create-stack](#) para criar uma pilha. Você deve fornecer o nome da pilha, a localização de um modelo válido e quaisquer parâmetros de entrada. Os nomes das chaves dos parâmetros diferenciam maiúsculas de minúsculas. Se você digitar incorretamente o nome de uma chave de parâmetro, o CloudFormation não criará a pilha e relatará que o modelo não contém esse parâmetro.

Os exemplos a seguir mostram como criar uma nova pilha com o nome, o modelo e os parâmetros de entrada especificados.

CLI

AWS CLI

Para criar uma pilha do AWS CloudFormation

O exemplo de `create-stacks` a seguir cria uma pilha com o nome de `myteststack` usando o arquivo de modelo `sampletemplate.json`:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Resultado:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

Para obter mais informações, consulte Pilhas no Guia do usuário do AWS CloudFormation.

- Consulte detalhes da API em [CreateStack](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: cria uma nova pilha com o nome especificado. O modelo é analisado a partir do conteúdo fornecido com parâmetros de personalização ("PK1" e "PK2" representam os nomes dos parâmetros declarados no conteúdo do modelo, "PV1" e "PV2" representam os valores desses parâmetros. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela não será revertida.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Exemplo 2: cria uma nova pilha com o nome especificado. O modelo é analisado a partir do conteúdo fornecido com parâmetros de personalização ("PK1" e "PK2" representam os nomes dos parâmetros declarados no conteúdo do modelo, "PV1" e "PV2" representam os valores desses parâmetros. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( $p1, $p2 ) `
              -OnFailure "ROLLBACK"
```

Exemplo 3: cria uma nova pilha com o nome especificado. O modelo é obtido da URL do Amazon S3 com parâmetros de personalização ("PK1" representa o nome de um parâmetro declarado no conteúdo do modelo, "PV1" representa o valor do parâmetro. Os parâmetros

de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida (o mesmo que especificar `-DisableRollback $false`).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 4: cria uma nova pilha com o nome especificado. O modelo é obtido da URL do Amazon S3 com parâmetros de personalização ("PK1" representa o nome de um parâmetro declarado no conteúdo do modelo, "PV1" representa o valor do parâmetro. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida (o mesmo que especificar `-DisableRollback $false`). Os AENs de notificação especificados receberão os eventos relacionados à pilha publicados.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Consulte detalhes da API em [CreateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: cria uma nova pilha com o nome especificado. O modelo é analisado a partir do conteúdo fornecido com parâmetros de personalização ("PK1" e "PK2" representam os nomes dos parâmetros declarados no conteúdo do modelo, "PV1" e "PV2" representam os valores desses parâmetros. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela não será revertida.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Exemplo 2: cria uma nova pilha com o nome especificado. O modelo é analisado a partir do conteúdo fornecido com parâmetros de personalização ("PK1" e "PK2" representam os nomes dos parâmetros declarados no conteúdo do modelo, "PV1" e "PV2" representam os valores desses parâmetros. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( $p1, $p2 ) `
              -OnFailure "ROLLBACK"
```

Exemplo 3: cria uma nova pilha com o nome especificado. O modelo é obtido da URL do Amazon S3 com parâmetros de personalização ("PK1" representa o nome de um parâmetro declarado no conteúdo do modelo, "PV1" representa o valor do parâmetro. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida (o mesmo que especificar -DisableRollback \$false).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 4: cria uma nova pilha com o nome especificado. O modelo é obtido da URL do Amazon S3 com parâmetros de personalização ("PK1" representa o nome de um parâmetro declarado no conteúdo do modelo, "PV1" representa o valor do parâmetro. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". Se houver falha na criação da pilha, ela será revertida (o mesmo que especificar -DisableRollback \$false). Os AENs de notificação especificados receberão os eventos relacionados à pilha publicados.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Para obter detalhes da API, consulte [CreateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Note

É possível usar a opção `--template-url` da AWS CLI para especificar a localização de um arquivo de modelo no Amazon S3 ou AWS Systems Manager.

Para o Amazon S3, o URL deve começar com `https://`. URLs de sites estáticos do S3 não são compatíveis.

```
--template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
```

Use o seguinte formato para o AWS Systems Manager:

```
--template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
```

Criar uma pilha que inclua transformações

Use o comando [deploy](#) para criar uma pilha que inclua transformações. Ao criar uma pilha com base em um modelo que inclui transformações, você deve usar um conjunto de alterações. O comando `deploy` combina duas etapas (criar um conjunto de alterações e executá-lo) em um único comando.

AWS CLI

O comando `deploy` a seguir cria uma pilha com o nome, o modelo e os parâmetros de entrada especificados.

```
aws cloudformation deploy --stack-name myteststack \
  --template /path_to_template/my-template.json \
  --parameter-overrides Key1=Value1 Key2=Value2
```

Excluir uma pilha

Use o comando [delete-stack](#) para excluir uma pilha. Para obter mais informações, consulte [Excluir uma pilha](#).

CLI

AWS CLI

Para excluir uma pilha

O exemplo de `delete-stack` a seguir exclui a pilha especificada.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Este comando não produz saída.

- Consulte detalhes da API em [DeleteStack](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: exclui a pilha especificada.

```
Remove-CFNStack -StackName "myStack"
```

- Consulte detalhes da API em [DeleteStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: exclui a pilha especificada.

```
Remove-CFNStack -StackName "myStack"
```

- Para obter detalhes da API, consulte [DeleteStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Se a exclusão apresentar falhas e retornar um estado de `DELETE_FAILED`, você poderá optar por excluir a pilha de forma forçada usando a opção `--deletion-mode`. Estes são os seguintes valores que podem ser usados com `deletion-mode`:

- `STANDARD`: exclui a pilha normalmente. Este é o modo de exclusão padrão.
- `FORCE_DELETE_STACK`: exclui a pilha e ignora todos os recursos que não estão sendo excluídos.

AWS CLI

O seguinte comando `delete-stack` força a exclusão da pilha *myteststack* usando o valor `FORCE_DELETE_STACK` com o parâmetro `deletion-mode`:

```
aws cloudformation delete-stack --stack-name myteststack \  
  --deletion-mode FORCE_DELETE_STACK
```

Este comando não produz saída.

Depois de usar `FORCE_DELETE_STACK`, você pode usar o comando `list-stack-resources` para listar os recursos que foram ignorados durante o processo de exclusão da pilha. Os recursos retidos mostrarão o status `DELETE_SKIPPED`. Para obter mais informações, consulte [Listar recursos de pilha](#).

Descrever os eventos da pilha

Use o comando [describe-stack-events](#) para descrever os eventos da pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

CLI

AWS CLI

Para descrever os eventos da pilha

O exemplo `describe-stack-events` a seguir exibe os dois eventos mais recentes da pilha especificada.

```
aws cloudformation describe-stack-events \  
  --stack-name my-stack \  
  --max-items 2
```

```

--max-items 2

{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.556Z",
      "ResourceStatus": "UPDATE_COMPLETE"
    },
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.127Z",
      "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
    }
  ],
  "NextToken": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- Consulte detalhes da API em [DescribeStackEvents](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna todos os eventos relacionados à pilha especificada.

```
Get-CFNStackEvent -StackName "myStack"
```

- Consulte detalhes da API em [DescribeStackEvents](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna todos os eventos relacionados à pilha especificada.

```
Get-CFNStackEvent -StackName "myStack"
```

- Para obter detalhes da API, consulte [DescribeStackEvents](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Descrever um recurso de pilha

CLI

AWS CLI

Para obter informações detalhadas sobre o recurso de uma pilha

O exemplo de `describe-stack-resource` a seguir exibe os detalhes do recurso denominado `MyFunction` da pilha especificada.

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

Resultado:

```
{  
  "StackResourceDetail": {  
    "StackName": "MyStack",  
    "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
    "LogicalResourceId": "MyFunction",  
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",  
    "ResourceType": "AWS::Lambda::Function",  
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",  
    "ResourceStatus": "UPDATE_COMPLETE",  
    "Metadata": "{}",  
    "DriftInformation": {
```

```
        "StackResourceDriftStatus": "IN_SYNC"  
      }  
    }  
  }
```

- Consulte detalhes da API em [DescribeStackResource](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna a descrição de um recurso identificado no modelo associado à pilha especificada pelo ID lógico "MyDBInstance".

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Consulte detalhes da API em [DescribeStackResource](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna a descrição de um recurso identificado no modelo associado à pilha especificada pelo ID lógico "MyDBInstance".

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Para obter detalhes da API, consulte [DescribeStackResource](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Descrever recurso de pilha

CLI

AWS CLI

Para obter informações detalhadas sobre o recurso de uma pilha

O exemplo de `describe-stack-resources` a seguir exibe os detalhes dos recursos na pilha especificada.

```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

Resultado:

```
{
  "StackResources": [
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "bucket",
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
      "ResourceType": "AWS::S3::Bucket",
      "Timestamp": "2019-10-02T04:34:11.345Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
      "Timestamp": "2019-10-02T05:34:27.989Z",
      "ResourceStatus": "UPDATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "functionRole",
      "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
      "ResourceType": "AWS::IAM::Role",
      "Timestamp": "2019-10-02T04:34:06.350Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
```

```
        "StackResourceDriftStatus": "IN_SYNC"
    }
}
]
```

- Consulte detalhes da API em [DescribeStackResources](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna descrições dos recursos da AWS, para até 100 recursos associados à pilha especificada. Para obter os detalhes de todos os recursos associados a uma pilha, use o `Get-CFNStackResourceSummary`, que também é compatível com a paginação manual dos resultados.

```
Get-CFNStackResourceList -StackName "myStack"
```

Exemplo 2: retorna a descrição da instância do Amazon EC2 identificada no modelo associado à pilha especificada pelo ID lógico "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Exemplo 3: retorna a descrição de até 100 recursos associados à pilha contendo uma instância do Amazon EC2 identificada pelo ID de instância "i-123456". Para obter os detalhes de todos os recursos associados a uma pilha, use o `Get-CFNStackResourceSummary`, que também é compatível com a paginação manual dos resultados.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Exemplo 4: retorna a descrição da instância do Amazon EC2 identificada pelo ID lógico "Ec2Instance" no modelo de uma pilha. A pilha é identificada usando o ID de recurso físico de um recurso que ela contém; nesse caso, também uma instância do Amazon EC2 com o ID de instância "i-123456". Um outro recurso físico também poderia ser usado para identificar a pilha, dependendo do conteúdo do modelo, por exemplo, um bucket do Amazon S.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Consulte detalhes da API em [DescribeStackResources](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna descrições dos recursos da AWS, para até 100 recursos associados à pilha especificada. Para obter os detalhes de todos os recursos associados a uma pilha, use o `Get-CFNStackResourceSummary`, que também é compatível com a paginação manual dos resultados.

```
Get-CFNStackResourceList -StackName "myStack"
```

Exemplo 2: retorna a descrição da instância do Amazon EC2 identificada no modelo associado à pilha especificada pelo ID lógico "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Exemplo 3: retorna a descrição de até 100 recursos associados à pilha contendo uma instância do Amazon EC2 identificada pelo ID de instância "i-123456". Para obter os detalhes de todos os recursos associados a uma pilha, use o `Get-CFNStackResourceSummary`, que também é compatível com a paginação manual dos resultados.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Exemplo 4: retorna a descrição da instância do Amazon EC2 identificada pelo ID lógico "Ec2Instance" no modelo de uma pilha. A pilha é identificada usando o ID de recurso físico de um recurso que ela contém; nesse caso, também uma instância do Amazon EC2 com o ID de instância "i-123456". Um outro recurso físico também poderia ser usado para identificar a pilha, dependendo do conteúdo do modelo, por exemplo, um bucket do Amazon S.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId "Ec2Instance"
```

- Para obter detalhes da API, consulte [DescribeStackResources](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Descrever pilhas

CLI

AWS CLI

Para descrever as pilhas do AWS CloudFormation

O comando `describe-stacks` a seguir mostra informações resumidas da pilha `myteststack`:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Resultado:

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template showing how to create a publicly accessible S3 bucket. **WARNING** This template creates an S3 bucket. You will be billed for the AWS resources used if you create a stack from this template.",
      "Tags": [],
      "Outputs": [
        {
          "Description": "Name of S3 bucket to hold website content",
          "OutputKey": "BucketName",
          "OutputValue": "myteststack-s3bucket-jssofilzie2w"
        }
      ],
      "StackStatusReason": null,
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false
    }
  ]
}
```


Para obter mais informações, consulte Pilhas no Guia do usuário do AWS CloudFormation.

- Consulte detalhes da API em [DescribeStacks](#) na AWS CLI Command Reference.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
import (
    "context"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
    CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
// structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
    string) StackOutputs {
    output, err := actor.CfnClient.DescribeStacks(ctx,
        &cloudformation.DescribeStacksInput{
            StackName: aws.String(stackName),
        })
    if err != nil || len(output.Stacks) == 0 {
        log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
            stackName, err)
    }
}
```

```
stackOutputs := StackOutputs{}  
for _, out := range output.Stacks[0].Outputs {  
    stackOutputs[*out.OutputKey] = *out.OutputValue  
}  
return stackOutputs  
}
```

- Consulte detalhes da API em [DescribeStacks](#) na AWS SDK para Go API Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna um conjunto de instâncias de pilha descrevendo todas as pilhas do usuário.

```
Get-CFNStack
```

Exemplo 2: retorna uma instância de pilha descrevendo a pilha especificada

```
Get-CFNStack -StackName "myStack"
```

- Para obter detalhes da API, consulte [DescribeStacks](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna um conjunto de instâncias de pilha descrevendo todas as pilhas do usuário.

```
Get-CFNStack
```

Exemplo 2: retorna uma instância de pilha descrevendo a pilha especificada

```
Get-CFNStack -StackName "myStack"
```

- Para obter detalhes da API, consulte [DescribeStacks](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Por padrão, o comando `describe-stacks` retorna valores de parâmetros. Para impedir que valores confidenciais de parâmetros, como senhas, sejam retornados, inclua uma propriedade `NoEcho` definida como `TRUE` nos modelos do CloudFormation.

Important

O uso do atributo `NoEcho` não mascara informações armazenadas no seguinte:

- A seção de modelo de Metadata. O CloudFormation não transforma, modifica nem edita nenhuma informação incluída na seção Metadata. Para obter mais informações, consulte [Metadados](#).
- A seção de modelo de Outputs. Para obter mais informações, consulte [Saídas](#).
- O atributo Metadata de uma definição de recurso. Para obter mais informações, consulte o atributo [Metadata](#).

É altamente recomendável não usar esses mecanismos para incluir informações confidenciais, como senhas ou segredos.

Important

Em vez de incorporar informações confidenciais diretamente em modelos do CloudFormation, recomendamos usar os parâmetros dinâmicos no modelo da pilha para fazer referência a informações confidenciais que são armazenadas e gerenciadas de forma externa ao CloudFormation, como no AWS Systems Manager Parameter Store ou no AWS Secrets Manager.

Para obter mais informações, consulte a prática recomendada [Não incorporar credenciais nos seus modelos](#).

Obter um modelo

CLI

AWS CLI

Para visualizar o corpo do modelo de uma pilha do AWS CloudFormation

O comando `get-template` a seguir mostra o modelo da pilha `myteststack`:

```
aws cloudformation get-template --stack-name myteststack
```

Resultado:

```
{
  "TemplateBody": {
    "AWSTemplateFormatVersion": "2010-09-09",
    "Outputs": {
      "BucketName": {
        "Description": "Name of S3 bucket to hold website content",
        "Value": {
          "Ref": "S3Bucket"
        }
      }
    },
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if
you create a stack from this template.",
    "Resources": {
      "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
          "AccessControl": "PublicRead"
        }
      }
    }
  }
}
```

- Consulte detalhes da API em [GetTemplate](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna o modelo associado à pilha especificada.

```
Get-CFNTemplate -StackName "myStack"
```

- Consulte detalhes da API em [GetTemplate](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna o modelo associado à pilha especificada.

```
Get-CFNTemplate -StackName "myStack"
```

- Para obter detalhes da API, consulte [GetTemplate](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Listar recursos de pilha

CLI

AWS CLI

Para listar os recursos de uma pilha

O comando a seguir exibe a lista dos recursos da pilha especificada.

```
aws cloudformation list-stack-resources \  
  --stack-name my-stack
```

Resultado:

```
{  
  "StackResourceSummaries": [  
    {  
      "LogicalResourceId": "bucket",  
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",  
      "ResourceType": "AWS::S3::Bucket",  
      "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",  
      "ResourceStatus": "CREATE_COMPLETE",  
      "DriftInformation": {  
        "StackResourceDriftStatus": "IN_SYNC"  
      }  
    },  
    {  
      "LogicalResourceId": "function",
```

```

        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
        "ResourceStatus": "UPDATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    },
    {
        "LogicalResourceId": "functionRole",
        "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
        "ResourceType": "AWS::IAM::Role",
        "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
        "ResourceStatus": "CREATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
]
}

```

- Consulte detalhes da API em [ListStackResources](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna as descrições de todos os recursos associados à pilha especificada.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Consulte detalhes da API em [ListStackResources](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna as descrições de todos os recursos associados à pilha especificada.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Para obter detalhes da API, consulte [ListStackResources](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Listar pilhas

Use o comando [list-stacks](#) para listar pilhas. Para listar apenas pilhas com os códigos de status especificados, inclua a opção `--stack-status-filter`. Você pode especificar um ou mais códigos de status da pilha para a opção `--stack-status-filter`. Para obter mais informações, consulte [Códigos de status da pilha](#).

CLI

AWS CLI

Para listar as pilhas do AWS CloudFormation

O comando `list-stacks` a seguir mostra um resumo de todas as pilhas que têm um status de `CREATE_COMPLETE`:

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Resultado:

```
[
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket:
Sample template showing how to create a publicly accessible S3 bucket.
**WARNING** This template creates an S3 bucket. You will be billed for the AWS
resources used if you create a stack from this template.",
    "StackStatusReason": null,
    "CreationTime": "2013-08-26T03:27:10.190Z",
    "StackName": "myteststack",
    "StackStatus": "CREATE_COMPLETE"
  }
]
```

- Consulte detalhes da API em [ListStacks](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: retorna informações resumidas de todas as pilhas.

```
Get-CFNStackSummary
```

Exemplo 2: retorna informações resumidas de todas as pilhas que estão sendo criadas no momento.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Exemplo 3: retorna informações resumidas de todas as pilhas que estão sendo criadas ou atualizadas no momento.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
"UPDATE_IN_PROGRESS")
```

- Consulte detalhes da API em [ListStacks](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: retorna informações resumidas de todas as pilhas.

```
Get-CFNStackSummary
```

Exemplo 2: retorna informações resumidas de todas as pilhas que estão sendo criadas no momento.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Exemplo 3: retorna informações resumidas de todas as pilhas que estão sendo criadas ou atualizadas no momento.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
"UPDATE_IN_PROGRESS")
```

- Para obter detalhes da API, consulte [ListStacks](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Atualizar um stack

Use o comando [update-stack](#) para atualizar diretamente uma pilha. Você especifica a pilha, os recursos e os valores de parâmetro que você deseja atualizar e, se quiser usar um modelo atualizado, o nome do modelo. Para obter mais informações, consulte [Atualizar pilhas diretamente](#).

CLI

AWS CLI

Para atualizar as pilhas do AWS CloudFormation

O comando `update-stack` a seguir atualiza o modelo e os parâmetros de entrada para a pilha `mystack`:

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template --  
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

O comando `update-stack` a seguir atualiza apenas o valor do parâmetro `SubnetIDs` da pilha `mystack`. Se você não especificar um valor para o parâmetro, o valor padrão especificado no modelo será usado:

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template  
--parameters ParameterKey=KeyPairName,UsePreviousValue=true  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

O comando `update-stack` a seguir adiciona à pilha `mystack` dois tópicos de notificação de pilha:

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --  
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-  
east-1:123456789012:mytopic2"
```

Para obter mais informações, consulte [Atualizações de pilhas do AWS CloudFormation](#) no Guia do usuário do AWS CloudFormation.

- Consulte detalhes da API em [UpdateStack](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "PK1" representa o nome de um parâmetro declarado no modelo e "PV1" representa seu valor. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 2: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemplo 3: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "PK1" representa o nome de um parâmetro declarado no modelo e "PV2" representa seu valor. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 4: atualiza a pilha "myStack" com o modelo obtido no Amazon S3 e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
```

```
-TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemplo 5: atualiza a pilha "myStack" que, este exemplo pressupõe conter recursos do IAM, com o modelo obtido do Amazon S3 e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". As pilhas que contêm recursos do IAM exigem que você especifique o parâmetro -Capabilities "CAPABILITY_IAM", senão haverá falha na atualização com o erro "InsufficientCapabilities".

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
    -Capabilities "CAPABILITY_IAM"
```

- Consulte detalhes da API em [UpdateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "PK1" representa o nome de um parâmetro declarado no modelo e "PV1" representa seu valor. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 2: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemplo 3: atualiza a pilha "myStack" com o modelo e os parâmetros de personalização especificados. "PK1" representa o nome de um parâmetro declarado no modelo e "PV2" representa seu valor. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemplo 4: atualiza a pilha "myStack" com o modelo obtido no Amazon S3 e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue".

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
    templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemplo 5: atualiza a pilha "myStack" que, este exemplo pressupõe conter recursos do IAM, com o modelo obtido do Amazon S3 e os parâmetros de personalização especificados. "K1" e "PK2" representam os nomes dos parâmetros declarados no modelo, "PV1" e "PV2" representam os valores solicitados. Os parâmetros de personalização também podem ser especificados usando "Key" e "Value" em vez de "ParameterKey" e "ParameterValue". As pilhas que contêm recursos do IAM exigem que você especifique o parâmetro -Capabilities "CAPABILITY_IAM", senão haverá falha na atualização com o erro "InsufficientCapabilities".

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
    templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
    -Capabilities "CAPABILITY_IAM"
```

- Para obter detalhes da API, consulte [UpdateStack](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Note

Para remover todas as notificações, especifique [] para a opção `--notification-arns`.

Validar seu modelo

Use o comando [validate-template](#) para verificar se há erros de sintaxe em seu arquivo de modelo.

Durante a validação, o CloudFormation primeiro verifica se o modelo é JSON válido. Caso não seja, o CloudFormation verifica se o modelo é YAML válido. Se as duas verificações falharem, o CloudFormation retornará um erro de validação de modelo.

CLI

AWS CLI

Para validar um modelo do AWS CloudFormation

O comando `validate-template` a seguir valida o modelo `sampletemplate.json`:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Resultado:

```
{
  "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
showing how to create a publicly accessible S3 bucket. **WARNING** This template
creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
  "Parameters": [],
  "Capabilities": []
}
```

Para obter mais informações, consulte [Trabalhar com modelos do AWS CloudFormation](#) no Guia do usuário do AWS CloudFormation.

- Consulte detalhes da API em [ValidateTemplate](#) na AWS CLI Command Reference.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: valida o conteúdo do modelo especificado. A saída detalha os recursos, a descrição e os parâmetros do modelo.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Exemplo 2: valida o modelo especificado acessado por meio de uma URL do Amazon S3. A saída detalha os recursos, a descrição e os parâmetros do modelo.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- Consulte detalhes da API em [ValidateTemplate](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V4).

Ferramentas para o PowerShell V5

Exemplo 1: valida o conteúdo do modelo especificado. A saída detalha os recursos, a descrição e os parâmetros do modelo.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Exemplo 2: valida o modelo especificado acessado por meio de uma URL do Amazon S3. A saída detalha os recursos, a descrição e os parâmetros do modelo.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- Para obter detalhes da API, consulte [ValidateTemplate](#) na Referência de cmdlets do Ferramentas da AWS para PowerShell (V5).

Esta é uma resposta de exemplo que produz um erro de validação.

```
{  
  "ResponseMetadata": {  
    "RequestId": "4ae33ec0-1988-11e3-818b-e15a6df955cd"  
  },  
}
```

```
"Errors": [
  {
    "Message": "Template format error: JSON not well-formed. (line 11, column
8)",
    "Code": "ValidationError",
    "Type": "Sender"
  }
],
"Capabilities": [],
"Parameters": []
}
```

A client error (ValidationError) occurred: Template format error: JSON not well-formed. (line 11, column 8)

Note

O comando `validate-template` tem o objetivo de verificar apenas a sintaxe do modelo. Ele não garante que valores de propriedades especificados para um recurso sejam válidos para este último. Ele também não determina o número de recursos existentes quando a pilha for criada.

Para verificar a validade operacional, é necessário tentar criar a pilha. Não há uma sandbox ou uma área de teste para pilhas do CloudFormation. Por isso, haverá cobranças pelos recursos que você criar durante os testes.

Fazer upload de artefatos locais em um bucket do S3 com a AWS CLI

Você pode usar a AWS CLI para carregar artefatos locais referenciados por um modelo do CloudFormation para um bucket do Amazon S3. Artefatos locais são arquivos aos quais você faz referência no seu modelo. Em vez de carregar manualmente os arquivos em um bucket do S3 e depois adicionar seus locais ao seu modelo, você pode especificar artefatos locais no seu modelo e usar o comando [package](#) para carregá-los rapidamente.

Um artefato local é um caminho para um arquivo ou pasta que o comando `package` faz upload no Amazon S3. Por exemplo, um artefato pode ser um caminho local para o código-fonte da função do AWS Lambda ou um arquivo OpenAPI da API REST do Amazon API Gateway.

Ao usar o comando `package`:

- Se você especificar um arquivo, o comando faz upload diretamente no bucket do S3.

- Se você especificar uma pasta, o comando cria um arquivo `.zip` para a pasta e, depois, faz upload do arquivo `.zip`.
- Se você não especificar um caminho, o comando criará um arquivo `.zip` para o diretório de trabalho e fará upload dele.

Você pode especificar um caminho absoluto ou relativo, no qual o caminho relativo é relativo para o local do modelo.

Depois de fazer upload dos artefatos, o comando retorna uma cópia de seu modelo, substituindo referências a artefatos locais pelo local do S3 onde o comando fez upload dos artefatos. Você poderá então usar o modelo retornado para criar ou atualizar uma pilha.

Note

Você pode usar artefatos locais apenas para as propriedades de recurso com as quais o comando `package` é compatível. Consulte mais informações sobre esse comando e uma lista de propriedades de recursos compatíveis na documentação do [package](#), na [Referência de comandos da AWS CLI](#).

Pré-requisitos

Antes de começar, é necessário ter um bucket do Amazon S3 existente.

Empacotar e implantar um modelo com artefatos locais

O modelo a seguir especifica o artefato local para um código-fonte da função do Lambda. O código-fonte é armazenado na pasta `/home/user/code/lambdafunction`.

Modelo original

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
```



```
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "/home/user/code/lambdafunction"
    }
}
```

O comando [package](#) a seguir cria e carrega um arquivo .zip da pasta do código-fonte da função para a raiz do bucket especificado.

```
aws cloudformation package \
  --s3-bucket amzn-s3-demo-bucket \
  --template /path_to_template/template.json \
  --output-template-file packaged-template.json \
  --output json
```

O comando gera um novo modelo no caminho especificado por `--output-template-file`. Ele substitui a referência do artefato pelo local do Amazon S3, conforme mostrado abaixo. O arquivo .zip é nomeado usando a soma de verificação MD5 do conteúdo da pasta, em vez de usar o próprio nome da pasta.

Modelo resultante

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "s3://amzn-s3-demo-bucket/md5_checksum"
      }
    }
  }
}
```

Depois de empacotar os artefatos do seu modelo, implante o modelo processado usando o comando [deploy](#).

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name
```

Ao implantar modelos maiores que 51.200 bytes, use o comando `deploy` com a opção `--s3-bucket` para carregar seu modelo no S3, como no exemplo a seguir.

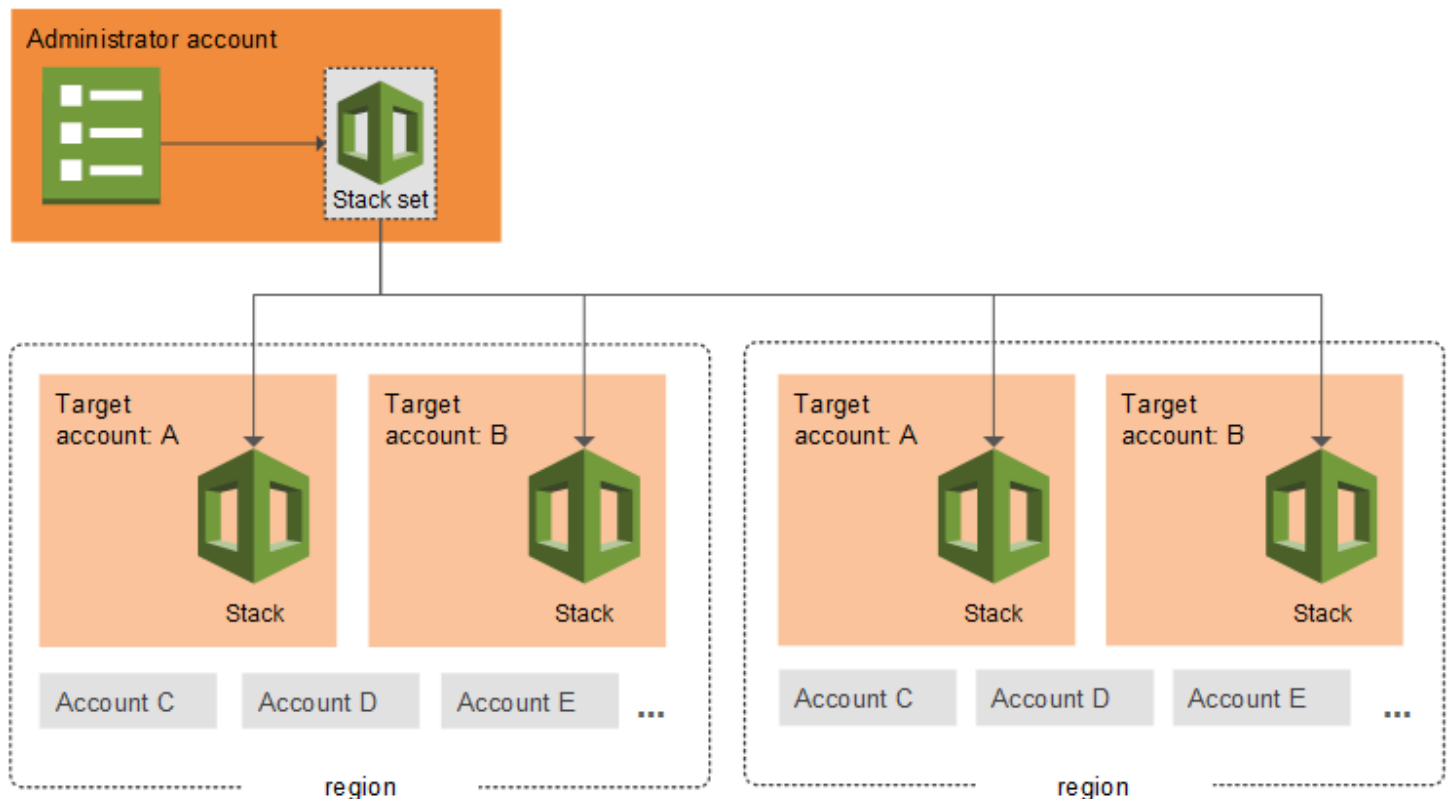
```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name \  
  --s3-bucket amzn-s3-demo-bucket
```

 Note

Para ver outro exemplo de uso do comando `package` para carregar artefatos locais, consulte [Dividir um modelo em partes reutilizáveis usando pilhas aninhadas](#).

Gerenciar pilhas em todas as contas e regiões com StackSets

O CloudFormation StackSets amplia a capacidade das pilhas, permitindo que você crie, atualize ou exclua pilhas em várias contas e Regiões da AWS com uma única operação. Usando uma conta de administrador, você define e gerencia um modelo do CloudFormation e usa-o como base para o provisionamento de pilhas para as contas de destino selecionadas nas várias Regiões da AWS especificadas.



Esta seção é para ajudar você a começar a usar StackSets. Aqui, você encontrará respostas às perguntas frequentes sobre como trabalhar e solucionar problemas de criação, atualização e exclusão de StackSets.

Tópicos

- [Conceitos do StackSets](#)
- [Pré-requisitos para usar o CloudFormation StackSets](#)
- [Começar a usar StackSets com um modelo de exemplo](#)
- [Criar CloudFormation StackSets com permissões autogerenciadas](#)

- [Criar CloudFormation StackSets com permissões gerenciadas pelo serviço](#)
- [Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations](#)
- [Atualizar o CloudFormation StackSets](#)
- [Adicionar pilhas ao CloudFormation StackSets](#)
- [Substituir valores de parâmetros nas pilhas do StackSet do CloudFormation](#)
- [Excluir pilhas do CloudFormation StackSets](#)
- [Excluir CloudFormation StackSets](#)
- [Evitar implantações de StackSets com falha usando portas de conta de destino](#)
- [Escolher o modo de simultaneidade para o CloudFormation StackSets](#)
- [Detectar desvios em StackSets do CloudFormation](#)
- [Importar pilhas para o CloudFormation StackSets](#)
- [Práticas recomendadas para usar o CloudFormation StackSets](#)
- [Modelos de exemplo do CloudFormation StackSets](#)
- [Solução de problemas de CloudFormation StackSets](#)

Conceitos do StackSets

Os conceitos e terminologia a seguir são fundamentais para o entendimento e uso de StackSets.

Tópicos

- [Contas de administrador e de destino](#)
- [CloudFormation StackSets](#)
- [Modelos de permissões para StackSets](#)
- [Instâncias da pilha](#)
- [Operações de StackSets](#)
- [Opções de operação de StackSets](#)
- [Tags](#)
- [Códigos de status do StackSets](#)
- [Códigos de status da instância de pilha](#)

Contas de administrador e de destino

Uma conta de administrador é a Conta da AWS em que você cria StackSets. Para StackSets com permissões gerenciadas pelo serviço, a conta de administrador é a conta de administrador delegado ou a conta gerencial da organização. Você pode gerenciar um StackSet fazendo login na conta de administrador da AWS que criou o StackSet.

A conta de destino é a conta em que você cria, atualiza ou exclui uma ou mais pilhas no seu StackSet. Antes de usar um StackSet para criar pilhas em uma conta de destino, estabeleça uma relação de confiança entre as contas de administrador e de destino.

CloudFormation StackSets

Um StackSet atua como um contêiner para várias pilhas que são implantadas em regiões e Contas da AWS específicas. Cada pilha é baseada no mesmo modelo do CloudFormation, mas você pode personalizar pilhas individuais usando parâmetros.

Depois de definir um StackSet, você poderá criar, atualizar ou excluir pilhas nas Regiões da AWS e contas de destino especificadas. Ao criar, atualizar ou excluir pilhas, também é possível especificar preferências de operação. Por exemplo, inclua a ordem das regiões em que você deseja realizar a operação, o limite de tolerância a falhas antes que as operações de pilha sejam interrompidas e o número de contas que realizam operações de pilha ao mesmo tempo.

Um StackSet é um recurso regional. Se você criar um StackSet em uma Região da AWS, apenas poderá vê-lo ou alterá-lo ao visualizar essa região.

Modelos de permissões para StackSets

Você pode criar StackSets usando permissões autogerenciadas ou permissões gerenciadas pelo serviço.

Com permissões autogerenciadas você cria os perfis do IAM necessários para o StackSets implantar em contas e regiões. Esses perfis são necessários para estabelecer um relacionamento confiável entre a conta em que você está administrando o StackSet e a conta em que você está implantando instâncias de pilha. Usando esse modelo de permissões, o StackSets pode implantar em qualquer Conta da AWS na qual você tenha permissões para criar um perfil do IAM.

Com permissões gerenciadas pelo serviço você pode implantar instâncias de pilha em contas gerenciadas pelo AWS Organizations. Usando esse modelo de permissões, você não precisa criar

os perfis do IAM necessários. O StackSets cria perfis do IAM em seu nome. Com esse modelo, você também pode ativar implantações automáticas em contas que serão adicionadas à sua organização no futuro.

O AWS Organizations integra-se ao CloudFormation e ajuda a gerenciar e governar centralmente seu ambiente à medida que você escala e expande seus recursos da AWS.

- Conta de gerenciamento: a conta que você usa para criar a organização. Para obter mais informações, consulte [Conceitos e terminologia do AWS Organizations](#).
- Administrador delegado: um serviço compatível da AWS pode registrar uma conta-membro da AWS da organização como administrador para as contas da organização nesse serviço. Para obter mais informações, consulte [Serviços da AWS que você pode usar com o AWS Organizations](#).

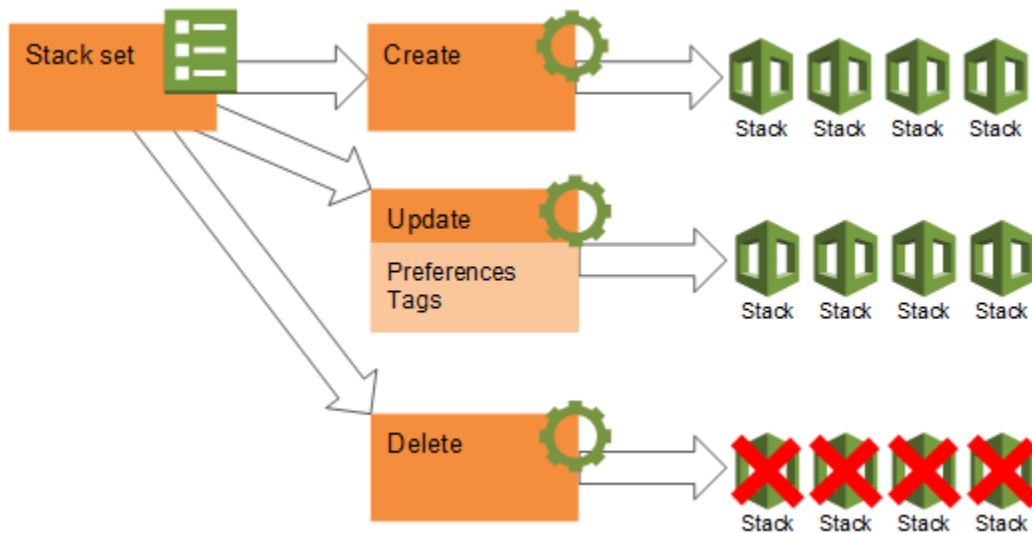
Para obter mais informações sobre como criar e gerenciar StackSets com permissões gerenciadas pelo serviço, consulte os seguintes tópicos:

- [Ativar o acesso confiável para StackSets com o AWS Organizations](#)
- [Registrar uma conta de membro de administrador delegado](#)
- [Criar CloudFormation StackSets com permissões gerenciadas pelo serviço](#)

Instâncias da pilha

Uma instância de pilha é uma referência a uma pilha em uma conta de destino dentro de uma região. Uma instância de pilha pode existir sem uma pilha. Por exemplo, se a pilha não puder ser criada por algum motivo, a instância de pilha mostrará o motivo da falha ao criar a pilha. Uma instância de pilha está associada a apenas um StackSet.

Confira a seguir as relações lógicas entre StackSets, operações de pilha e pilhas. Ao atualizar um StackSet, todas as instâncias de pilha associadas serão atualizadas em todas as contas e regiões.



Operações de StackSets

Você pode executar as operações a seguir em StackSets.

Criar um StackSet

A criação de um StackSet inclui a especificação de um modelo do CloudFormation que você usará para criar pilhas, especificando as contas de destino em que deseja criá-las e as Regiões da AWS em que deseja implantá-las nas suas contas de destino. Um StackSet garante uma implantação consistente dos mesmos recursos de pilha, com as mesmas configurações, para todas as contas de destino especificadas nas regiões escolhidas.

Atualizar um StackSet

Ao atualizar um StackSet, você envia as mudanças para as pilhas no seu StackSet. Você pode atualizar um StackSet de uma das maneiras a seguir. Suas atualizações de modelo sempre afetam todas as pilhas; não é possível atualizar o modelo seletivamente para algumas pilhas no StackSet, mas não para outras.

- Altere as configurações existentes no modelo ou adicione novos recursos, como atualização de configurações de parâmetros para um serviço específico ou adição de novas instâncias do Amazon EC2.
- Substitua o modelo por um modelo diferente.
- Adicione as pilhas a contas de destino existentes/adicionais ou regiões existentes/adicionais.

Excluir pilhas

Ao excluir pilhas, você remove uma pilha e todos os recursos associados das contas de destino e nas regiões que especificou. Você pode excluir suas pilhas das seguintes maneiras.

- Exclua pilhas de algumas contas de destino, enquanto deixa outras pilhas em outras contas de destino em execução.
- Exclua as pilhas de algumas regiões e deixe pilhas em outras regiões em execução.
- Exclua pilhas do seu StackSet, mas salve-as para que continuem sendo executadas independentemente do seu StackSet. Para isso, basta escolher a opção Reter pilhas. Você pode gerenciar as pilhas retidas fora do seu StackSet no CloudFormation.
- Exclua todas as pilhas do seu StackSet, em preparação para excluir todo o StackSet.

Excluir um StackSet

Você só poderá excluir seu StackSet quando não houver instâncias de pilha nele.

Opções de operação de StackSets

As opções descritas nesta seção ajudam você a controlar o tempo e o número de falhas permitidas para executar com sucesso as operações do StackSet e evitar a perda dos recursos de pilhas.

Máximo de contas concorrentes

Com esta configuração (disponível na criação, atualização e exclusão de fluxos de trabalho), é possível especificar a quantidade máxima ou porcentagem de contas de destino em que uma operação é executada ao mesmo tempo. Uma quantidade ou porcentagem inferior significa que uma operação será realizada em menos contas de destino ao mesmo tempo. As operações são realizadas em uma região de cada vez, na ordem especificada na caixa Deployment order (Ordem de implantação). Por exemplo, se você estiver implantando pilhas em 10 contas de destino dentro de duas regiões, a definição de Maximum concurrent accounts (Máximo de contas simultâneas) como 50 e By percentage (Por porcentagem) implanta as pilhas em cinco contas na primeira região, depois, nas próximas cinco contas na primeira região até passar para a próxima região e iniciar a implantação nas primeiras cinco contas de destino.

Ao escolher Por porcentagem, se a porcentagem especificada não representar um número inteiro das suas contas especificadas, o CloudFormation arredondará o valor para baixo. Por exemplo, se você estiver implantando pilhas em 10 contas de destino e definir Máximo de contas

simultâneas como 25 e Por porcentagem, o CloudFormation arredondará o valor de implantação de 2,5 pilhas simultaneamente (o que não seria possível) para 2 pilhas simultaneamente.

Configuração permite que você especifique o máximo para as operações. Para grandes implementações, em determinadas circunstâncias, o número real de contas atuadas simultaneamente pode ser menor devido à limitação do serviço.

O Máximo de contas simultâneas pode depender do valor da Tolerância a falhas, dependendo do Modo de simultaneidade. Se o Modo de simultaneidade estiver definido como Tolerância rigorosa a falhas, o Máximo de contas simultâneas poderá ser no máximo uma a mais do que a configuração de Tolerância a falhas.

Modo de simultaneidade

Essa configuração, disponível nos fluxos de trabalho de criação, atualização e exclusão, permite que você escolha como o nível de simultaneidade se comportará durante as operações do StackSet. Para obter mais informações, consulte [Escolher o modo de simultaneidade para o CloudFormation StackSets](#).

Tolerância a falhas

Esta configuração (disponível na criação, atualização e exclusão de fluxos de trabalho) permite especificar o número máximo ou porcentagem de falhas de operação da pilha que podem ocorrer, por região, além de qual CloudFormation interrompe uma operação automaticamente. Uma quantidade ou porcentagem menor significa que a operação será executada em menos pilhas, mas que você poderá começar a solucionar as operações com falha mais rapidamente. Por exemplo, se você estiver atualizando 10 pilhas em 10 contas de destino em três regiões, a definição de Failure tolerance (Tolerância a falhas) como 20 e By percentage (Por porcentagem) significa que um máximo de duas atualizações de pilhas em uma região poderá falhar para que a operação continue. Se uma terceira pilha na mesma região falhar, o CloudFormation interromperá a operação. Se não for possível atualizar uma pilha na primeira região, a operação de atualização continuará nela e depois passará para a próxima. Se não for possível atualizar duas pilhas na segunda região, a tolerância a falhas de 20% será atingida. Se uma terceira pilha na região falhar, o CloudFormation interromperá a operação de atualização e não passará para as regiões subsequentes.

Ao escolher Por porcentagem, se a porcentagem especificada não representar um número inteiro das pilhas em cada região, o CloudFormation arredondará o valor para baixo. Por exemplo, se você estiver implantando pilhas em 10 contas de destino em três regiões e definir Tolerância a

falhas como 25 e Por porcentagem, o CloudFormation arredondará a tolerância de falha de 2,5 pilhas (o que não seria possível) para 2 pilhas por região.

Reter pilhas

Essa configuração, disponível nos fluxos de trabalho de exclusão de pilha, permite que você mantenha as pilhas e os recursos em execução mesmo após terem sido excluídas de um StackSet. Quando você retém pilhas, o CloudFormation deixa as pilhas intactas nas contas e regiões individuais. As pilhas são desassociadas do StackSet, mas a pilha e seus recursos são salvos. Após concluir a operação de exclusão de pilhas, você gerenciará as pilhas retidas no CloudFormation, usando a conta de destino (e não da conta de administrador) que as criou.

Simultaneidade de região

Essa configuração, disponível nos fluxos de trabalho de criação, atualização e exclusão, permite escolher como os StackSets são implantados em regiões.

Sequential (Sequencial): implanta a operação de StackSets em uma região por vez, conforme especificado pela caixa Deployment order (Ordem de implantação) da região, desde que as falhas de implantação de uma região não excedam uma tolerância a falhas especificada. A implantação sequencial é a seleção padrão.

Paralela: implanta as operações do StackSets em todas as regiões especificadas simultaneamente, desde que as falhas de implantação de uma região não excedam uma tolerância a falhas especificada.

Tags

Você pode adicionar tags durante as operações de criação e atualização do StackSet, especificando pares de chave/valor. As tags são úteis para classificar e filtrar recursos do StackSet para faturamento e alocação de custos. Para obter mais informações sobre como usar tags na AWS, consulte [Organizing and tracking costs using AWS cost allocation tags](#) no Gerenciamento de Faturamento e Custos da AWS User Guide. Depois de especificar o par chave-valor, escolha + para salvar a tag. Você pode excluir tags que não está mais usando selecionando o X vermelho à direita de uma tag.

As tags que você aplica aos StackSets são aplicadas a todas as pilhas e aos recursos criados por elas. Também é possível adicionar tags no nível somente da pilha no CloudFormation, mas talvez elas não sejam mostradas no StackSets.

Embora o StackSets não adicione tags definida pelo sistema, você não deve iniciar os nomes de chaves de nenhuma tag com a string `aws` :

Códigos de status do StackSets

O CloudFormation StackSets gera códigos de status para operações de StackSets.

A tabela a seguir descreve os códigos de status das operações de StackSets.

RUNNING

A operação está em andamento.

SUCCEEDED

A operação terminou sem exceder a tolerância a falhas.

FAILED

A quantidade de pilhas em que a operação não pôde ser concluída excedeu a tolerância a falhas definida pelo usuário. O valor da tolerância a falhas definido para uma operação é aplicado a cada região durante as operações de criação e atualização de pilhas. Se a quantidade de pilhas com falha dentro de uma região exceder a tolerância a falhas, o status da operação na região será alterado para `FAILED`. O status da operação como um todo também será definido como `FAILED`, e o CloudFormation cancelará a operação em todas as demais regiões.

QUEUED

[Service-managed permissions] Para implantações automáticas que exigem uma sequência de operações, a operação entra em uma fila para ser executada. Por exemplo:

- Mover uma conta de uma unidade organizacional (UO) 0U1, para outra 0U2, aciona uma implantação automática. StackSets executa uma operação de exclusão para remover a instância de pilha da conta 0U1 de destino na região de destino e enfileira uma operação de criação para adicionar uma instância de pilha à conta 0U2 de destino na região de destino.
- A adição de uma conta AccountA a uma UO aciona uma implantação automática. O StackSets executa uma operação de criação para adicionar uma instância de pilha à AccountA na região de destino. Se você adicionar outra conta AccountB à UO enquanto essa operação de criação estiver em execução, o StackSets enfileirá uma segunda operação de criação. Quando a primeira operação de criação estiver concluída, o StackSets executará a segunda operação de criação para adicionar uma instância de pilha à AccountB na região de destino.

STOPPING

A pedido do usuário, a operação está em processo de interrupção.

STOPPED

A pedido do usuário, a operação foi interrompida.

Códigos de status da instância de pilha

O CloudFormation StackSets gera códigos de status para instâncias de pilhas.

A tabela a seguir descreve os códigos de status para instâncias de pilhas no StackSets.

CURRENT

A pilha está atualizada com o StackSet.

OUTDATED

A pilha não está atualizada com o StackSet por um dos motivos a seguir.

- Uma operação [CreateStackSet](#) ou [UpdateStackSet](#) na pilha associada falhou.
- A pilha fazia parte de uma operação [CreateStackSet](#) ou [UpdateStackSet](#) que falhou, ou que foi interrompida antes da criação ou atualização da pilha.

INOPERABLE

Uma operação [DeleteStackInstances](#) falhou e deixou a pilha em um estado instável. As pilhas neste estado são excluídas de outras operações [UpdateStackSet](#). Pode ser necessário executar uma operação [DeleteStackInstances](#), com `RetainStacks` definido como `true`, para excluir a instância de pilha e, em seguida, excluir a pilha manualmente.

CANCELLED

A operação na conta e região especificadas foi cancelada. Isso ocorre porque um usuário interrompeu a operação do StackSet ou porque as operações do StackSet excederam a tolerância a falhas.

FAILED

Falha na operação na conta e região especificadas. Se houver falha na operação do StackSet em contas suficientes dentro de uma região, a tolerância a falhas para a operação do StackSet como um todo poderá ser excedida.

FAILED_IMPORT

Houve falha na importação da instância da pilha na conta e na região especificadas, deixando a pilha em um estado instável. Depois que os problemas que causaram a falha são corrigidos, a operação de importação pode ser repetida. Se houver bastante falhas em operações de StackSets em contas suficientes de uma região, a tolerância a falhas para a operação do StackSet como um todo poderá ser excedida.

PENDING

A operação na conta e região especificadas ainda não foi iniciada.

RUNNING

A operação na conta e região especificadas está em andamento no momento.

SKIPPED_SUSPENDED_ACCOUNT

A operação na conta e região especificadas foi ignorada porque a conta foi suspensa no momento da operação.

SUCCEEDED

A operação na conta e região especificadas foi concluída com êxito.

Pré-requisitos para usar o CloudFormation StackSets

Os StackSets estendem a funcionalidade das pilhas para que você possa criar, atualizar ou excluir pilhas de várias contas e regiões com uma única operação.

Como o StackSets realiza operações de pilha em várias contas, antes de começar a criar o seu primeiro StackSet, você precisará ter as permissões necessárias definidas nas suas Contas da AWS.

Você pode gerenciar StackSets usando permissões autogerenciadas ou gerenciadas pelo serviço.

- Para StackSets autogerenciados, você deve criar e gerenciar perfis do IAM em cada conta e Região da AWS de destino. Para obter mais informações, consulte [Conceder permissões autogerenciadas](#).
- Para StackSets gerenciados pelo serviço, você não precisa criar e gerenciar manualmente perfis do IAM em cada conta; a AWS trata da criação de perfis e permissões para você. Para obter mais informações, consulte [Ativar o acesso confiável](#).

Tópicos

- [Preparar-se para realizar de operações de StackSets em Regiões da AWS desabilitadas por padrão](#)
- [Conceder permissões autogerenciadas](#)
- [Ativar o acesso confiável para StackSets com o AWS Organizations](#)

Preparar-se para realizar de operações de StackSets em Regiões da AWS desabilitadas por padrão

As Regiões da AWS introduzidas após 20 de março de 2019, como Ásia-Pacífico (Hong Kong), estão desabilitadas por padrão. Você deve habilitar essas regiões para sua(s) conta(s) para poder usá-las. Para obter informações consulte [Habilitar ou desabilitar Regiões da AWS na sua conta](#) no Guia de referência do AWS Gerenciamento de contas.

Para criar um StackSet em uma conta de administrador de StackSet (se estiver usando permissões autogerenciadas) ou em uma conta gerencial da organização (se estiver usando permissões gerenciadas pelo serviço) em uma região desabilitada por padrão, primeiro você deve habilitar essa região para a conta de administrador ou gerencial.

Para que o CloudFormation crie ou atualize com êxito uma instância de pilha:

- A conta de destino deve residir em uma região atualmente habilitada para essa conta de destino.
- A conta de administrador do StackSet ou a conta gerencial da organização deve estar com a mesma região habilitada que a conta de destino.

Important

Esteja ciente de que, durante as operações do StackSet, as contas de administrador e de destino trocam metadados referentes às próprias contas, bem como ao StackSet e às instâncias do StackSet envolvidas.

Além disso, caso desabilite uma região que contém uma conta em que residem instâncias do StackSet, você será responsável por excluir essas instâncias ou recursos, se desejado. Além disso, esteja ciente de que os metadados relativos à conta de destino na região desabilitada serão retidos na conta de administrador.

Conceder permissões autogerenciadas

Este tópico fornece instruções sobre como criar os perfis de serviço do IAM requeridos pelo StackSets para a implantação em contas e em Regiões da AWS com permissões autogerenciadas. Esses perfis são necessários para estabelecer um relacionamento confiável entre a conta em que você está administrando o StackSet e a conta em que você está implantando instâncias de pilha. Usando esse modelo de permissões, o StackSets pode implantar em qualquer Conta da AWS na qual você tenha permissões para criar um perfil do IAM.

Para usar permissões gerenciadas pelo serviço, consulte [Ativar o acesso confiável](#).

Tópicos

- [Visão geral das permissões autogerenciadas](#)
- [Concessão de permissões para o gerenciamento de pilhas em todas as contas de destino a todos os usuários da conta do administrador](#)
- [Definir opções de permissões avançadas para operações de StackSets](#)
- [Configurar chaves globais para atenuar problemas de confused deputy](#)

Visão geral das permissões autogerenciadas

Antes de criar um StackSet com permissões autogerenciadas, é necessário ter criado perfis de serviço do IAM em cada conta.

As etapas básicas são:

1. Determine qual Conta da AWS é a conta do administrador.

Os StackSets serão criados nessa conta de administrador. Uma conta de destino é a conta em que você cria as pilhas individuais que pertencem a um StackSet.

2. Determine como você deseja estruturar as permissões para o StackSet.

A configuração de permissões mais simples (e mais permissiva) é oferecer a todos os usuários e grupos na conta de administrador a capacidade de criar e atualizar todos os StackSets gerenciados por meio da conta. Se precisar de um controle mais refinado, defina permissões para especificar:

- Quais usuários e grupos podem executar operações do StackSet em quais contas de destino.
- Quais recursos os usuários e grupos podem incluir em seus StackSets.

- Quais operações do StackSet os usuários e grupos específicos podem executar.
3. Crie as funções de serviço do IAM necessárias nas contas de administrador e de destino para definir as permissões desejadas.

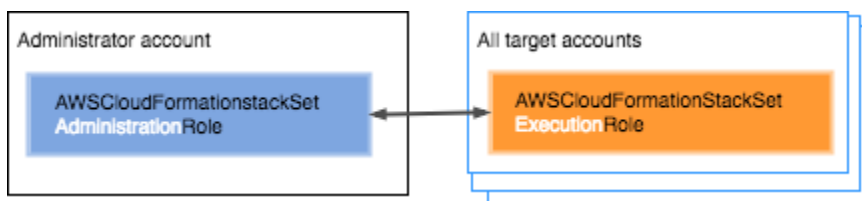
Especificamente, os dois perfis necessários são:

- `AWSCloudFormationStackSetAdministrationRole`: este perfil é implantado na conta do administrador.
- `AWSCloudFormationStackSetExecutionRole`: este perfil é implantado em todas as contas nas quais você cria instâncias de pilha.

Concessão de permissões para o gerenciamento de pilhas em todas as contas de destino a todos os usuários da conta do administrador

Esta seção mostra como configurar permissões para permitir que todos os usuários e os grupos da conta do administrador executem operações do StackSet em todas as contas de destino. A seção orientará você na criação dos perfis de serviço do IAM necessários nas contas de administrador e de destino. Qualquer pessoa que usa a conta do administrador pode criar, atualizar ou excluir pilhas em qualquer uma das contas de destino.

Ao estruturar permissões dessa maneira, os usuários não passam um perfil de administração ao criar ou atualizar um StackSet.



Administrator account

Na conta do administrador, crie uma função do IAM denominada `AWSCloudFormationStackSetAdministrationRole`.

É possível fazer isso ao criar uma pilha usando o modelo do CloudFormation disponível em <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.yml>.

Example Exemplo de política de permissões

O perfil de administração criado pelo modelo disponibilizado anteriormente inclui a política de permissões apresentada a seguir.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example Exemplo de política de confiança 1

O modelo disponibilizado anteriormente também inclui a política de confiança, apresentada a seguir, que concede ao serviço a permissão para usar o perfil de administração e as permissões anexadas ao perfil.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Example Exemplo de política de confiança 2

Para implantar instâncias de pilha em uma conta de destino que reside em uma região desabilitada por padrão, você também deverá incluir a entidade principal de serviço regional para essa região. Cada região desabilitada por padrão terá sua própria entidade principal de serviço regional.

O exemplo de política de confiança apresentado a seguir concede ao serviço a permissão para usar o perfil de administração na região Ásia-Pacífico (Hong Kong) (ap-east-1), uma região que está desabilitada por padrão.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obter mais informações, consulte [Preparar-se para realizar de operações de StackSets em Regiões da AWS desabilitadas por padrão](#). Para obter uma lista de códigos de região, consulte [Regional endpoints](#) no Referência geral da AWS Guide.

Target accounts

Em cada conta de destino, crie uma função de serviço chamada `AWSCloudFormationStackSetExecutionRole` que confie na conta do administrador. Essa função deve ter esse nome exato. É possível fazer isso ao criar uma pilha usando o modelo do CloudFormation disponível em <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.yml>. Ao usar este modelo, você será solicitado a fornecer o ID da conta do administrador com a qual sua conta de destino deve ter uma relação de confiança.

⚠ Important

Lembre-se de que este modelo concede acesso de administrador. Depois de usar o modelo para criar uma função de execução da conta de destino, você deve definir o escopo das permissões na declaração de política para os tipos de recursos que você criará usando o StackSets.

O perfil de serviço da conta de destino requer permissões para executar quaisquer operações especificadas no modelo do CloudFormation. Por exemplo, se o seu modelo estiver criando um bucket do S3, então você precisará de permissões para criar novos objetos para o S3. A conta de destino sempre precisa da concessão de permissões totais por parte do CloudFormation, que incluem permissões para criar, atualizar, excluir e descrever pilhas.

Example Exemplo 1 de política de permissões

A função criada por este modelo habilita a seguinte política em uma conta de destino.

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

Example Exemplo 2 de política de permissões

O exemplo a seguir mostra uma declaração de política com as permissões mínimas para que o StackSets funcione. Para criar pilhas em contas de destino que usam recursos de serviços que são diferentes dos recursos usados pelo CloudFormation, é necessário adicionar essas ações e esses recursos de serviço à instrução de política `AWSCloudFormationStackSetExecutionRole` para cada conta de destino.

```
{
```

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

Example Exemplo de política de confiança

A seguinte relação de confiança é criada pelo modelo. O ID da conta do administrador é mostrado como *admin_account_id*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Você pode configurar a relação de confiança de uma função de execução da conta de destino existente para confiar em uma função específica na conta do administrador. Se você excluir a função na conta de administrador e criar uma nova para substituí-la, deve configurar suas relações de confiança da conta de destino com a função da nova conta de administrador, representada por *admin_account_id* no exemplo anterior.

Definir opções de permissões avançadas para operações de StackSets

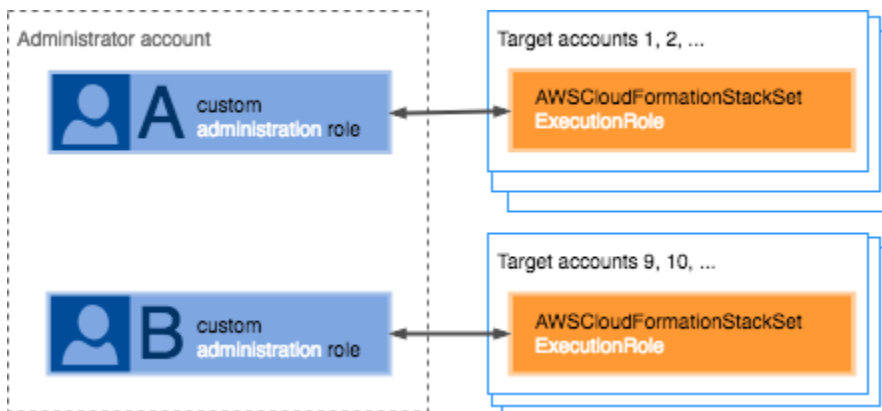
Se precisar de um controle mais refinado sobre os StackSets que os usuários e os grupos criam por meio de uma única conta de administrador, use perfis do IAM para especificar:

- Quais usuários e grupos podem executar operações do StackSet em quais contas de destino.
- Quais recursos os usuários e grupos podem incluir em seus StackSets.
- Quais operações do StackSet os usuários e grupos específicos podem executar.

Controlar quais usuários podem executar operações de StackSets em contas de destino específicas

Use os perfis de administração personalizados para controlar quais usuários e grupos podem executar operações de StackSets em diferentes contas de destino. Você pode querer controlar quais usuários da conta de administrador podem executar as operações de StackSets em quais contas de destino. Para fazer isso, crie uma relação de confiança entre cada conta de destino e uma função de administração personalizada específica, em vez de criar a função de serviço `AWSCloudFormationStackSetAdministrationRole` na própria conta de administrador. Você então ativa usuários e grupos específicos para usar o perfil de administração personalizado ao executar operações de StackSets em uma determinada conta de destino.

Por exemplo, você pode criar uma função A e uma função B na sua conta de administrador. Você pode conceder à função A permissões para acessar as contas de destino de 1 a 8. Você pode conceder à função B permissões para acessar as contas de destino de 9 a 16.



A configuração das permissões necessárias envolve a definição de um perfil de administração personalizado, a criação de um perfil de serviço para a conta de destino e a concessão de permissão aos usuários para passar o perfil de administração personalizado ao executar as operações de StackSets.

Em geral, uma vez que você já tenha as permissões necessárias, é desta forma que funciona: ao criar um StackSet, o usuário deve especificar um perfil de administração personalizado. O usuário deve ter a permissão para transferir o perfil para o CloudFormation. Além disso, o perfil de administração personalizado deve ter uma relação de confiança com as contas de destino especificadas para o StackSet. O CloudFormation cria o StackSet e associa o perfil de administração personalizado a ele. Ao atualizar um StackSet, o usuário deve especificar um perfil de administração personalizado de forma explícita, mesmo que seja o mesmo perfil de administração personalizado usado anteriormente com esse StackSet. O CloudFormation usa esse perfil para atualizar a pilha, de acordo com os requisitos apresentados acima.

Administrator account

Example Exemplo de política de permissões

Para cada StackSet, crie um perfil de administração personalizado com permissões para assumir o perfil de execução da conta de destino.

O nome do perfil de execução da conta de destino deve ser semelhante para todas as contas de destino. Se o nome do perfil for `AWSCloudFormationStackSetExecutionRole`, o StackSets o usará automaticamente ao criar um StackSet. Se você especificar um nome de perfil personalizado, os usuários deverão fornecer o nome do perfil de execução ao criar um StackSet.

Crie um [perfil de serviço do IAM](#) com um nome personalizado e a política de permissões apresentada a seguir. Nos exemplos abaixo, *custom_execution_role* refere-se ao perfil de execução nas contas de destino.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/custom_execution_role"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Para especificar diversas contas em uma única instrução, separe-as usando vírgulas.

```
"Resource": [  
  "arn:aws:iam::target_account_id_1:role/custom_execution_role",  
  "arn:aws:iam::target_account_id_2:role/custom_execution_role"  
]
```

É possível especificar todas as contas de destino ao usar um caractere curinga (*) em vez de um ID da conta.

```
"Resource": [  
  "arn:aws:iam::*:role/custom_execution_role"  
]
```

Example Exemplo de política de confiança 1

Você deve fornecer uma política de confiança para o perfil de serviço com a finalidade de definir quais entidades principais do IAM podem assumir o perfil.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "cloudformation.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

Example Exemplo de política de confiança 2

Para implantar instâncias de pilha em uma conta de destino que reside em uma região desabilitada por padrão, você também deverá incluir a entidade principal de serviço regional para essa região. Cada região desabilitada por padrão terá sua própria entidade principal de serviço regional.

O exemplo de política de confiança apresentado a seguir concede ao serviço a permissão para usar o perfil de administração na região Ásia-Pacífico (Hong Kong) (ap-east-1), uma região que está desabilitada por padrão.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obter mais informações, consulte [Preparar-se para realizar de operações de StackSets em Regiões da AWS desabilitadas por padrão](#). Para obter uma lista de códigos de região, consulte [Regional endpoints](#) no Guia de referência geral da AWS.

Example Exemplo de política para transferência do perfil

Você também precisa de uma política de permissões do IAM para os usuários do IAM que permita que o usuário passe o perfil de administração personalizado ao executar operações de StackSets. Para obter mais informações, consulte [Conceder a um usuário permissões para transmitir uma função a um produto da AWS](#).

No exemplo abaixo, *customized_admin_role* refere-se ao perfil de administração que o usuário precisa transferir.

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



```
    "Action": [
      "iam:GetRole",
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::*:role/customized_admin_role"
  }
]
```

Target accounts

Em cada conta de destino, crie um perfil de serviço que tenha uma relação de confiança com o perfil de administração personalizado que você deseja usar com essa conta.

O perfil da conta de destino requer permissões para executar quaisquer operações especificadas no modelo do CloudFormation. Por exemplo, se o seu modelo estiver criando um bucket do S3, você precisará de permissões para criar novos objetos no S3. A conta de destino sempre precisa da concessão de permissões totais por parte do CloudFormation, que incluem permissões para criar, atualizar, excluir e descrever pilhas.

O nome do perfil da conta de destino deve ser semelhante para todas as contas de destino. Se o nome do perfil for `AWSCloudFormationStackSetExecutionRole`, o StackSets o usará automaticamente ao criar um StackSet. Se você especificar um nome de perfil personalizado, os usuários deverão fornecer o nome do perfil de execução ao criar um StackSet.

Example Exemplo de política de permissões

O exemplo a seguir mostra uma declaração de política com as permissões mínimas para que o StackSets funcione. Para criar pilhas em contas de destino que usam recursos de serviços que são diferentes dos recursos usados pelo CloudFormation, é necessário adicionar essas ações e esses recursos de serviço à política de permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
```

```

    "Resource": "*"
  }
]
}

```

Example Exemplo de política de confiança

Você deve fornecer a política de confiança apresentada a seguir ao criar o perfil para definir a relação de confiança.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

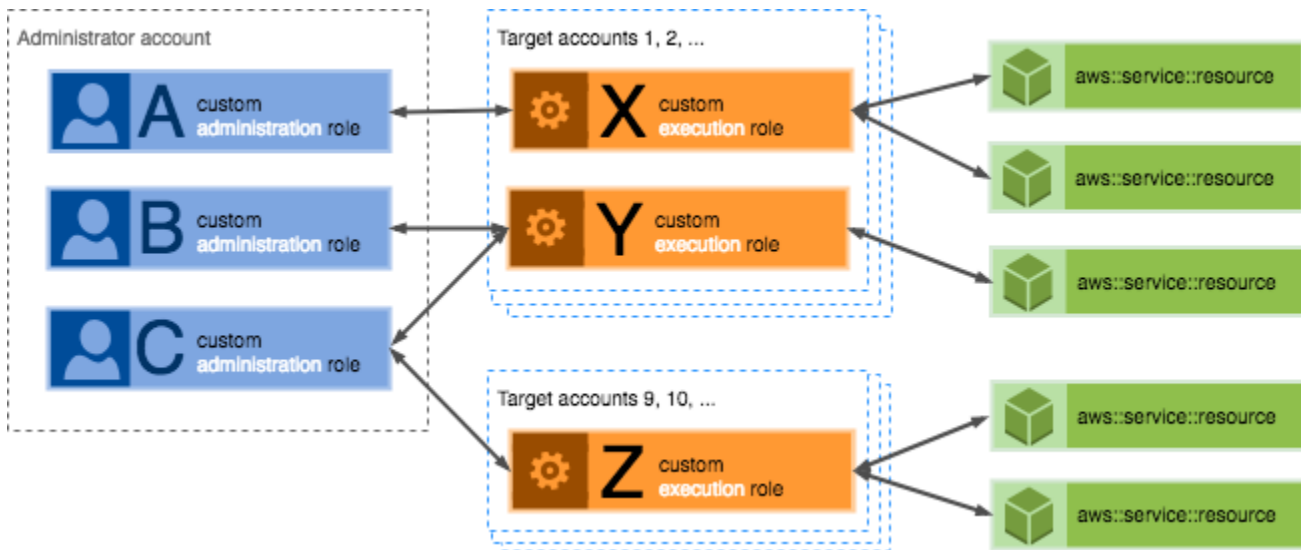
```

Controlar os recursos que os usuários podem incluir em StackSets específicos

Use perfis de execução personalizados para controlar quais recursos de pilha os usuários e grupos podem incluir em seus StackSets. Por exemplo, você pode querer configurar um grupo que só pode incluir recursos relacionados ao Amazon S3 nos StackSets criados, enquanto outra equipe só pode incluir recursos do DynamoDB. Para fazer isso, você cria uma relação de confiança entre o perfil de administração personalizado para cada grupo e um perfil de execução personalizado para cada conjunto de recursos. O perfil de execução personalizado define quais recursos de pilha podem ser incluídos nos StackSets. O perfil de administração personalizado reside na conta do administrador, enquanto o perfil de execução personalizado reside em cada conta de destino em que você deseja criar StackSets usando os recursos definidos. Você então ativa usuários e grupos específicos para usar o perfil de administração personalizado ao executar operações de StackSets.

Por exemplo, é possível criar os perfis de administração personalizados A, B e C na conta do administrador. Usuários e grupos com permissão para usar o Perfil A podem criar StackSets que

contêm os recursos de pilha especificamente listados no perfil de execução personalizado X, mas não aqueles nos perfis Y ou Z, ou os recursos não incluídos em qualquer perfil de execução.



Ao atualizar um StackSet, o usuário deve especificar um perfil de administração personalizado de forma explícita, mesmo que seja o mesmo perfil de administração personalizado usado anteriormente com esse StackSet. O CloudFormation executa a atualização usando o perfil de administração personalizado especificado, desde que o usuário tenha as permissões para executar operações nesse StackSet.

Da mesma forma, o usuário também pode especificar uma função de execução personalizada. Se um perfil de execução personalizado for especificado, o CloudFormation usará esse perfil para atualizar a pilha, de acordo com os requisitos apresentados acima. Se o usuário não especificar um perfil de execução personalizado, o CloudFormation realizará a atualização usando o perfil de execução personalizado anteriormente associado ao StackSet, desde que o usuário tenha as permissões para executar operações nesse StackSet.

Administrator account

Crie um perfil de administração personalizado em sua conta de administrador, conforme detalhado em [Controlar quais usuários podem executar operações de StackSets em contas de destino específicas](#). Inclua uma relação de confiança entre o perfil de administração personalizado e os perfis de execução personalizados que você deseja usar.

Example Exemplo de política de permissões

O exemplo apresentado a seguir corresponde a uma política de permissões para `AWSCloudFormationStackSetExecutionRole` definido para a conta de destino, além de um perfil de execução personalizado.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487980684000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole",
        "arn:aws:iam::*:role/custom_execution_role"
      ]
    }
  ]
}
```

Target accounts

Nas contas de destino em que deseja criar seus StackSets, crie um perfil de execução personalizado que conceda permissões aos serviços e recursos que você deseja que os usuários e grupos possam incluir nos StackSets.

Example Exemplo de política de permissões

O exemplo a seguir fornece as permissões mínimas para StackSets, junto com a permissão para criar tabelas do Amazon DynamoDB.

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

```

    "Action": [
      "cloudformation:*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:createTable"
    ],
    "Resource": "*"
  }
]
}

```

Example Exemplo de política de confiança

Você deve fornecer a política de confiança apresentada a seguir ao criar o perfil para definir a relação de confiança.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Definir permissões para operações específicas de StackSets

Além disso, você pode definir permissões para quais usuários e grupos podem executar operações específicas do StackSet, como criação, atualização ou exclusão de StackSets ou instâncias de pilha. Para obter mais informações, consulte [Actions, resources, and condition keys for CloudFormation](#) na Service Authorization Reference.

Configurar chaves globais para atenuar problemas de confused deputy

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição global [aws:SourceArn](#) e [aws:SourceAccount](#) nas políticas de recursos para limitar as permissões que o CloudFormation StackSets concede a outro serviço para o recurso. Se você utilizar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta `aws:SourceArn` no valor deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

A maneira mais eficaz de se proteger do problema ‘confused deputy’ é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se especificar vários recursos, use a chave de contexto de condição global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:cloudformation::123456789012:*`. Sempre que possível, use `aws:SourceArn`, pois ela é mais específica. Use `aws:SourceAccount` apenas quando não puder determinar o ARN ou padrão de ARN correto.

Quando o StackSets assume a função de Administração na sua conta de administrador, ele preenche o ID dessa conta de administrador e nome do recurso da Amazon (ARN) do StackSets. Portanto, é possível definir condições para [chaves globais](#) `aws:SourceAccount` e `aws:SourceArn` nas relações de confiança para evitar [problemas de confused deputy](#). O exemplo a seguir mostra como é possível usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` em StackSets para evitar o problema “confused deputy”.

Administrator account

Example Chaves globais para **aws:SourceAccount** e **aws:SourceArn**

Ao usar o StackSets, defina as chaves globais `aws:SourceAccount` e `aws:SourceArn` na política de confiança `AWSCloudFormationStackSetAdministrationRole` para evitar problemas de confused deputy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
        }
      }
    }
  ]
}
```

Example ARNs de StackSets

Especifique seus ARNs de StackSets associados para um controle mais definido.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": [
```

```

    "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-1",
    "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-2"
  ]
}
}
}
]
}

```

Ativar o acesso confiável para StackSets com o AWS Organizations

Este tópico fornece instruções sobre como ativar o acesso confiável com o AWS Organizations, que é requerido pelo StackSets para realizar implantações em contas e em Regiões da AWS usando permissões gerenciadas pelo serviço. Para usar permissões autogerenciadas, consulte [Conceder permissões autogerenciadas](#).

Antes de criar um StackSet com permissões gerenciadas pelo serviço, você deve primeiro concluir as seguintes tarefas:

- [Habilitar todos os recursos](#) no AWS Organizations. Apenas com os recursos de faturamento consolidado habilitados, não é possível criar um StackSet com permissões gerenciadas pelo serviço.
- Ativa o acesso confiável com AWS Organizations. Essa ação permite que o CloudFormation crie um perfil vinculado ao serviço na conta de gerenciamento. Depois que o acesso confiável é ativado, quando você cria um StackSet com permissões gerenciadas pelo serviço, o CloudFormation cria tanto o perfil vinculado ao serviço necessário quanto um perfil de serviço denominado `stacksets-exec-*` nas contas de destino (membro).

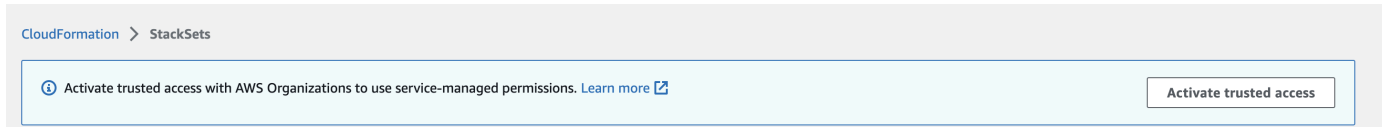
Com o acesso confiável ativado, a conta gerencial e as contas de administrador delegado podem criar e gerenciar StackSets gerenciados pelo serviço para sua organização.

Para ativar o acesso confiável, você deve ser um usuário administrador na conta de gerenciamento. Um usuário administrador é um usuário que tem todas as permissões em sua conta da Conta da AWS. Para obter mais informações, consulte [Criar um usuário administrador](#) no Guia de administração do AWS Gerenciamento de contas. Para obter recomendações para proteger

a segurança da conta de gerenciamento, consulte [Práticas recomendadas para a conta de gerenciamento](#) no Manual do usuário do AWS Organizations.

Para ativar o acesso confiável

1. Faça login na AWS como administrador da conta de gerenciamento e abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets. Se o acesso confiável estiver desativado, um banner será exibido solicitando que você ative o acesso confiável.



3. Escolha Ativar acesso confiável.

O acesso confiável será ativado com êxito quando o banner a seguir for exibido.



Note

Ativar o acesso às organizações é o mesmo que Habilitar o acesso às organizações, e Desativar o acesso às organizações é o mesmo que Desabilitar o acesso às organizações. Esses termos foram atualizados com base nas diretrizes de marketing.

Para desativar o acesso confiável

Consulte [CloudFormation StackSets e AWS Organizations](#) no Guia do usuário do AWS Organizations.

Antes de desativar o acesso confiável com o AWS Organizations, você deve cancelar o registro de todos os administradores delegados. Para obter mais informações, consulte [Registro de um administrador delegado](#).

Note

Para obter informações sobre o uso de operações de API em vez do console para ativar ou desativar o acesso confiável, consulte:

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

Perfis vinculados ao serviço

A conta de gerenciamento usa o perfil vinculado ao serviço

AWSServiceRoleForCloudFormationStackSetsOrgAdmin. Você só poderá modificar ou excluir esse perfil se o acesso confiável com o AWS Organizations estiver desativado.

Cada conta de destino usa um perfil vinculado ao serviço

AWSServiceRoleForCloudFormationStackSetsOrgMember. Você poderá modificar ou excluir esse perfil apenas com duas condições: se o acesso confiável com o AWS Organizations estiver desativado ou se a conta for removida da organização de destino ou da unidade organizacional (UO).

Para mais informações, consulte [CloudFormation StackSetsAWS Organizations](#) no Guia do usuário do AWS Organizations.

Registrar uma conta de membro de administrador delegado

Além da conta gerencial da sua organização, as contas de membro com permissões de administrador delegado podem criar e gerenciar StackSets com permissões gerenciadas pelo serviço para a organização. Os StackSets com permissões gerenciadas pelo serviço são criados na conta gerencial, incluindo os StackSets criados por administradores delegados. Para ser registrado como administrador delegado para sua organização, sua conta de membro deve estar na organização. Para obter mais informações sobre o ingresso em uma organização, consulte [Inviting an Conta da AWS to join your organization](#).

Sua organização pode ter até cinco administradores delegados registrados de uma só vez. Os administradores delegados podem optar por implantar em todas as contas da sua organização ou OUs específicas. O acesso confiável com o AWS Organizations deve ser ativado antes que os administradores delegados possam implantar em contas gerenciadas pelo Organizations. Para obter mais informações, consulte [Ativar o acesso confiável para StackSets com o AWS Organizations](#).

 Important

Esteja ciente do seguinte:

- Os administradores delegados têm permissões completas para implantar em contas em sua organização. A conta gerencial não pode limitar as permissões de administrador delegado para implantar em UOs específicas ou para executar operações específicas do StackSet.
- Certifique-se de que seus administradores delegados tenham permissões `organizations:ListDelegatedAdministrators` para evitar possíveis erros.

É possível registrar administradores delegados para sua organização nas seguintes regiões: Leste dos EUA (Ohio), Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Canadá (Central), Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Paris), Europa (Estocolmo), Israel (Tel Aviv), América do Sul (São Paulo), AWS GovCloud (Leste dos EUA) e AWS GovCloud (Oeste dos EUA).

Você pode registrar e cancelar o registro de administradores delegados usando o [CloudFormation console](#), [a AWS CLI](#) ou [SDKs](#) da AWS.

Para registrar um administrador delegado (console)

1. Faça login na AWS como administrador da conta gerencial e abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation/>.
2. No painel de navegação, escolha StackSets.
3. Em Delegated administrators (Administradores delegados), escolha Register delegated administrator (Registrar administrador delegado).
4. Na caixa de diálogo Register delegated administrator (Registrar administrador delegado), escolha Register delegated administrator (Registrar administrador delegado).

A mensagem de sucesso indica que a conta de membro foi registrada com êxito como administrador delegado.

Para cancelar o registro de um administrador delegado (console)

1. Faça login na AWS como administrador da conta gerencial e abra o console do CloudFormation em <https://console.aws.amazon.com/>.
2. No painel de navegação, escolha StackSets.
3. Em Delegated administrators (Administradores delegados), selecione a conta que deseja cancelar o registro e escolha Deregister (Cancelar registro).

A mensagem de sucesso indica que a conta de membro teve o registro cancelado de administrador delegado com êxito.

Você pode registrar esta conta novamente a qualquer momento.

Para registrar um administrador delegado do (AWS CLI)

1. Abra a AWS CLI.
2. Execute o comando `register-delegated-administrator`.

```
$ aws organizations register-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Execute o comando `list-delegated-administrators` para verificar se a conta de membro especificada foi registrada com êxito como administrador delegado.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Para cancelar o registro de um administrador delegado (AWS CLI)

1. Abra a AWS CLI.
2. Execute o comando `deregister-delegated-administrator`.

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Execute o comando `list-delegated-administrators` para verificar se a conta de membro especificada foi cancelada com êxito como administrador delegado.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Você pode registrar esta conta novamente a qualquer momento.

Começar a usar StackSets com um modelo de exemplo

Este tutorial ajudará você a começar a usar StackSets com o Console de gerenciamento da AWS. Ele vai orientar você na criação de um StackSet usando um modelo de exemplo. Você aprenderá a implantar pilhas em várias regiões, a monitorar as operações do StackSet e a visualizar os resultados.

Neste tutorial, você vai criar um StackSet que habilitará o AWS Config em sua Conta da AWS nas regiões Oeste dos EUA (Oregon) (us-west-2) e Leste dos EUA (Norte da Virgínia) (us-east-1). Com StackSets, você pode criar, atualizar ou excluir pilhas em várias contas e regiões com uma única operação, o que o torna a solução ideal para gerenciar a infraestrutura em grande escala. Embora este tutorial use uma única conta para simplificar, ele demonstra de forma eficaz os recursos multirregionais de StackSets.

O modelo de exemplo está disponível no seguinte bucket do S3: <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfig.yml>.

Note

Os StackSets são gratuitos, mas você será cobrado pelos recursos da AWS que criar com eles, como o AWS Config neste tutorial. Para obter mais informações, consulte [Preços do AWS Config](#).

Tópicos

- [Pré-requisitos](#)
- [Criar um StackSet com um modelo de exemplo do console](#)
- [Monitorar a criação de um StackSet](#)
- [Visualizar os resultados do StackSet](#)

- [Atualizar seu StackSet](#)
- [Adicionar pilhas ao seu StackSet](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Antes de começar este tutorial, certifique-se de ter cumprido os seguintes pré-requisitos:

- Você deve ter configurado os perfis do IAM necessários para permissões autogerenciadas. Para criar um StackSet e implantar pilhas em uma única conta, você precisa dos seguintes perfis em sua conta:
 - `AWSCloudFormationStackSetAdministrationRole`
 - `AWSCloudFormationStackSetExecutionRole`

Para obter instruções detalhadas sobre como configurar esses perfis, consulte [Conceder permissões autogerenciadas](#).

Criar um StackSet com um modelo de exemplo do console

Para criar um StackSet que habilite o AWS Config

1. Abra o [console do CloudFormation](#).
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você deseja gerenciar o StackSet.

Você pode escolher qualquer região compatível com StackSets. A região que você selecionar não afetará as regiões em que você pode implantar com seu StackSet.

3. No painel de navegação, escolha StackSets.
4. Na parte superior da página StackSets, escolha Criar StackSet.
5. Em Permissões, escolha Permissões de autoatendimento e depois os perfis do IAM que você criou nos pré-requisitos.
 - Para o perfil de administrador do IAM, escolha `AWSCloudFormationStackSetAdministrationRole`.

- Para o nome do perfil de execução do IAM, escolha `AWSCloudFormationStackSetExecutionRole`.
6. Em Prerequisite - Prepare template (Pré-requisito - preparar modelo), escolha Use a sample template (Usar um modelo de exemplo).
 7. Em Selecionar um modelo de exemplo, escolha o modelo Habilitar AWS Config. Em seguida, escolha Próximo.

Esse modelo criará os recursos necessários para habilitar o AWS Config em sua conta, incluindo um gravador de configuração e um canal de entrega.

8. Na página Especificar detalhes do StackSet, em Nome do StackSet, insira **my-awsconfig-stackset**.
9. Em Descrição do StackSet, insira **A StackSet that enables Config across multiple Regions**.
10. Em Parâmetros, configure as configurações do AWS Config da seguinte forma:
 - a. Em Compatível com todos os tipos de recursos, mantenha o valor padrão, true, para registrar todos os tipos de recursos compatíveis.
 - b. Em Incluir os tipos de recursos globais, mantenha o valor padrão, false, para excluir os recursos globais, como os perfis do IAM.
 - c. Deixe Lista de tipos de recursos se nem todos forem compatíveis definido como <All>.
 - d. Em A região que contém o recurso de perfil vinculado ao serviço de configuração, substitua <DeployToAnyRegion> por **us-west-2**.

Isso significa que o perfil vinculado ao serviço denominado `AWSServiceRoleForConfig` será criado apenas se houver uma pilha implantada na região Oeste dos EUA (Oregon). Você escolherá as regiões de implantação mais adiante neste procedimento.

- e. Em Frequência de registro do gravador de configuração, escolha gravação DIÁRIA.
11. Escolha Próximo para continuar.
12. Na página Configurar opções do StackSet, escolha Adicionar nova tag e adicione uma tag ao especificar um par de chave/valor:
 - a. Em Chave, digite **Stage**.
 - b. Em Valor, insira **Test**.

As tags que você aplica aos StackSets são aplicadas a todos os recursos criados pelas suas pilhas.

13. Em Configuração de execução, escolha Ativo para habilitar o tratamento otimizado de operações do CloudFormation:

- As operações não conflitantes são executadas simultaneamente para tempos de implantação mais rápidos.
- As operações conflitantes são automaticamente colocadas em fila e processadas na ordem em que foram solicitadas.

Quando há operações sendo executadas ou na fila, o CloudFormation coloca na fila todas as operações recebidas, mesmo que elas não gerem conflitos. Você não pode alterar as configurações de execução durante esse período.

14. Escolha Próximo.

15. Na página Definir opções de implantação, em Adicionar pilhas ao StackSet, escolha Implantar novas pilhas.

16. Em Accounts (Contas), escolha Deploy stacks in accounts (Implantar pilhas em contas).

17. Na caixa de texto, insira o ID da Conta da AWS.

18. Em Especificar regiões, selecione as seguintes regiões nesta ordem:

1. Região Oeste dos EUA (Oregon) (us-west-2)
2. Região Leste dos EUA (Norte da Virgínia) (us-east-1)

Se necessário, clique na seta para cima ao lado de região Oeste dos EUA (Oregon) a fim de movê-la para ser a primeira entrada na lista. A ordem das regiões determina a ordem de implantação.

19. Em Opções de implantação, configure as seguintes configurações:

- a. Em Maximum concurrent accounts (Máximo de contas concorrentes), mantenha os valores padrão de Number (Número) e 1.

Para implantações de várias contas, essa configuração significa que o CloudFormation implantará sua pilha em apenas uma conta de cada vez.

- b. Em Failure tolerance (Tolerância a falhas), mantenha os padrões de Number (Número) e 0.

Isso significa que nenhuma implantação de pilha poderá falhar em uma das suas regiões especificadas antes que o CloudFormation interrompa a implantação na região atual e cancele as implantações nas regiões restantes.

- c. Em Simultaneidade de regiões, escolha Sequencial (padrão).

Essa configuração garante que o CloudFormation conclua as implantações em uma região antes de passar para a próxima.

- d. Em Modo de simultaneidade, mantenha o padrão de Tolerância a falhas estrita.

Para implantações com várias contas, isso reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.

20. Escolha Próximo.

21. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.

22. Quando estiver tudo pronto para criar seu StackSet, escolha Enviar.

Monitorar a criação de um StackSet

Depois de escolher Enviar, o CloudFormation começa a criar seu StackSet e a implantar pilhas nas regiões especificadas em sua conta. A página de detalhes do StackSet é aberta automaticamente, em que você pode monitorar o progresso da operação.

Para monitorar a criação do StackSet

1. Na página de detalhes do StackSet, a guia Operações é exibida por padrão, mostrando a operação atual em andamento.
2. O status da operação deve ser inicialmente RUNNING. O CloudFormation cria pilhas nas regiões que você especificou de acordo com as opções de implantação configuradas.
3. Para ver mais detalhes sobre a operação, selecione o ID da operação na lista.
4. Na página de detalhes da operação, você pode visualizar o status das instâncias de pilha que estão sendo criadas em cada região.
5. Aguarde até que o status da operação mude para SUCCEEDED, o que indica que o StackSet e todas as suas instâncias de pilha foram criados com êxito.

Visualizar os resultados do StackSet

Depois que a criação do StackSet for concluída, você poderá visualizar as instâncias de pilha implantadas e verificar se o AWS Config foi habilitado em sua conta nas regiões especificadas.

Para visualizar os resultados do StackSet

1. Na página de detalhes do StackSet, escolha a guia Instâncias de pilha.
2. Você deve ver uma lista de instâncias de pilha que foram criadas em sua conta nas regiões especificadas. Cada instância de pilha deve ter um status SUCCEEDED, indicando que foi implantada com êxito.
3. Para verificar se o AWS Config está habilitado em sua conta, você pode verificar o console do AWS Config em cada uma das regiões implantadas.

Atualizar seu StackSet

Depois de criar seu StackSet, você pode querer atualizá-lo para modificar os valores dos parâmetros ou adicionar mais regiões. Esta seção mostra como atualizar o parâmetro de frequência de gravação do AWS Config.

Para atualizar seu StackSet

1. Na página StackSets, selecione seu **my-awsconfig-stackset**.
2. Com o StackSet selecionado, escolha Editar detalhes do StackSet no menu Ações.
3. Na página Escolher um modelo, em Pré-requisito - Preparar modelo, escolha Usar modelo atual.
4. Escolha Próximo.
5. Na página Especificar detalhes do StackSet, em Parâmetros, encontre a Frequência de registro do gravador de configuração e altere-a de DIÁRIA para CONTÍNUA.
6. Escolha Próximo.
7. Na página Configurar opções do StackSet, deixe as configurações como estão e escolha Avançar.
8. Na página Definir opções de implantação, especifique o ID da sua conta e as mesmas regiões que você usou ao criar o StackSet.
9. Para Opções de implantação, mantenha as mesmas configurações de antes.
10. Escolha Próximo.

11. Na página Analisar, analise suas alterações e escolha Enviar.
12. O CloudFormation começa a atualizar seu StackSet. Você pode monitorar o progresso na guia Operações da página de detalhes do StackSet.

Adicionar pilhas ao seu StackSet

Você pode adicionar mais pilhas ao seu StackSet implantando em regiões adicionais. Esta seção mostra como adicionar pilhas a uma nova região.

Para adicionar pilhas ao seu StackSet

1. Na página StackSets, selecione seu **my-awsconfig-stackset**.
2. Com o StackSet selecionado, escolha Adicionar pilhas ao StackSet no menu Ações.
3. Na página Definir opções de implantação, em Adicionar pilhas ao StackSet, escolha Implantar novas pilhas.
4. Em Contas, escolha Implantar pilhas em contas e insira o ID da sua conta.
5. Em Especificar regiões, selecione uma nova região, como Europa (Irlanda) (eu-west-1).
6. Para Opções de implantação, mantenha as mesmas configurações de antes.
7. Escolha Próximo.
8. Na página Especificar substituições, deixe os valores da propriedade conforme especificados e escolha Avançar.
9. Na página Review (Revisar), reveja suas escolhas e selecione Submit (Enviar).
10. O CloudFormation começa a criar novas pilhas na região especificada. Você pode monitorar o progresso na guia Operações da página de detalhes do StackSet.

Limpeza

Para evitar cobranças por recursos indesejados do AWS Config, você deve fazer a limpeza excluindo as pilhas do seu StackSet, excluindo o próprio StackSet e removendo os perfis do IAM que você criou para este tutorial. Como todos os recursos são implantados em sua conta, a limpeza é simples.

Para excluir pilhas do seu StackSet

1. Na página StackSets, selecione seu **my-awsconfig-stackset**.
2. Com o StackSet selecionado, escolha Excluir pilhas do StackSet no menu Ações.

3. Na página Definir opções de implantação, em Contas, escolha Implantar pilhas em contas e insira o ID da sua conta.
4. Em Especificar regiões, selecione todas as regiões em que você implantou pilhas.
5. Em Opções de implantação, mantenha as configurações padrão.
6. Certifique-se de que Reter pilhas não esteja ativado, para que as pilhas e seus recursos sejam excluídos.
7. Escolha Próximo.
8. Na página Review (Revisar), reveja suas escolhas e selecione Submit (Enviar).
9. O CloudFormation começa a excluir as pilhas do seu StackSet. Você pode monitorar o progresso na guia Operações da página de detalhes do StackSet.

Para excluir seu StackSet

1. Depois que todas as pilhas tiverem sido excluídas, na página StackSets, selecione seu **my-awsconfig-stackset**.
2. Com o StackSet selecionado, escolha Excluir StackSet no menu Ações.
3. Quando a confirmação for solicitada, selecione Excluir.

Para excluir os perfis de serviço do IAM

Como você implantou somente na sua conta, você só precisará excluir os perfis do IAM dessa única conta, tornando a limpeza muito mais simples do que as implantações com várias contas.

1. Abra o [console do IAM](#).
2. No painel de navegação, escolha Roles (Funções).
3. Na caixa de pesquisa, insira **AWSCloudFormationStackSet** para encontrar os perfis que você criou para este tutorial.
4. Marque a caixa de seleção ao lado de **AWSCloudFormationStackSetAdministrationRole**.
5. Escolha Excluir na parte superior da página.
6. Na caixa de diálogo de confirmação, insira **delete** e escolha Excluir.
7. Repita o mesmo processo para excluir o **AWSCloudFormationStackSetExecutionRole**.

Depois de excluir o StackSet, um bucket do Amazon S3 permanecerá em cada Região da AWS devido ao atributo `DeletionPolicy` no recurso `AWS::S3::Bucket`. Isso preserva seus dados

de histórico do AWS Config. Se você não precisar mais desses dados, poderá excluir o bucket manualmente com segurança. Para excluir um bucket, primeiro exclua seu conteúdo. Esvaziar um bucket exclui todos os objetos nele contidos.

Para esvaziar e excluir os buckets do Amazon S3

1. Abra o [console Amazon S3](#).
2. No painel de navegação, no lado esquerdo do console, selecione Buckets.
3. Na lista Buckets, você verá os buckets criados para esse StackSet em cada região em que você implantou. Selecione a opção ao lado do nome do bucket criado para esse StackSet e escolha Esvaziar.
4. Na página Esvaziar bucket, confirme se deseja esvaziar o bucket digitando **permanently delete** no campo de texto, depois escolha Esvaziar.
5. Monitore o andamento do processo de esvaziamento do bucket na página Esvaziar bucket: status.
6. Para retornar à sua lista de buckets, escolha Exit (Sair).
7. Selecione a opção ao lado do nome do bucket e escolha Excluir.
8. Quando for solicitada a confirmação, digite o nome do bucket e escolha Excluir bucket.
9. Monitore o andamento do processo de exclusão do bucket na lista Buckets. Quando o Amazon S3 concluir a exclusão do bucket, remove-o da lista.
10. Repita esse processo para cada bucket criado pelo StackSet nas diferentes regiões.

Próximas etapas

Parabéns! Você criou com sucesso um StackSet usando um modelo de exemplo, implantou pilhas em várias regiões da sua conta, atualizou o StackSet, adicionou mais pilhas e limpou seus recursos. Ao se concentrar na implantação de uma única conta, você simplificou o processo de limpeza e, ao mesmo tempo, aprendeu os principais recursos multirregionais de StackSets.

Para saber mais sobre StackSets, confira os seguintes tópicos:

- [Substituir valores de parâmetros nas pilhas do StackSet do CloudFormation](#): saiba como substituir valores de parâmetros para contas e regiões específicas.
- [Criar CloudFormation StackSets com permissões gerenciadas pelo serviço](#): estude a criação de StackSets para implantações em várias contas com o AWS Organizations.

Criar CloudFormation StackSets com permissões autogerenciadas

Este tópico descreve como criar StackSets com permissões autogerenciadas para implantar pilhas em regiões e Contas da AWS.

Note

Antes de continuar, crie os perfis de serviço do IAM necessários para StackSets para estabelecer uma relação de confiança entre a conta em que você está administrando o StackSet e a conta em que você está implantando as pilhas. Para obter mais informações, consulte [Conceder permissões autogerenciadas](#).

Tópicos

- [Criar um StackSet com permissões autogerenciadas \(console\)](#)
- [Criar um StackSet com permissões autogerenciadas \(AWS CLI\)](#)

Criar um StackSet com permissões autogerenciadas (console)

Para criar um StackSet

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você deseja gerenciar o StackSet.
3. No painel de navegação, escolha StackSets.
4. Na parte superior da página StackSets, escolha Criar StackSet.
5. Em Permissões, escolha Permissões de autoatendimento e depois os perfis do IAM que você criou.
6. Em Pré-requisito: prepare o modelo, escolha O modelo está pronto.
7. Em Specify template (Especificar modelo), escolha especificar o URL para o bucket do S3 que contém o modelo de pilha ou fazer upload de um arquivo de modelo de pilha. Em seguida, escolha Próximo.
8. Na página Especificar detalhes do StackSet, forneça um nome para o StackSet, especifique quaisquer parâmetros e escolha Avançar.

9. Escolha Próximo para continuar.
10. Na página Configure StackSet options (Configurar opções do StackSet) em Tags, especifique todas as tags a serem aplicadas aos recursos em sua pilha. Para obter mais informações sobre como as tags são usadas na AWS, consulte [Organizar e acompanhar custos usando tags de alocação de custos da AWS](#) no Guia do usuário do Gerenciamento de Faturamento e Custos da AWS.
11. Em Configuração de execução, escolha Ativo para habilitar o tratamento otimizado de operações do CloudFormation:
 - As operações não conflitantes são executadas simultaneamente para tempos de implantação mais rápidos.
 - As operações conflitantes são automaticamente colocadas em fila e processadas na ordem em que foram solicitadas.

Quando há operações sendo executadas ou na fila, o CloudFormation coloca na fila todas as operações recebidas, mesmo que elas não gerem conflitos. Você não pode alterar as configurações de execução durante esse período.

12. Se o seu modelo contém recursos do IAM, em Recursos, escolha Eu reconheço que este modelo pode criar recursos do IAM para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
13. Escolha Próximo.
14. Na página Definir opções de implantação, em Adicionar pilhas ao StackSet, escolha Implantar novas pilhas.
15. Em Accounts (Contas), escolha Deploy stacks in accounts (Implantar pilhas em contas). Cole os números da sua Conta da AWS de destino na caixa de texto, separando-os com vírgulas.

 Note

Você poderá incluir o ID da sua conta de administrador se quiser implantar pilhas nessa conta também.

16. Em Especificar regiões, escolha as regiões nas quais você deseja implantar as pilhas.
17. Em Opções de implantação, faça o seguinte:

- Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
- Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.
- Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
- Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
 - Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.
 - Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.

18. Escolha Próximo.

19. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.

20. Quando estiver tudo pronto para criar seu StackSet, escolha Enviar.

O CloudFormation começa a criar seu StackSet. Visualize o progresso e o status da criação das pilhas em seu StackSet na página de detalhes do StackSet que é aberta ao escolher Enviar.

Criar um StackSet com permissões autogerenciadas (AWS CLI)

Siga as etapas nesta seção para usar o AWS CLI para:

- Crie o contêiner do StackSet.
- Implante instâncias de pilha.

Para criar um StackSet

1. Use o comando [create-stack-set](#) para criar um novo StackSet chamado *my-stackset*. O exemplo a seguir usa um modelo armazenado em um bucket do S3 e inclui um parâmetro que define um *KeyPairName* com o valor *TestKey*.

```
aws cloudformation create-stack-set \
```



```
--stack-set-name my-stackset \  
--template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
--parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Quando o comando `create-stack-set` estiver concluído, execute o comando [list-stack-sets](#) para ver qual StackSet foi criado. Você verá seu novo StackSet nos resultados.

```
aws cloudformation list-stack-sets
```

3. Use o comando [create-stack-instances](#) para implantar pilhas em seu StackSet. O exemplo a seguir implanta pilhas em duas Contas da AWS (*account_ID_1* e *account_ID_2*) em duas regiões (*us-west-2* e *us-east-1*).

Defina o processamento simultâneo de contas e outras preferências de implantação usando a opção `--operation-preferences`. Este exemplo usa configurações baseadas em contagem. Observe que `MaxConcurrentCount` não deve exceder `FailureToleranceCount` + 1. Para configurações baseadas em porcentagem, use `FailureTolerancePercentage` ou `MaxConcurrentPercentage` em vez disso.

```
aws cloudformation create-stack-instances \  
--stack-set-name my-stackset \  
--accounts account_ID_1 account_ID_2 \  
--regions us-west-2 us-east-1 \  
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para obter mais informações, consulte [CreateStackInstances](#) na AWS CloudFormation API Reference.

4. Use o comando [describe-stack-set-operation](#) para verificar se suas pilhas foram criadas com êxito. Para a opção `--operation-id`, especifique o ID da operação que foi retornado como parte da saída `create-stack-instances`.

```
aws cloudformation describe-stack-set-operation \  
--stack-set-name my-stackset \  
--operation-id operation_ID
```

Criar CloudFormation StackSets com permissões gerenciadas pelo serviço

Com permissões gerenciadas pelo serviço, você pode implantar pilhas em contas gerenciadas pelo AWS Organizations em regiões específicas. Com esse modelo, você não precisa criar perfis do IAM em cada conta de destino e Região da AWS. O CloudFormation cria os perfis do IAM em seu nome. Para obter mais informações, consulte [Ativar o acesso confiável](#).

Tópicos

- [Considerações](#)
- [Criar um StackSet com permissões gerenciadas pelo serviço \(console\)](#)
- [Criar um StackSet com permissões gerenciadas pelo serviço \(AWS CLI\)](#)

Considerações

Antes de criar um StackSet com permissões gerenciadas pelo serviço, considere o seguinte:

- Os StackSets com permissões gerenciadas pelo serviço podem ser iniciados pela conta gerencial de sua organização ou por contas de administrador delegado, mas todas as operações são executadas pela conta gerencial.
- O CloudFormation não implanta pilhas na conta gerencial, mesmo que essa conta faça parte da sua organização ou pertença a uma unidade organizacional (UO).
- O StackSet pode ter como destino toda a sua organização (inclui todas as contas) ou UOs especificadas. Quando um StackSet tem uma UO pai como destino, ele inclui automaticamente qualquer UO filha. Por padrão, quando um StackSet tem como alvo UOs específicas, ele inclui todas as contas dentro dessas UOs. No entanto, você pode ter contas específicas como destino usando as opções de filtro de contas.
- Vários StackSets podem ter como destino a mesma organização ou UO.
- Não é possível ter contas de destino de fora da sua organização.
- Sua autorização para implantar o StackSets depende das permissões atribuídas a entidade principal do IAM (usuário, perfil ou grupo) que você usa para fazer login na conta gerencial. Para obter um exemplo de política do IAM que concede permissões para implantar em uma organização, consulte [Restringir operações de conjunto de pilhas com base na região e nos tipos de recursos](#).

- Os administradores delegados têm permissões completas para implantar em qualquer conta em sua organização. A conta gerencial não pode limitar as permissões de administrador delegado para implantar em UOs ou em operações do StackSet específicas.
- As configurações de implantação automática se aplicam no nível do StackSet. Não é possível ajustar implantações automáticas seletivamente para OUs, contas ou regiões.
- Os StackSets que usam permissões gerenciadas por serviços não oferecem suporte a pilhas ou modelos aninhados que contenham macros ou transformações.

Criar um StackSet com permissões gerenciadas pelo serviço (console)

Para criar um StackSet

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você deseja gerenciar o StackSet.
3. No painel de navegação, escolha StackSets.
4. Na parte superior da página StackSets, escolha Criar StackSet.
5. Em Permissions (Permissões), escolha Service-managed permissions (Permissões gerenciadas pelo serviço).

Note

Se o acesso confiável com o AWS Organizations estiver desabilitado, será exibido um banner. É necessário o acesso confiável para criar ou atualizar um StackSet com permissões gerenciadas pelo serviço. Somente o administrador na conta de gerenciamento da organização tem permissões para [Ativar o acesso confiável para StackSets com o AWS Organizations](#).

6. Em Pré-requisito: prepare o modelo, escolha O modelo está pronto.
7. Em Specify template (Especificar modelo), escolha especificar o URL para o bucket do S3 que contém o modelo de pilha ou fazer upload de um arquivo de modelo de pilha. Em seguida, escolha Próximo.
8. Na página Especificar detalhes do StackSet, forneça um nome para o StackSet, especifique quaisquer parâmetros e escolha Avançar.

9. Na página Configure StackSet options (Configurar opções do StackSet) em Tags, especifique todas as tags a serem aplicadas aos recursos em sua pilha. Para obter mais informações sobre como as tags são usadas na AWS, consulte [Organizar e acompanhar custos usando tags de alocação de custos da AWS](#) no Guia do usuário do Gerenciamento de Faturamento e Custos da AWS.
10. Em Configuração de execução, escolha Ativo para habilitar o tratamento otimizado de operações do CloudFormation:
 - As operações não conflitantes são executadas simultaneamente para tempos de implantação mais rápidos.
 - As operações conflitantes são automaticamente colocadas em fila e processadas na ordem em que foram solicitadas.

Quando há operações sendo executadas ou na fila, o CloudFormation coloca na fila todas as operações recebidas, mesmo que elas não gerem conflitos. Você não pode alterar as configurações de execução durante esse período.

11. Se o seu modelo contém recursos do IAM, em Recursos, escolha Eu reconheço que este modelo pode criar recursos do IAM para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).
12. Escolha Próximo para continuar e habilitar o acesso confiável caso ainda não esteja habilitado.
13. Na página Definir opções de implantação, em Destinos de implantação, faça o seguinte:
 - Para implantar em todas as contas em sua organização, escolha Implantar na organização.
 - Para implantar em todas as contas em OUs específicas, escolha Implantar em unidades organizacionais (OUs). Escolha Add an OU (Adicionar uma OU) e cole o ID de destino na caixa de texto. Repita para cada nova OU de destino.

Se você escolher Implantar em unidades organizacionais (UOs), em Tipo de filtro de conta, você pode definir seus alvos de implantação como contas individuais específicas escolhendo uma das opções a seguir e fornecendo números de conta.

- Nenhuma (padrão): implantar pilhas em todas as contas nas UOs especificadas.
- Interseção: implantar pilhas apenas de determinadas contas individuais nas UOs selecionadas.

- Diferença: implantar pilhas em todas as contas nas UOs selecionadas, exceto determinadas contas.
 - União: implantar pilhas nas UOs especificadas e também em contas individuais adicionais.
14. Em Implantação automática, escolha se implantará automaticamente em contas que serão adicionadas à organização de destino ou em UOs no futuro. Para obter mais informações, consulte [Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations](#).
15. Se você habilitou a implantação automática, em Account removal behavior (Comportamento de remoção de conta), escolha se os recursos de pilha serão retidos ou excluídos quando uma conta for removida de uma organização ou OU de destino.

 Note

Com a opção Reter pilhas selecionada, as pilhas são removidas do StackSet, mas as pilhas e os recursos associados a elas são retidos. Os recursos permanecem em seu estado atual, mas não farão mais parte do StackSet.

16. Em Especificar regiões, escolha as regiões nas quais você deseja implantar as pilhas.
17. Em Opções de implantação, faça o seguinte:
- Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
 - Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.
 - Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
 - Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
 - Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.
 - Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.
 - Para dependências do StackSet, adicione ARNs dependentes do StackSet, permanecendo com dez dependências no máximo. Para obter mais informações, consulte [Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations](#).

18. Escolha Próximo para continuar.
19. Na página Analisar, verifique se o StackSet será implantado nas contas corretas nas regiões corretas e escolha Criar StackSet.

A página StackSet details (Detalhes do StackSet) é aberta. Você pode visualizar o progresso e o status da criação das pilhas em seu StackSet.

Criar um StackSet com permissões gerenciadas pelo serviço (AWS CLI)

Siga as etapas nesta seção para usar o AWS CLI para:

- Crie o contêiner do StackSet.
- Implante instâncias de pilha.

Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

Deploy to your organization

Para criar um StackSet

1. Use o comando [create-stack-set](#) para criar um novo StackSet chamado *my-stackset*. O exemplo a seguir usa um modelo armazenado em um bucket do S3, permite implantações automáticas e preserva as pilhas quando as contas são removidas. Para obter mais informações, consulte [Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations](#).

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment  
Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

2. Use o comando [list-stack-sets](#) para confirmar que seu StackSet foi criado. O novo StackSet é relacionado nos resultados.

```
aws cloudformation list-stack-sets
```

- Se você definir a opção `--call-as` como `DELEGATED_ADMIN` enquanto estiver na sua conta de membro, `list-stack-sets` retornará todas as pilhas com permissões gerenciadas pelo serviço na conta gerencial da organização.
 - Se você definir a opção `--call-as` como `SELF` enquanto estiver na sua Conta da AWS, `list-stack-sets` retornará todos os StackSets autogerenciados da sua Conta da AWS.
 - Se você definir a opção `--call-as` como `SELF` enquanto estiver na conta gerencial da organização, `list-stack-sets` retornará todas as pilhas da conta gerencial da organização.
3. Use o comando [create-stack-instances](#) para adicionar pilhas ao seu StackSet. Para a opção `--deployment-targets`, especifique o ID raiz da organização para implantar em todas as contas da sua organização.

Defina o processamento simultâneo de contas e outras preferências de implantação usando a opção `--operation-preferences`. Este exemplo usa configurações baseadas em contagem. Observe que `MaxConcurrentCount` não deve exceder `FailureToleranceCount + 1`. Para configurações baseadas em porcentagem, use `FailureTolerancePercentage` ou `MaxConcurrentPercentage` em vez disso.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=r-a1b2c3d4e5 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para obter mais informações, consulte [CreateStackInstances](#) na AWS CloudFormation API Reference.

4. Usando o `operation-id` que foi retornado como parte da saída de `create-stack-instances`, use o comando [describe-stack-set-operation](#) a seguir para verificar se as pilhas foram criadas com êxito.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to organizational units (OUs)

Para criar um StackSet

1. Use o comando [create-stack-set](#) para criar um novo StackSet chamado *my-stackset*. O exemplo a seguir usa um modelo armazenado em um bucket do S3 e inclui um parâmetro que define um *KeyPairName* com o valor *TestKey*

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Use o comando [list-stack-sets](#) para confirmar que seu StackSet foi criado. O novo StackSet é relacionado nos resultados.

```
aws cloudformation list-stack-sets
```

- Se você definir a opção `--call-as` como `DELEGATED_ADMIN` enquanto estiver na sua conta de membro, `list-stack-sets` retornará todas as pilhas com permissões gerenciadas pelo serviço na conta gerencial da organização.
 - Se você definir a opção `--call-as` como `SELF` enquanto estiver na sua Conta da AWS, `list-stack-sets` retornará todos os StackSets autogerenciados da sua Conta da AWS.
 - Se você definir a opção `--call-as` como `SELF` enquanto estiver na conta gerencial da organização, `list-stack-sets` retornará todas as pilhas da conta gerencial da organização.
3. Use o comando [create-stack-instances](#) para adicionar pilhas ao seu StackSet. Para a opção `--deployment-targets`, especifique os IDs das UOs nas quais implantar.

Defina o processamento simultâneo de contas e outras preferências de implantação usando a opção `--operation-preferences`. Este exemplo usa configurações baseadas em contagem. Observe que `MaxConcurrentCount` não deve exceder `FailureToleranceCount + 1`. Para configurações baseadas em porcentagem, use `FailureTolerancePercentage` ou `MaxConcurrentPercentage` em vez disso.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-sl1r5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=2,FailureToleranceCount=1
```



```
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para obter mais informações, consulte [CreateStackInstances](#) na AWS CloudFormation API Reference.

4. Usando o `operation-id` que foi retornado como parte da saída de `create-stack-instances`, use o comando [describe-stack-set-operation](#) a seguir para verificar se as pilhas foram criadas com êxito.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

Você pode ter unidades organizacionais (UOs) específicas como destino e usar a filtragem de contas para controlar com precisão quais contas recebem implantações de pilha. Por padrão, as pilhas serão implantadas em todas as contas dentro das UOs específicas se nenhuma filtragem de contas for especificada.

Na AWS CLI, você especifica a filtragem de contas com a opção `--deployment-targets`. Para obter mais informações, consulte [DeploymentTargets](#).

Depois de criar o contêiner do StackSet com o comando `create-stack-set`, use um dos exemplos a seguir para implantar pilhas em contas específicas.

Contas específicas como destino em uma UO

O exemplo a seguir implanta pilhas somente nas contas A1 e A2 na UO1.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets  
  OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=INTERSECTION \  
  --regions us-west-2 us-east-1
```

Excluir contas de uma UO

O exemplo a seguir implanta pilhas em todas as contas na UO1, exceto nas contas A1 e A2.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets  
  OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=INTERSECTION \  
  --regions us-west-2 us-east-1
```

```
--deployment-targets  
OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=DIFFERENCE \  
--regions us-west-2 us-east-1
```

Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations

O CloudFormation pode implantar automaticamente pilhas adicionais em novas contas do AWS Organizations quando elas são adicionadas à organização ou às unidades organizacionais (UOs) de destino. Você pode habilitar implantações automáticas e escolher se as pilhas e seus recursos associados serão excluídos ou retidos quando as contas forem removidas das UOs de destino. Essas configurações podem ser modificadas a qualquer momento em StackSets que usam permissões gerenciadas pelo serviço.

Como as implantações automáticas funcionam

Quando as implantações automáticas estão habilitadas, elas são acionadas quando as contas são adicionadas a uma organização ou UO de destino, removidas de uma organização ou UO de destino ou transferidas entre UOs de destino.

Por exemplo, considere StackSet1, que tem como destino OU1 na região *us-east-1*, e StackSet2, que tem como destino OU2 na região *us-east-1*. OU1 contém AccountA.

Se passarmos a AccountA da OU1 para a OU2 com as implantações automáticas habilitadas, o CloudFormation executará automaticamente uma operação de exclusão para remover a pilha StackSet1 da AccountA e colocará na fila uma operação de criação que adicionará a pilha StackSet2 à AccountA.

Considerações

As seguintes considerações se aplicam quando implantações automáticas são usadas:

- O recurso de implantações automáticas é habilitado no nível do StackSet. Não é possível ajustar implantações automáticas seletivamente para OUs, contas ou regiões.
- Os valores de parâmetro substituídos só se aplicam às contas que estão atualmente nas UOs de destino e nas UOs a elas subordinadas. As contas adicionadas às UOs de destino e suas UOs secundárias no futuro usarão os valores padrão do StackSet e não os valores substituídos.

- As implantações automáticas não consideram filtros de segmentação em nível de conta. Se você usar contas de destino específicas e habilitar implantações automáticas, o StackSet continuará a implantar em novas contas adicionadas na organização implantada. Para evitar implantações em contas recém-adicionadas, desabilite implantações automáticas.
- Gerenciamento de dependências: defina até 10 dependências por StackSet, com até 100 dependências por conta. Por exemplo, se você tiver cinco StackSets com cinco dependências cada, você terá 25 dependências contabilizando para o limite de 100 dependências. Você pode solicitar um aumento de limite pelo [console do Service Quotas](#). As dependências são removidas quando os StackSets são excluídos ou as organizações são desativadas.
- É recomendável habilitar a execução gerenciada ao usar implantações automáticas. A execução gerenciada permitirá que as operações de implantação automática em várias contas de destino sejam executadas simultaneamente em um StackSet, aumentando a velocidade de processamento, especialmente para organizações maiores.

Habilitar ou desabilitar implantações automáticas (console)

Para habilitar ou desabilitar implantações automáticas

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. No painel de navegação, escolha StackSets.
4. Na página StackSets, selecione a opção ao lado do nome do StackSet a ser atualizado.
5. Escolha Editar implantação automática no menu Ações, no canto superior direito.
6. Na caixa de diálogo que é aberta, faça o seguinte:
 - a. Em Implantação automática, escolha Ativada ou Desativada.
 - b. Em Comportamento de remoção de conta, escolha Excluir pilhas ou Reter pilhas. Os recursos retidos permanecerão em seu estado atual, mas não mais farão parte do StackSet.
 - c. Para dependências do StackSet, adicione ARNs dependentes do StackSet, permanecendo com dez dependências no máximo.
7. Escolha Salvar.

Habilitar ou desabilitar implantações automáticas (AWS CLI)

Para habilitar ou desabilitar implantações automáticas

1. Use o comando [update-stack-set](#) com a opção `--auto-deployment`.

O comando a seguir habilita as implantações automáticas.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

Como alternativa, para desabilitar as implantações automáticas, especifique `Enabled=false` como o valor para a opção `--auto-deployment`, como no exemplo a seguir.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment Enabled=false
```

2. Usando o ID da operação que foi retornado como parte da saída de `update-stack-set`, execute [describe-stack-set-operation](#) para verificar se o StackSet foi atualizado com êxito.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Atualizar o CloudFormation StackSets

Você pode atualizar seu StackSet usando o console do CloudFormation ou a AWS CLI.

Para adicionar e remover contas e regiões de um StackSet, consulte [Adicionar pilhas a StackSets](#) e [Excluir pilhas de StackSets](#). Para substituir os valores dos parâmetros de uma pilha, consulte [Substituir parâmetros em pilhas](#).

Tópicos

- [Atualizar seu StackSet \(console\)](#)
- [Atualizar seu StackSet \(AWS CLI\)](#)

Atualizar seu StackSet (console)

Para atualizar um StackSet

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. No painel de navegação, escolha StackSets.
4. Na página StackSets, selecione o StackSet que você deseja atualizar.
5. Com o StackSet selecionado, escolha Editar detalhes do StackSet no menu Ações.
6. Na página Escolher um modelo, atualize a seção Permissões conforme necessário ou vá para a próxima etapa.
7. Em Pré-requisito: preparar o modelo, escolha Usar modelo atual para usar o modelo atual ou Substituir modelo atual para especificar um URL do S3 para outro modelo ou fazer upload de um novo modelo.
8. Escolha Próximo.
9. Na página Especificar detalhes do StackSet, em Descrição do StackSet, atualize a descrição do StackSet conforme necessário.
10. Em Parâmetros, atualize os valores dos parâmetros conforme necessário.
11. Escolha Próximo.
12. Na página Configurar opções do StackSet, em Tags, modifique as tags conforme necessário. É possível adicionar, atualizar ou excluir tags. Para obter mais informações sobre como as tags são usadas na AWS, consulte [Organizar e acompanhar custos usando tags de alocação de custos da AWS](#) no Guia do usuário do Gerenciamento de Faturamento e Custos da AWS.
13. Em Configuração de execução, é possível atualizar a configuração de execução conforme necessário.

Note

Lembre-se de que você não pode alterar as configurações de execução quando há operações em execução ou em fila.

14. Se o seu modelo contém recursos do IAM, em Recursos, escolha Eu reconheço que este modelo pode criar recursos do IAM para especificar que você deseja usar recursos do IAM

no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).

15. Escolha Próximo.

16. Na página Definir opções de implantação, informe forneça as contas e regiões para a atualização.

O CloudFormation implantará atualizações de pilhas nas contas especificadas na primeira região, depois passará para a próxima e assim por diante, desde que as falhas de implantação de uma região não excedam uma tolerância a falhas especificada.

a. [Permissões autogerenciadas] Em Contas, Locais de implantação, escolha Implantar pilhas em contas. Cole os IDs das contas de destino usadas para criar o StackSet na caixa de texto, separando os números distintos com vírgulas.

[Permissões gerenciadas pelo serviço] Siga um destes procedimentos:

- Escolha Deploy to organizational units (OUs) (Implantar em unidades organizacionais (OUs)). Insira as UOs de destino que você usou para criar seu StackSet.
- Escolha Deploy to accounts (Implantar em contas). Cole os IDs das UOs ou das contas de destino usados para criar seu StackSet.

b. Em Especificar regiões, especifique a ordem na qual você deseja que o CloudFormation implemente suas atualizações.

c. Em Opções de implantação, faça o seguinte:

- Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
- Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.
- Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
- Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
- Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.

- Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.
- [Permissões gerenciadas pelo serviço] Para dependências do StackSet, adicione ARNs dependentes do StackSet, permanecendo com dez dependências no máximo. Para obter mais informações, consulte [Habilite ou desabilite implantações automáticas de StackSets no AWS Organizations](#).

d. Escolha Próximo para continuar.

17. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.
18. Quando estiver pronto para prosseguir, escolha Enviar.

O CloudFormation começa a aplicar as atualizações ao seu StackSet e exibe a guia Operações da página de detalhes do StackSet. É possível visualizar o progresso e o status das operações de atualização na guia Operations (Operações).

Atualizar seu StackSet (AWS CLI)

Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

1. Para atualizar um StackSet

Use o comando [update-stack-set](#) para fazer alterações no seu StackSet.

Nos exemplos a seguir, vamos atualizar o StackSet usando a opção `--parameters`. Especificamente, alteramos a frequência de entrega do snapshot padrão para a configuração do canal de entrega de `TwentyFour_Hours` para `Twelve_Hours`. Como continuaremos a usar o modelo atual, adicionaremos a opção `--use-previous-template`.

Defina o processamento simultâneo de contas e outras preferências de implantação usando a opção `--operation-preferences`. Esses exemplos usam configurações baseadas em contagem. Observe que `MaxConcurrentCount` não deve exceder `FailureToleranceCount + 1`. Para configurações baseadas em porcentagem, use `FailureTolerancePercentage` ou `MaxConcurrentPercentage` em vez disso.

[Permissões autogerenciadas] Para a opção `--accounts`, forneça os IDs das contas que você deseja que a atualização tenha como destino.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

[Permissões gerenciadas pelo serviço] Na opção `--deployment-targets`, forneça o ID da raiz da organização ou os IDs das unidades organizacionais (UOs) às quais você deseja que a atualização se aplique.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para obter mais informações, consulte [UpdateStackSet](#) na Referência de API do AWS CloudFormation.

2. Verifique se o StackSet foi atualizado com êxito executando o comando `describe-stack-set-operation` para mostrar o status e os resultados da operação de atualização. Em `--operation-id`, use o ID de operação que foi retornado pelo comando `update-stack-set`.

```
aws cloudformation describe-stack-set-operation \  
  --operation-id operation_ID
```

Adicionar pilhas ao CloudFormation StackSets

Ao criar um StackSet, você pode criar as pilhas para ele. O CloudFormation também permite que você adicione mais pilhas, para contas e regiões adicionais, a qualquer momento após o StackSet ser criado. É possível adicionar pilhas usando o console ou a AWS CLI do CloudFormation.

Tópicos

- [Adicionar pilhas a um StackSet \(console\)](#)
- [Adicionar pilhas a um StackSet \(AWS CLI\)](#)

Adicionar pilhas a um StackSet (console)

Para adicionar pilhas a um StackSet

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. No painel de navegação, escolha StackSets. Na página StackSets, selecione o StackSet que você criou.
4. Com o StackSet selecionado, escolha Adicionar pilhas ao StackSet no menu Ações.
5. Adicione as seguintes informações à página Definir opções de implantação:
 - a. Em Adicionar pilhas ao StackSet, escolha Implantar novas pilhas.
 - b. Depois, faça o seguinte de acordo com sua configuração de permissões do StackSet:
 - [Permissões autogerenciadas] Em Contas, Locais de implantação, escolha Implantar pilhas em contas. Cole os números da sua conta de destino na caixa de texto, separando-os com vírgulas.
 - [Permissões gerenciadas pelo serviço] Em Destinos de implantação, faça o seguinte:
 - Escolha Deploy to organization (Implantar na organização) para implantar em todas as contas em sua organização.
 - Escolha Deploy to organizational units (OUs) (Implantar em unidades organizacionais (OUs)) para implantar em todas as contas em OUs específicas. Escolha Add another OU (Adicionar outra OU) e cole o ID da OU de destino na caixa de texto. Repita para cada nova OU de destino. O CloudFormation também tem como destino as UOs secundárias dos destinos selecionados.

Note

Se você adicionar uma UO que o StackSet já tem como destino, o CloudFormation criará novas pilhas em todas as contas da UO que ainda não

tenham pilhas do StackSet (por exemplo, contas que foram adicionadas à UO depois que o StackSet foi criado e com implantações automáticas desabilitadas).

- c. Em Especificar regiões, especifique em quais Regiões da AWS implantar nas contas de destino especificadas na etapa anterior. Por padrão, o CloudFormation implantará pilhas nas contas especificadas na primeira região, depois passará para a próxima e assim por diante, desde que as falhas de implantação de uma região não excedam uma tolerância a falhas especificada.
 - d. Em Opções de implantação, faça o seguinte:
 - Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
 - Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.
 - Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
 - Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
 - Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.
 - Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.
 - e. Escolha Próximo.
6. Na página Specify Overrides (Especificar substituições), deixe os valores da propriedade conforme especificados. Você não substituirá nenhum valor de propriedade para as pilhas que criará. Escolha Próximo.
 7. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.
 8. Quando estiver pronto para prosseguir, escolha Enviar.

O CloudFormation começa a criar as pilhas. Visualize o progresso e o status da criação das pilhas em seu StackSet na página de detalhes do StackSet que é aberta ao escolher Enviar. Após a conclusão, as novas pilhas devem estar listadas na guia Instâncias da pilha.

Adicionar pilhas a um StackSet (AWS CLI)

Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

Para adicionar pilhas a um StackSet com permissões autogerenciadas

Use o comando de CLI de `create-stack-instances`. Na opção `--accounts`, forneça os IDs das contas em que você deseja criar pilhas.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --accounts account_id --regions eu-west-1 us-west-2
```

Para adicionar pilhas a um StackSet com permissões gerenciadas pelo serviço

Use o comando de CLI de `create-stack-instances`. Na opção `--deployment-targets`, forneça o ID da organização (raiz) ou os IDs das UOs em que você deseja criar pilhas. Por exemplo, comandos que têm como destino contas específicas, consulte [Criar um StackSet com permissões gerenciadas pelo serviço \(AWS CLI\)](#).

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-west-2
```

Note

Se você adicionar uma UO que o StackSet já tem como destino, o CloudFormation criará novas pilhas em todas as contas da UO que ainda não tenham pilhas do StackSet (por exemplo, contas que foram adicionadas à UO depois que o StackSet foi criado e com implantações automáticas desabilitadas).

Substituir valores de parâmetros nas pilhas do StackSet do CloudFormation

Em determinados casos, você pode querer que as pilhas de certas regiões ou contas tenham valores de propriedades diferentes dos especificados no próprio StackSet. Por exemplo, você pode especificar um valor diferente para um determinado parâmetro com base no uso de desenvolvimento ou de produção da conta. Nessas situações, o CloudFormation permite que você substitua os valores dos parâmetros das pilhas por conta e região. Você pode substituir os valores dos parâmetros do modelo ao criar as pilhas e também os valores dos parâmetros de pilhas já existentes. Você pode definir apenas os parâmetros que você substituiu anteriormente nas pilhas de volta para os valores especificados no StackSet.

As substituições de valores de parâmetros se aplicam às pilhas das contas e regiões selecionadas. Durante as atualizações do StackSet, todos os valores dos parâmetros substituídos de uma pilha não são atualizados, mas retêm o valor substituído.

Você só pode substituir valores de parâmetros especificados no StackSet; para adicionar ou excluir um parâmetro, você precisa atualizar o modelo do StackSet. Se você adicionar um parâmetro a um modelo de StackSet, para poder substituir o valor desse parâmetro em uma pilha, será necessário primeiro atualizar todas as pilhas com o novo parâmetro e valor especificado no StackSet. Quando todas as pilhas tiverem sido atualizadas com o novo parâmetro, você poderá substituir o valor do parâmetro nas pilhas individuais como desejado.

Para saber como substituir os valores dos parâmetros do StackSet ao criar pilhas, consulte [Adicionar pilhas a StackSets](#).

Tópicos

- [Substituir parâmetros em pilhas \(console\)](#)
- [Substituir parâmetros em pilhas \(AWS CLI\)](#)

Substituir parâmetros em pilhas (console)

Para substituir parâmetros de pilhas específicas

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

2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. No painel de navegação, escolha StackSets. Na página StackSets, selecione seu StackSet.
4. Com o StackSet selecionado, escolha Substituir parâmetros do StackSet no menu Ações.
5. Na página Definir opções de implantação, forneça as contas e as regiões das pilhas para as quais você pretende criar substituições.

Por padrão, o CloudFormation implantará pilhas nas contas especificadas na primeira região, depois passará para a próxima e assim por diante, contanto que as falhas de implantação de uma região não excedam uma tolerância a falhas especificada.

- a. [Permissões autogerenciadas] Em Locais de implantação, escolha Implantar pilhas em contas. Cole alguns ou todos os IDs de conta de destino usados para criar o StackSet.

[Permissões gerenciadas pelo serviço] Siga um destes procedimentos:

- Escolha Deploy to organizational units (OUs) (Implantar em unidades organizacionais (OUs)). Insira uma ou mais das UOs de destino que você usou para criar seu StackSet. Os valores de parâmetro substituídos só se aplicam às contas que estão atualmente nas OUs de destino e em suas OUs filhas. As contas adicionadas às UOs de destino e suas UOs secundárias no futuro usarão os valores padrão do StackSet e não os valores substituídos.
 - Escolha Deploy to accounts (Implantar em contas). Cole alguns ou todos os IDs de UOs ou de contas de destino usados para criar o StackSet.
- b. Em Especificar regiões, adicione uma ou mais regiões nas quais você implantou pilhas para esse StackSet.

Se você adicionar várias regiões, a ordem das regiões em Specify regions (Especificar regiões) determinará a ordem de implantação.

- c. Em Opções de implantação, faça o seguinte:
 - Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
 - Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.

- Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
 - Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
 - Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.
 - Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.
- d. Escolha Próximo.
6. Na página Especificar substituições, marque as caixas de seleção correspondentes aos parâmetros que serão substituídos e escolha Substituir valor do StackSet no menu Editar valor de substituição.
 7. Na página Substituir valores de parâmetros do StackSet, faça suas alterações e escolha Salvar alterações.

 Note

Para definir os parâmetros substituídos de volta para usarem o valor especificado no StackSet, verifique todos os parâmetros e escolha Definir como o valor do StackSet no menu Editar valor de substituição. Fazer isso removerá todos os valores substituídos quando você atualizar as pilhas.

8. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.
9. Quando estiver pronto para prosseguir, escolha Enviar.

O CloudFormation começa a atualizar as pilhas. Visualize o progresso e o status das pilhas na página de detalhes do StackSet que é exibida quando você escolhe Enviar.

Substituir parâmetros em pilhas (AWS CLI)

 Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

Para substituir parâmetros de pilhas específicas

1. Execute o comando [update-stack-instances](#) da AWS CLI e especifique a opção `--parameter-overrides`.

[Permissões autogerenciadas] Na opção `--accounts`, forneça os IDs das contas para as quais você deseja substituir valores de parâmetros nas pilhas.

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --accounts account_id --regions us-east-1
```

[Permissões gerenciadas pelo serviço] Na opção `--deployment-targets`, forneça o ID da raiz da organização, as IDs das UOs ou as IDs de contas do AWS Organizations nas quais você deseja substituir parâmetros em pilhas. Neste exemplo, substituímos os valores de parâmetros nas pilhas de todas as contas da UO com o ID *ou-rcuk-1x5j1lwo*.

Os valores de parâmetros substituídos só se aplicam às contas que estão atualmente na OU de destino e em suas OUs filhas. As contas adicionadas à UO de destino e suas UOs secundárias no futuro usarão os valores padrão do StackSet e não os valores substituídos.

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo \  
  --regions us-east-1
```

2. Verifique se os valores dos parâmetros foram substituídos com êxito nas pilhas executando o comando `describe-stack-set-operation` para mostrar o status e os resultados da operação de atualização. Em `--operation-id`, use o ID de operação que foi retornado pelo comando `update-stack-instances`.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Excluir pilhas do CloudFormation StackSets

É possível excluir pilhas de StackSets usando o console ou a AWS CLI do CloudFormation.

Tópicos

- [Excluir pilhas do seu StackSet \(console\)](#)
- [Excluir pilhas do StackSet \(AWS CLI\)](#)

Note

Excluir pilhas de uma unidade organizacional (UO) do nível superior removerá essa UO como um destino do StackSet.

Excluir pilhas do seu StackSet (console)

Para excluir pilhas

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. No painel de navegação, escolha StackSets. Na página StackSets, selecione o StackSet.
4. Com o StackSet selecionado, escolha Excluir pilhas do StackSet no menu Ações.
5. Na página Definir opções de implantação, primeiro escolha as contas e as regiões nas quais você deseja excluir as pilhas.
 - a. [Permissões autogerenciadas] Em Contas, escolha Implantar pilhas em contas ou Implantar pilhas em unidades organizacionais.

Se você escolher Implantar pilhas em contas, cole os números das contas de destino na caixa de texto Números das contas, separando os números com vírgulas.

Se você escolher Implantar pilhas em unidades organizacionais, cole a ID da OU de destino na caixa de texto Números da organização para que todas as contas da organização especificada sejam destino da implantação.

- b. [Permissões gerenciadas pelo serviço] Em Unidades organizacionais (UOs), especifique os IDs das UOs de destino.

 Important

O CloudFormation excluirá as pilhas das UOs de destino especificadas e das OUs subordinadas a elas.

Em Tipo de filtro da conta, você pode refinar de quais contas as pilhas serão excluídas escolhendo uma das opções a seguir e fornecendo os números das contas.

- Nenhuma (padrão): excluir as pilhas de todas as contas da UOs especificadas.
- Interseção: excluir as pilhas apenas de determinadas contas individuais das UOs selecionadas.
- Diferença: excluir as pilhas de todas as contas das UOs selecionadas, exceto determinadas contas.
- União: excluir as pilhas das UOs especificadas e também de contas individuais adicionais.

- c. Em Especificar regiões, escolha as regiões das quais você deseja excluir as pilhas das contas de destino.

6. Em Opções de implantação, faça o seguinte:

- Em Máximo de contas simultâneas, especifique quantas contas são processadas simultaneamente.
- Em Tolerância a falhas, especifique o número máximo de falhas de conta permitidas por região. A operação será interrompida e não prosseguirá para outras regiões quando esse limite for atingido.
- Em Reter pilhas, habilite esta opção para salvar as pilhas e os recursos a elas associados ao removê-las do StackSet. Os recursos permanecem em seu estado atual, mas não fazem mais parte do StackSet.
- Em Simultaneidade de regiões, escolha o modo de processar as regiões: Sequencial (uma região de cada vez) ou Paralelo (várias regiões simultaneamente).
- Em Modo de simultaneidade, escolha como a simultaneidade se comporta durante a execução de uma operação.
 - Tolerância a falhas estrita: reduz o nível de simultaneidade da conta quando ocorrem falhas, permanecendo dentro da faixa de Tolerância a falhas +1.

- Tolerância a falhas flexível: mantém o nível de simultaneidade especificado (o valor de Máximo de contas simultâneas) a despeito de falhas.

7. Escolha Próximo.
8. Na página Analisar, examine suas escolhas. Para fazer alterações, escolha Editar na seção relacionada.
9. Quando estiver tudo pronto para remover as pilhas do StackSet, escolha Enviar.

Após a exclusão de pilhas ser concluída, é possível verificar se as pilhas foram excluídas do StackSet na página de detalhes do StackSet, na guia Instâncias de pilha.

Excluir pilhas do StackSet (AWS CLI)

Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

Use o comando `delete-stack-instances` com o nome do StackSet.

Nestes exemplos, usamos a opção `--no-retain-stacks` porque não vamos reter nenhuma pilha. Use `--retain-stacks` em vez de `--no-retain-stacks` se quiser manter as pilhas e seus recursos.

Em `--regions`, especifique as Regiões da AWS das quais você deseja excluir pilhas, por exemplo, `us-west-2` e `us-east-1`.

Defina o processamento simultâneo de contas e outras preferências usando a opção `--operation-preferences`. Esses exemplos usam configurações baseadas em contagem. Observe que `MaxConcurrentCount` não deve exceder `FailureToleranceCount + 1`. Para configurações baseadas em porcentagem, use `FailureTolerancePercentage` ou `MaxConcurrentPercentage` em vez disso.

Para excluir pilhas (permissões autogerenciadas)

Na opção `--accounts`, especifique os IDs das contas das quais as pilhas serão excluídas.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
--accounts account_ID_1 account_ID_2 \
```

```
--regions us-west-2 us-east-1 \  
--no-retain-stacks \  
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para excluir pilhas (permissões gerenciadas pelo serviço)

Em `--deployment-targets`, especifique o ID raiz da organização ou os IDs das unidades organizacionais (UOs) das quais as pilhas serão excluídas.

Important

O CloudFormation excluirá as pilhas das UOs de destino especificadas e das OUs subordinadas a elas.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=ou-rcuk-1x5jlwo,ou-rcuk-slrl5lh0a \  
--regions us-west-2 us-east-1 \  
--no-retain-stacks \  
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para obter mais informações, consulte [DeleteStackInstances](#) na AWS CloudFormation API Reference.

Opcionalmente, após a exclusão das pilhas ser concluída, verifique se as pilhas foram excluídas do StackSet executando o comando `describe-stack-set-operation` para mostrar o status e os resultados da operação de exclusão de pilhas. Em `--operation-id`, use o ID de operação que foi retornado pelo comando `delete-stack-instances`.

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \  
--operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```

Excluir CloudFormation StackSets

Para excluir um StackSet, você deve excluir primeiro todas as pilhas do StackSet. Para obter informações sobre como excluir todas as pilhas, consulte [Excluir pilhas de StackSets](#).

Tópicos

- [Excluir um StackSet \(console\)](#)

- [Excluir um StackSet \(AWS CLI\)](#)
- [Excluir funções de serviço \(opcional\)](#)

Excluir um StackSet (console)

Para excluir um StackSet

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que você criou o StackSet.
3. Na página StackSets, selecione o StackSet.
4. Com o StackSet selecionado, escolha Excluir StackSet no menu Ações.
5. Quando for solicitado a confirmar que você deseja excluir o StackSet, escolha a opção Excluir StackSet.

Excluir um StackSet (AWS CLI)

Note

Ao atuar como um administrador delegado, você deve incluir `--call-as DELEGATED_ADMIN` no comando.

Para excluir um StackSet

1. Use o seguinte comando `delete-stack-set`. Quando for solicitado, digite **y** e, em seguida, pressione Enter para confirmar.

```
aws cloudformation delete-stack-set --stack-set-name my-stackset
```

2. Verifique se o StackSet foi excluído executando o comando `list-stack-sets`. Os resultados do comando `list-stack-sets` devem mostrar sua pilha com status DELETED.

```
aws cloudformation list-stack-sets
```

Excluir funções de serviço (opcional)

Se os perfis de serviço do IAM exigidos pelo CloudFormation para realizar operações de StackSets não forem mais necessários, recomendamos excluí-los.

Para StackSets autogerenciados, os perfis que você criou. Para obter mais informações sobre esses perfis, consulte [Conceder permissões autogerenciadas](#).

Para StackSets gerenciados pelo serviço, os perfis que foram criados automaticamente para StackSets têm o sufixo CloudFormationStackSets0rgAdmin na conta gerencial da organização e CloudFormationStackSets0rgMember em todas as contas de destino. Para obter mais informações, consulte [Perfis vinculados ao serviço](#).

Para excluir um perfil de serviço (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Perfis e marque a caixa de seleção ao lado da função que deseja excluir.
3. No menu Ações da função, na parte superior da página, escolha Excluir função.
4. Na caixa de diálogo de confirmação, escolha Yes, Delete. Se você tiver certeza, prossiga com a exclusão, mesmo se os dados do último acesso ao serviço ainda estiverem sendo carregados.

Para excluir um perfil de serviço (AWS CLI)

- Use o seguinte comando delete-role. Quando a confirmação for solicitada, digite **y** e, em seguida, pressione Enter.

```
aws iam delete-role --role-name role name
```

Para obter mais informações sobre como excluir perfis, consulte [Excluir funções ou perfis de instância](#) no Guia do usuário do IAM.

Evitar implantações de StackSets com falha usando portas de conta de destino

Uma porta de conta é um recurso opcional que ajuda você a verificar se uma conta de destino atende a determinados requisitos antes que o CloudFormation comece a executar operações de StackSets nessa conta. Essa verificação é realizada por meio de uma função do AWS Lambda que atua como uma verificação de pré-requisito.

Um exemplo comum de um portão de conta é a verificação de que não há alarmes do CloudWatch ativos ou não resolvidos na contas de destino. O CloudFormation invoca a função do Lambda toda vez que você inicia as operações de pilha na conta de destino, e só continua se a função retorna um código SUCCEEDED. Se a função do Lambda retornar um status FAILED, o CloudFormation não continuará a operação solicitada. Se você não tiver uma função do Lambda com uma porta de conta configurada, o CloudFormation vai ignorar a verificação e continuar com a operação.

Se a sua conta de destino falhar em uma verificação de portão de conta, a operação com falha entrará na contabilização do número ou porcentagem especificado de falhas toleradas em pilhas. Para obter mais informações sobre tolerância a falhas, consulte [Opções de operação de StackSets](#).

O portão de conta está disponível somente para operações do StackSets. Esse recurso não está disponível para outras operações do CloudFormation fora do StackSets.

Requisitos

Os seguintes requisitos devem ser atendidos para a liberação da conta:

- Sua função do Lambda deve ser nomeada `AWSCloudFormationStackSetAccountGate` para usar esse recurso.
- A `AWSCloudFormationStackSetExecutionRole` precisa de permissões para invocar sua função do Lambda. Sem essas permissões, o CloudFormation ignora a verificação da liberação da conta e continua com as operações de pilha.
- A permissão `InvokeFunction` do Lambda deve ser adicionada às contas de destino para que o portão de conta funcione. A política de confiança da conta de destino deve ter uma relação de confiança com a conta do administrador. Veja a seguir um exemplo de instrução de política que concede as permissões de `InvokeFunction` do Lambda.

JSON

```
{
```

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

Modelos do CloudFormation para a criação de funções do Lambda

Use os modelos de exemplo a seguir para criar funções

AWS CloudFormation StackSet Account Gate do Lambda. Para criar uma nova pilha usando um desses modelos, consulte [Criar uma pilha no console do CloudFormation](#).

Local do modelo	Descrição
https://s3.amazonaws.com/cloudformation-sta-ckset-sample-templates-us-east-1/AccountGateSucceeded.yml	Cria uma pilha que implementa uma função Lambda de portão de conta que retornará o status SUCCEEDED .
https://s3.amazonaws.com/cloudformation-sta-ckset-sample-templates-us-east-1/AccountGateFailed.yml	Cria uma pilha que implementa uma função Lambda de portão de conta que retornará o status FAILED.

Escolher o modo de simultaneidade para o CloudFormation StackSets

O Modo de simultaneidade é um parâmetro para [StackSetOperationPreferences](#) que permite que você escolha como o nível de simultaneidade se comporta durante as operações de StackSets. Você pode escolher entre os seguintes modos:

- Alta tolerância a falhas: essa opção reduz dinamicamente o nível de simultaneidade para garantir que o número de contas com falha nunca exceda o valor de Tolerância a falhas +1. A

simultaneidade real inicial é definida como o menor valor do Máximo de contas simultâneas ou o valor de Tolerância a falhas +1. A simultaneidade real é então reduzida proporcionalmente pelo número de falhas. Esse é o comportamento padrão.

- Tolerância moderada a falhas: essa opção desacopla a Tolerância a falhas da simultaneidade real. Isso permite que as operações de StackSets sejam executadas no nível de simultaneidade definido pelo valor de Máximo de contas simultâneas, independentemente do número de falhas.

A Tolerância a falhas estrita reduz a velocidade de implantação à medida que ocorrem falhas na operação do StackSet porque a simultaneidade diminui para cada falha. A Tolerância moderada a falhas prioriza a velocidade de implantação e, ao mesmo tempo, aproveita os recursos de segurança do CloudFormation. Isso permite que você analise e resolva falhas de operação do StackSet relacionadas a problemas comuns, como as relacionadas a recursos, cotas de serviço e permissões existentes.

Para obter mais informações sobre falhas na operação de pilha do StackSets, consulte [Motivos comuns de falha na operação da pilha](#).

Para obter mais informações sobre o Máximo de contas simultâneas e a Tolerância a falhas, consulte [Opções de operação de StackSets](#).

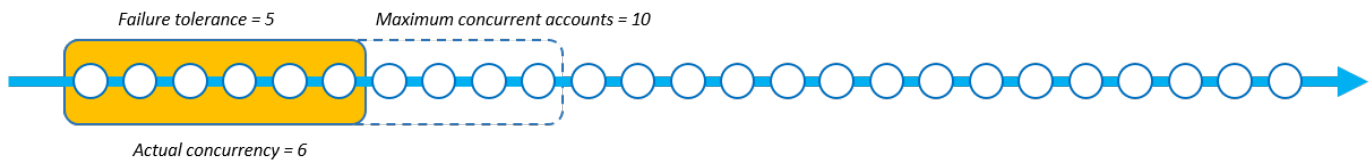
Como cada Modo de Simultaneidade funciona

As imagens abaixo fornecem uma representação visual de como cada Modo de simultaneidade funciona durante uma operação de StackSet. A sequência de nós representa uma implantação em uma única Região da AWS e cada nó é uma Conta da AWS de destino.

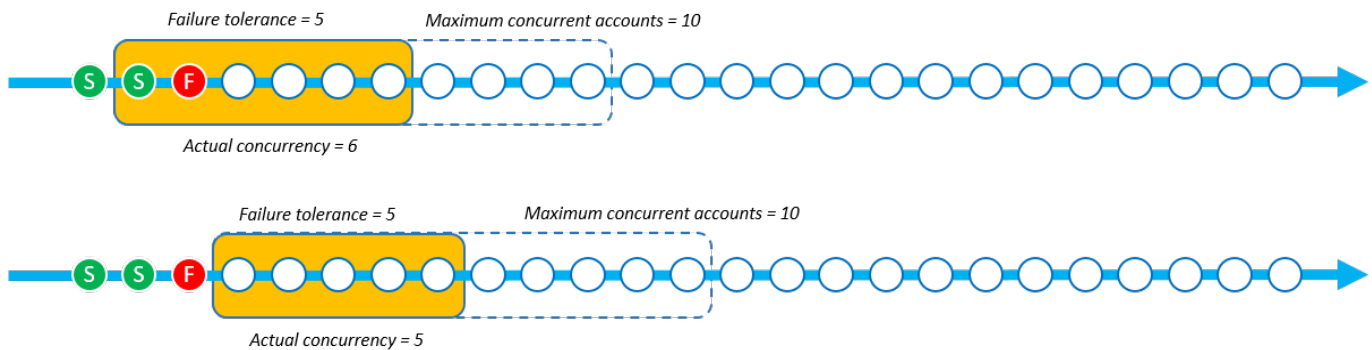
Alta tolerância a falhas

Quando uma operação de StackSet usando a Tolerância a falhas estrita tem o valor de Tolerância a falhas definido como 5 e o valor de Máximo de contas simultâneas definido como 10, a simultaneidade real é 6. A simultaneidade real é 6 porque o valor de Tolerância a falhas de 5 +1 é menor do que o valor do Máximo de contas simultâneas.

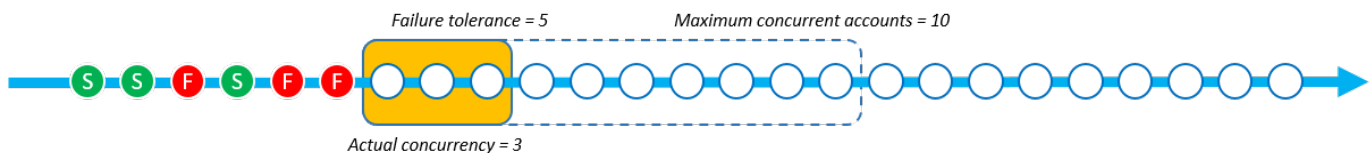
A imagem abaixo mostra o impacto que o valor da Tolerância a falhas tem no valor de Máximo de contas simultâneas, e o impacto que ambos têm na simultaneidade real da operação do StackSet:



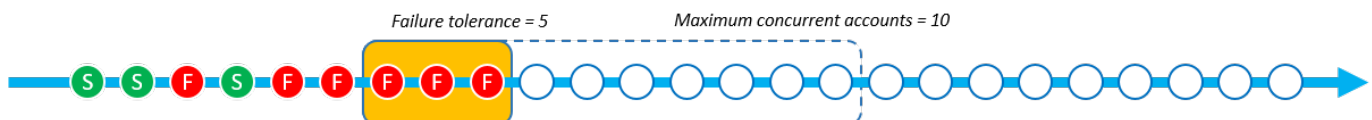
Quando a implantação começa e há instâncias de pilha com falha, a simultaneidade real é reduzida para fornecer uma experiência de implantação segura. A simultaneidade real é reduzida de 6 para 5 quando o StackSets falha ao implantar uma instância de pilha.



O modo Alta tolerância a falhas reduz a simultaneidade real proporcionalmente ao número de instâncias de pilha com falha. No exemplo a seguir, a simultaneidade real é reduzida de 5 para 3 quando o StackSets falha ao implantar mais duas instâncias de pilha, elevando o total de instâncias de pilha com falha para três.



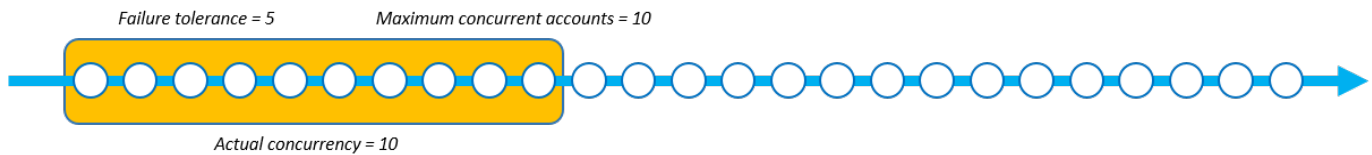
O StackSets falha na operação de StackSets quando o número de instâncias de pilha com falha é igual ao valor definido de Tolerância a falhas +1. No exemplo a seguir, o StackSets falha na operação quando há seis instâncias de pilha com falha e o valor de Tolerância a falhas é 5.



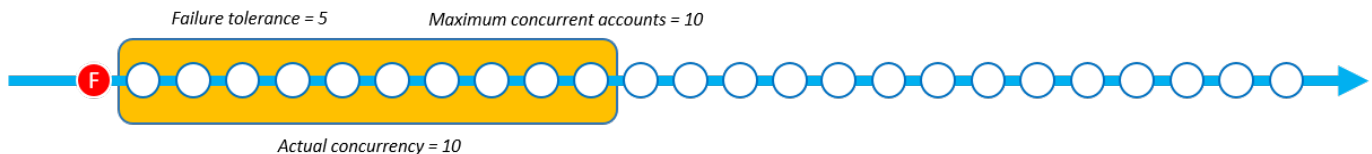
Neste exemplo, o CloudFormation implantou nove instâncias de pilha (três com êxito e seis com falha) antes de interromper a operação do StackSet.

Tolerância moderada a falhas

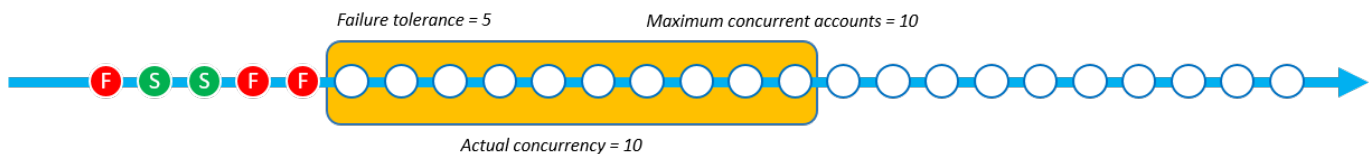
Quando uma operação de StackSet usando a Tolerância moderada a falhas tem o valor de Tolerância a falhas definido como 5 e o valor de Máximo de contas simultâneas definido como 10, a simultaneidade real é 10.



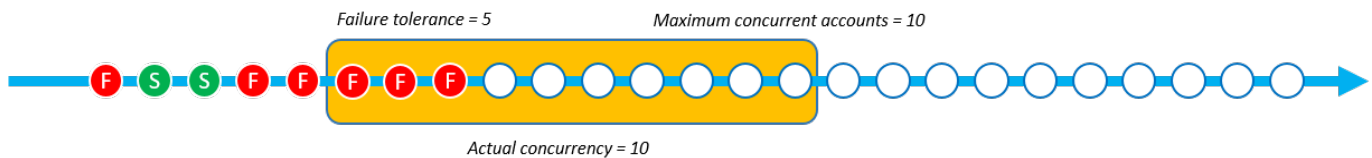
Quando a implantação começa e há instâncias de pilha com falha, a simultaneidade real não muda. No exemplo a seguir, uma operação de pilha falhou, mas a simultaneidade real permanece em 10.



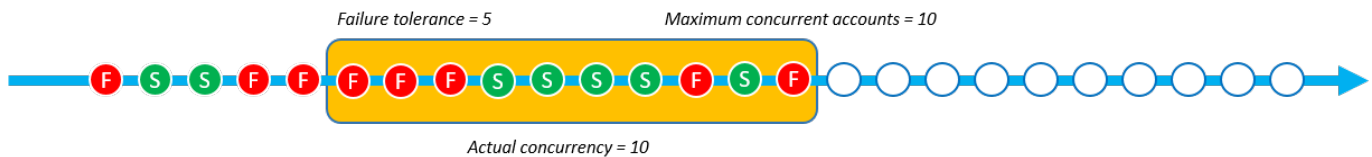
A simultaneidade real permanece em 10, mesmo após mais duas falhas na instância de pilha.



O StackSets falha na operação de StackSets quando o número de instâncias de pilha com falha excede o valor de Tolerância a falhas. No exemplo a seguir, o StackSets falha na operação quando há seis instâncias de pilha com falha e a contagem de Tolerância a falhas é 5. No entanto, a operação não terminará até que as operações restantes na fila de simultaneidade terminem.



O StackSets continua a implantar instâncias de pilha que já estão na fila de simultaneidade. Isso significa que o número de instâncias de pilha com falha pode ser maior do que a Tolerância a falhas. No exemplo a seguir, há oito instâncias de pilha com falha porque a fila de simultaneidade ainda tinha sete operações a serem executadas, embora a operação do StackSet tivesse atingido a Tolerância a falhas de cinco.



Neste exemplo, o StackSets implantou 15 instâncias de pilha (sete com êxito e oito com falha) antes de interromper a operação da pilha.

Escolha entre Alta tolerância a falhas e Tolerância moderada a falhas com base na velocidade de implantação

Escolher entre Tolerância a falhas estrita e Tolerância moderada a falhas depende da velocidade de preferência da implantação do StackSet e do número permitido de falhas de implantação.

As tabelas a seguir mostram como cada modo de simultaneidade lida com uma operação de StackSet que falha ao tentar implantar um total de mil instâncias de pilha. Em cada cenário, o valor de Tolerância a falhas é definido como 100 instâncias de pilha e o valor Máximo de contas simultâneas é definido como 250 instâncias de pilha.

Embora o StackSets realmente coloque as contas em fila como uma janela deslizante (consulte [Como cada Modo de Simultaneidade funciona](#)), este exemplo mostra a operação em lotes para demonstrar a velocidade de cada modo.

Alta tolerância a falhas

Este exemplo usando o modo de Alta tolerância a falhas reduz a simultaneidade real em relação ao número de falhas que ocorrem em cada lote anterior. Cada lote tem vinte instâncias com falha, o que reduz a simultaneidade real do lote seguinte em vinte até que a operação do StackSet atinja o valor de Tolerância a falhas de cem.

Na tabela a seguir, a simultaneidade real inicial do primeiro lote é de 101 instâncias de pilha. A simultaneidade real é 101 porque é o valor mais baixo do Máximo de contas simultâneas (250) e da Tolerância a falhas (100) + 1. Cada lote contém 20 implantações de instâncias de pilha com falha, o que reduz a simultaneidade real de cada lote seguinte em 20 instâncias de pilha.

Alta tolerância a falhas	Lote 1	Lote 2	Lote 3	Lote 4	Lote 5	Lote 6
Contagem real de simultaneidade	101	81	61	41	21	-
Contagem de instâncias com falha	20	20	20	20	20	-
Contagem de instâncias de pilha com êxito	81	61	41	21	1	-

A operação usando a Tolerância a falhas estrita concluiu 305 implantações de instâncias de pilha em cinco lotes no momento em que a operação do StackSet atingiu a Tolerância a falhas de cem instâncias de pilha. A operação do StackSet implanta com êxito 205 instâncias de pilha antes de falhar.

Tolerância moderada a falhas

Este exemplo usando o modo de Tolerância moderada a falhas mantém a mesma contagem real de simultaneidade definida pelo valor Máximo de contas simultâneas de 250 instâncias de pilha, independentemente do número de instâncias com falha. As operações do StackSet mantêm a mesma simultaneidade real até atingir o valor de Tolerância a falhas de cem instâncias.

Na tabela a seguir, a simultaneidade real inicial do primeiro lote é de 250 instâncias de pilha. A simultaneidade real é 250 porque o valor do Máximo de contas simultâneas está definido como 250 e o modo Tolerância moderada a falhas permite que o StackSets use esse valor como a simultaneidade real, independentemente do número de falhas. Embora haja 50 falhas em cada um dos lotes neste exemplo, a simultaneidade real permanece inalterada.

Tolerância moderada a falhas	Lote 1	Lote 2	Lote 3	Lote 4	Lote 5	Lote 6
Contagem real de simultaneidade	250	250	-	-	-	-
Contagem de instâncias com falha	50	50	-	-	-	-
Contagem de instâncias de pilha com êxito	200	200	-	-	-	-

Usando o mesmo valor de Máximo de contas simultâneas e Tolerância a falhas, a operação usando o modo Tolerância moderada a falhas concluiu 500 implantações de instâncias de pilha em dois lotes. A operação do StackSet implanta com êxito 400 instâncias de pilha antes de falhar.

Escolher o modo de simultaneidade (console)

Ao criar ou atualizar um StackSet, na página Definir opções de implantação, em Modo de simultaneidade, escolha Tolerância a falhas estrita ou Tolerância moderada a falhas.

Escolher o modo de simultaneidade (AWS CLI)

Você pode usar o parâmetro `ConcurrencyMode` com os seguintes comandos do StackSets:

- [create-stack-instances](#)
- [delete-stack-instances](#)
- [detect-stack-set-drift](#)
- [import-stacks-to-stack-set](#)

- [update-stack-instances](#)
- [update-stack-set](#)

Esses comandos têm um parâmetro existente chamado `--operation-preferences` que pode usar a configuração `ConcurrencyMode`. O `ConcurrencyMode` pode ser definido com um dos seguintes valores:

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

O exemplo a seguir cria uma instância de pilha usando o `ConcurrencyMode` `STRICT_FAILURE_TOLERANCE` com `FailureToleranceCount` definida como 10 e `MaxConcurrentCount` definida como 5.

```
aws cloudformation create-stack-instances \  
  --stack-set-name example-stackset \  
  --accounts 123456789012 \  
  --regions eu-west-1 \  
  --operation-preferences  
ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

Note

Para obter os procedimentos detalhados de criação e atualização de um `StackSet`, consulte os seguintes tópicos:

- [Criar StackSets \(permissões autogerenciadas\)](#)
- [Criar um conjunto de pilhas \(permissões gerenciadas pelo serviço\)](#)
- [Atualizar StackSets](#)

Detectar desvios em StackSets do CloudFormation

Mesmo que você gerencie as pilhas e seus respectivos recursos usando o CloudFormation, os usuários poderão alterar esses recursos fora do CloudFormation. Os usuários podem editar recursos diretamente usando o serviço subjacente que os criou. Ao realizar a detecção de desvios em um

StackSet, você pode determinar se existem instâncias da pilha pertencentes ao StackSet que sejam diferentes ou apresentem desvios com relação à configuração esperada.

Tópicos

- [Como o CloudFormation executa a detecção de desvios em um StackSet](#)
- [Detectar desvios em um StackSet \(console\)](#)
- [Detectar desvios em um StackSet \(AWS CLI\)](#)
- [Como interromper a detecção de desvios em um StackSet](#)

Como o CloudFormation executa a detecção de desvios em um StackSet

Quando o CloudFormation executa a detecção de desvios em um StackSet, ele a executa na pilha associada a cada instância de pilha no StackSet. Para isso, o CloudFormation compara o estado atual de cada recurso na pilha com o estado esperado desse recurso, conforme definido no modelo da pilha e em todos os parâmetros de entrada especificados. Se o estado atual de um recurso for diferente do estado esperado, esse recurso será considerado como contendo desvios. Se um ou mais recursos em uma pilha apresentarem desvios, a própria pilha será considerada como contendo desvios e as instâncias da pilha às quais a pilha está associada também serão consideradas como contendo desvios. Se uma ou mais instâncias da pilha em um StackSet apresentarem desvios, o próprio StackSet será considerado como contendo desvios.

A detecção de desvios identifica alterações não gerenciadas; ou seja, as alterações feitas nas pilhas fora do CloudFormation. As alterações feitas diretamente em uma pilha por meio do CloudFormation, em vez de no nível do StackSet, não são consideradas como desvios. Por exemplo, suponha que você tenha uma pilha que está associada a uma instância de pilha de um StackSet. Se você usar o CloudFormation para atualizar essa pilha a fim de usar um modelo diferente, isso não será considerado um desvio, mesmo que essa pilha tenha um modelo diferente das outras pilhas pertencentes ao StackSet. Isso ocorre porque a pilha ainda corresponderá ao modelo esperado e à configuração do parâmetro no CloudFormation.

Para obter informações detalhadas sobre como o CloudFormation realiza a detecção de desvios em uma pilha, consulte [Detectar alterações de configuração não gerenciadas em pilhas e recursos com detecção de desvios](#).

Como o CloudFormation executa a detecção de desvios individualmente em cada pilha, ele considera todos os valores de parâmetros substituídos ao determinar se uma pilha está com desvios.

Para saber mais sobre como substituir parâmetros de modelos em instâncias de pilhas, consulte [Substituir valores de parâmetros nas pilhas do StackSet do CloudFormation](#).

Caso execute a detecção de desvios [diretamente em uma pilha](#) que esteja associada a uma instância de pilha, esses resultados de desvios não estarão disponíveis na página do console StackSets (Conjunto de pilhas).

Detectar desvios em um StackSet (console)

Para detectar desvios em um StackSet

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página StackSets, selecione o StackSet em que deseja executar a detecção de desvios.
3. No menu Actions (Ações), selecione Detect drifts (Detectar desvios).

O CloudFormation exibe uma barra de informações que declara que a detecção de desvios foi iniciada para o StackSet selecionado.

4. Opcional: para monitorar o progresso da operação de detecção de desvios:
 - a. Selecione o nome do StackSet para exibir a página Detalhes do StackSet.
 - b. Selecione a guia Operations (Operações), selecione a operação de detecção de desvios e selecione View drift details (Visualizar detalhes do desvio).

O CloudFormation exibe a caixa de diálogo Operation details (Detalhes da operação).

5. Aguarde até que o CloudFormation conclua a operação de detecção de desvios. Quando a operação de detecção de desvios for concluída, o CloudFormation atualizará o Status do desvio e o Tempo da última verificação de desvios do seu StackSet. Esses campos estão listados na guia Visão geral da página Detalhes do StackSet do StackSet selecionado.

A operação de detecção de desvios pode levar algum tempo, dependendo do número de instâncias de pilha e do número de recursos incluídos no StackSet. Você só pode executar uma operação de detecção de desvios por vez em um determinado StackSet. O CloudFormation continua a operação de detecção de desvios mesmo depois de você descartar a barra de informações.

6. Para analisar os resultados de detecção de desvios para as instâncias de pilha em um StackSet, selecione a guia Instâncias de pilha.

A coluna Stack name (Nome da pilha) lista o nome da pilha associada a cada instância de pilha e a coluna Drift status (Status de desvio) lista o status do desvio da pilha. Considera-se que uma pilha foi desviada se um ou mais de seus recursos foram desviados.

7. Para revisar os resultados da detecção de desvios da pilha associada a uma instância de pilha específica:
 - a. Escolha a guia Operações.
 - b. Selecione a operação de desvio para a qual você deseja visualizar os resultados da detecção de desvio. Um painel dividido exibirá o status da instância da pilha e o motivo da operação selecionada. Para uma operação de desvio, a coluna de motivo do status mostra o status do desvio de uma instância de pilha.
 - c. Escolha a instância de pilha para a qual você deseja visualizar os detalhes do desvio e escolha Exibir desvios de recursos. Na tabela Status de desvios de recursos, na página Desvios de recursos, cada recurso de pilha é listado com seu status de desvio dele e a última vez em que a detecção de desvios foi iniciada no recurso. O ID lógico e o ID físico de cada recurso são exibidos para ajudar você a identificá-los.
8. É possível classificar os recursos com base em seus status de desvio usando a coluna Drift status (Status de desvio).

Para visualizar os detalhes em um recurso modificado:

- Com o recurso selecionado, escolha Visualizar detalhes de desvio.

O CloudFormation exibirá a página de detalhes de desvios desse recurso em particular. Esta página lista as diferenças do recurso. Ela também lista os valores de propriedade esperados e atuais do recurso.

 Note

Se a pilha pertencer a uma região e conta diferentes daquela em que você está conectado atualmente, o botão Detectar desvio será desabilitado e você não poderá ver os detalhes.

Detectar desvios em um StackSet (AWS CLI)

Para detectar desvios em uma pilha inteira usando a AWS CLI, use o seguinte procedimento:

Para detectar desvios em um StackSet

1. Use o comando [detect-stack-set-drift](#) para detectar desvios em todo o StackSet e nas instâncias de pilha associadas.

O exemplo a seguir inicia a detecção de desvios no StackSet `stack-set-drift-example`.

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

Resultado:

```
{  
  "OperationId": "c36e44aa-3a83-411a-b503-cb611example"  
}
```

2. Como as operações de detecção de desvios em um StackSet podem demorar, use o comando [describe-stack-set-operation](#) para monitorar o status da operação de desvios. Esse comando usa o ID da operação do StackSet retornado pelo comando `detect-stack-set-drift`.

Os exemplos a seguir usam o ID da operação do exemplo anterior para retornar informações sobre a operação de detecção de desvios do StackSet. Neste exemplo, a operação ainda está em andamento. Das sete instâncias de pilha associadas a esse StackSet, uma instância de pilha já foi detectada como com desvio, duas instâncias estão sincronizadas e a detecção de desvios para as outras quatro instâncias de pilha ainda está em andamento. Como uma instância está com desvio, o status de desvio do próprio StackSet está agora como `DRIFTED`.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Resultado:

```
{  
  "StackSetOperation": {  
    "Status": "RUNNING",  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "OperationPreferences": {  
      "RegionOrder": []  
    }  
  }  
}
```

```

    },
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "StackSetDriftDetectionDetails": {
        "DriftedStackInstancesCount": 1,
        "TotalStackInstancesCount": 7,
        "LastDriftCheckTimestamp": "2019-12-04T20:34:28.543Z",
        "InSyncStackInstancesCount": 2,
        "InProgressStackInstancesCount": 4,
        "DriftStatus": "DRIFTED",
        "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
}
}

```

Ao executar o mesmo comando posteriormente, este exemplo mostra as informações retornadas após a operação de detecção de desvios ter sido concluída. Duas das sete instâncias de pilha associadas a esse StackSet estão com desvio, renderizando o status de desvio do próprio StackSet como DRIFTED.

```

aws cloudformation describe-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id c36e44aa-3a83-411a-b503-cb611example

```

Resultado:

```

{
  "StackSetOperation": {
    "Status": "SUCCEEDED",
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "OperationPreferences": {
      "RegionOrder": []
    }
  },
  "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
  "EndTimestamp": "2019-12-04T20:37:32.829Z",
  "StackSetDriftDetectionDetails": {
    "DriftedStackInstancesCount": 2,

```

```

        "TotalStackInstancesCount": 7,
        "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
        "InSyncStackInstancesCount": 5,
        "InProgressStackInstancesCount": 0,
        "DriftStatus": "DRIFTED",
        "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
}
}

```

3. Quando a operação de detecção de desvios do StackSet estiver concluída, use os comandos `describe-stack-set`, `list-stack-instances`, `describe-stack-instance` e `list-stack-instance-resource-drifts` para analisar os resultados.

O comando [describe-stack-set](#) inclui as mesmas informações detalhadas de desvios retornadas pelo comando `describe-stack-set-operation`.

```

aws cloudformation describe-stack-set \
    --stack-set-name stack-set-drift-example

```

Resultado:

```

{
  "StackSet": {
    "Status": "ACTIVE",
    "Description": "Demonstration of drift detection on stack sets.",
    "Parameters": [],
    "Tags": [
      {
        "Value": "Drift detection",
        "Key": "Feature"
      }
    ],
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "Capabilities": [],
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",

```

```

    "StackSetDriftDetectionDetails": {
      "DriftedStackInstancesCount": 2,
      "TotalStackInstancesCount": 7,
      "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
      "InProgressStackInstancesCount": 0,
      "DriftStatus": "DRIFTED",
      "DriftDetectionStatus": "COMPLETED",
      "InSyncStackInstancesCount": 5,
      "FailedStackInstancesCount": 0
    },
    "StackSetARN": "arn:aws:cloudformation:us-east-1:123456789012:stackset/stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",
    "TemplateBody": [details omitted],
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22ebexample",
    "StackSetName": "stack-set-drift-example"
  }
}

```

Você pode usar o comando [list-stack-instances](#) para retornar informações resumidas sobre as instâncias de pilha associadas a um StackSet, incluindo o status de desvio de cada instância de pilha.

Neste exemplo, executar `list-stack-instances` no StackSet de exemplo com o filtro de status de desvio definido como `DRIFTED` permite identificar quais duas instâncias de pilha têm um status de desvio de `DRIFTED`.

```

aws cloudformation list-stack-instances \
  --stack-set-name stack-set-drift-example \
  --filters Name=DRIFT_STATUS,Values=DRIFTED

```

Resultado:

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:eu-west-1:123456789012:stack/StackSet-stack-set-drift-example-b0fb6083-60c0-4e39-af15-2f071e0db90c/0e4f0940-16d4-11ea-93d8-0641cexample",
      "Status": "CURRENT",
      "Account": "012345678910",
      "Region": "eu-west-1",

```

```

        "LastDriftCheckTimestamp": "2019-12-04T20:37:32.687Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
        "Status": "CURRENT",
        "Account": "123456789012",
        "Region": "us-east-1",
        "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    [additional stack instances omitted]
]
}

```

O comando [describe-stack-instance](#) também retorna essas informações, mas para uma única instância da pilha, como no exemplo abaixo.

```

aws cloudformation describe-stack-instance \
  --stack-set-name stack-set-drift-example \
  --stack-instance-account 012345678910 --stack-instance-region us-east-1

```

Resultado:

```

{
  "StackInstance": {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
    "Status": "CURRENT",
    "Account": "123456789012",
    "Region": "us-east-1",

```

```

    "ParameterOverrides": [],
    "DriftStatus": "DRIFTED",
    "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
    "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}

```

4. Após identificar as instâncias de pilha em que ocorreram desvios, você pode usar as informações sobre as instâncias de pilhas retornadas pelos comandos `list-stack-instances` ou `describe-stack-instance` para executar o comando [list-stack-instance-resource-drifts](#). Este comando retorna as informações detalhadas sobre quais recursos na pilha apresentam desvios para uma operação de desvio em particular.

O exemplo a seguir usa o parâmetro `--stack-instance-resource-drift-statuses` para solicitar informações de desvio de pilha para os recursos que foram modificados ou excluídos no exemplo de operação de desvio anterior. A solicitação retorna informações sobre o único recurso que foi modificado, incluindo detalhes sobre duas de suas propriedades e seus valores alterados. Nenhum recurso foi excluído.

```

aws cloudformation list-stack-instance-resource-drifts \
  --stack-set-name my-stack-set-with-resource-drift \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-east-1 \
  --operation-id c36e44aa-3a83-411a-b503-cb611example \
  --stack-instance-resource-drift-statuses MODIFIED DELETED

```

Resultado:

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-
stack-set-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ResourceType": "AWS::SQS::Queue",
      "Timestamp": "2018-03-26T17:23:34.489Z",
      "PhysicalResourceId": "https://sqs.us-
east-1.amazonaws.com/123456789012/my-stack-with-resource-drift-Queue-494PBHC076H4",
      "StackResourceDriftStatus": "MODIFIED",
      "PropertyDifferences": [

```

```
        {
            "PropertyPath": "/DelaySeconds",
            "ActualValue": "120",
            "ExpectedValue": "20",
            "DifferenceType": "NOT_EQUAL"
        },
        {
            "PropertyPath": "/RedrivePolicy/maxReceiveCount",
            "ActualValue": "12",
            "ExpectedValue": "10",
            "DifferenceType": "NOT_EQUAL"
        }
    ],
    "LogicalResourceId": "Queue"
}
]
```

Como interromper a detecção de desvios em um StackSet

Como a detecção de desvios em um StackSet pode ser uma operação de longa execução, é possível que existam instâncias quando decidir interromper uma operação de detecção de desvios em andamento em um StackSet.

Para interromper a detecção de desvios em um StackSet (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na página StackSets selecione o nome do StackSet.

O CloudFormation exibe a página Detalhes dos StackSets para o StackSet selecionado.

3. Na página StackSets details (Detalhes dos conjuntos de pilhas), selecione a guia Operations (Operações) e selecione a operação de detecção de desvios.
4. Selecione Stop operation (Interromper operação).

Para interromper a detecção de desvios em um StackSet (AWS CLI)

- Use o comando [stop-stack-set-operation](#). Você deve fornecer o nome e o ID da operação de detecção de desvios do StackSet.


```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id 624af370-311a-11e8-b6b7-500cexample
```

Importar pilhas para o CloudFormation StackSets

Uma operação de importação de pilhas pode importar pilhas existentes para StackSets novos ou existentes, de modo que você possa migrar as pilhas existentes para um StackSet em uma única operação.

Para StackSets autogerenciados, a operação de importação pode importar pilhas na conta do administrador ou em diferentes contas e Regiões da AWS de destino. Para StackSets gerenciados por serviços, a operação de importação pode importar qualquer pilha na mesma AWS Organizations que a conta de gerenciamento.

As seguintes considerações e limitações se aplicam à importação de pilhas para StackSets:

- A operação de importação pode importar até dez pilhas utilizando IDs de pilha em linha ou até 200 pilhas utilizando um objeto do Amazon S3.
- A propriedade NoEcho não é compatível. As pilhas que contêm NoEcho não serão importadas para novos StackSets por meio da importação do StackSet.
- As pilhas apenas podem pertencer a um StackSet.
- Você pode implementar tags de pilhas no StackSet, especificando tags explicitamente como parâmetros na operação de importação de pilhas.
- Substituições de parâmetros personalizados de uma pilha não são afetadas durante a operação de importação.
- Os parâmetros das instâncias de pilha importadas não podem ser atualizados usando a opção Editar detalhes do StackSet. Você deve usar a opção Substituir parâmetros do StackSet. Para obter mais informações sobre parâmetros de substituição, consulte [Substituir parâmetros em pilhas](#).
- Cotas e instâncias de pilha de StackSets são aplicáveis ao importar pilhas. Para obter mais informações sobre cotas, consulte [Entender as cotas do CloudFormation](#).

Tópicos

- [Importação de pilhas autogerenciadas para StackSets do CloudFormation](#)

- [Importação de pilhas gerenciadas para StackSets do CloudFormation](#)
- [Reverter importações de pilhas para o CloudFormation StackSets](#)

Importação de pilhas autogerenciadas para StackSets do CloudFormation

A operação importação de pilhas do CloudFormation pode importar pilhas existentes para StackSets novos ou existentes para que você possa migrar pilhas existentes para um StackSet em uma só operação. Ao utilizar a importação de pilhas, você evita tempo de inatividade e interrupções por não precisar excluir e recriar esses recursos. Depois que a pilha é importada para um StackSet, a pilha original se torna uma instância da pilha do conjunto de pilhas especificado.

Considerações sobre as importações de pilhas autogerenciadas

- A operação de importação de pilhas requer uma conta de administrador na qual você cria um StackSet e uma conta de destino que contenha uma pilha.
- A conta de destino deve ter permissão para utilizar a operação `GetTemplate` com a especificação do ID da pilha ou do ARN. Por isso, as permissões `AWSCloudFormationStackSetsAdministration` ou `AWSCloudFormationStackSetsExecutionRole` devem ser concedidas à sua conta de administrador.

Tópicos

- [Importar uma pilha existente para um novo StackSet \(console\)](#)
- [Importar uma pilha existente para um StackSet existente \(console\)](#)
- [Importar uma pilha para um StackSet \(AWS CLI\)](#)

Importar uma pilha existente para um novo StackSet (console)

Antes de começar, identifique a pilha que deseja importar.

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets.
3. No topo da página StackSets, escolha Create StackSet (Criar StackSet).
4. Na página Escolher um modelo, escolha um modelo por meio de uma das seguintes opções e escolha Avançar.

- Escolha Amazon S3 URL e especifique o URL do modelo na caixa de texto.
 - Escolha Upload a template file (Fazer upload de um arquivo de modelo) e procure o arquivo de modelo.
 - Escolha From stack ID (Do ID da pilha) e insira seu ID da pilha.
5. Na página Especificar detalhes do StackSet, insira o nome de um StackSet que você deseja criar e escolha Avançar.

(Opcional) Insira uma descrição do StackSet.

6. Na página Configurar opções do StackSet, reveja suas seleções e escolha Avançar.
7. Na página Definir opções de implantação, escolha Importar pilhas para o conjunto de pilhas.
8. Insira o ID da pilha a ser importada no campo Stacks to import (Pilhas para importar).
Por exemplo, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Opcional) Escolha Adicionar outro ID de pilha e insira o ID da outra pilha que você deseja importar. É possível adicionar até 10 pilhas por operação de importação.

9. Revise as opções de implantação e escolha Avançar.
10. Na página Analisar, analise suas escolhas e as propriedades do seu StackSet. Quando estiver tudo pronto para importar a pilha para o StackSet, escolha Enviar.

Resultados: a pilha importada é agora uma instância de pilha do StackSet especificado. Para saber mais sobre o status de importação das pilhas, consulte [Códigos de status do StackSets](#).

Importar uma pilha existente para um StackSet existente (console)

Antes de começar, identifique a pilha que deseja importar.

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets.
3. Na página StackSets, escolha o StackSet para o qual você deseja importar uma pilha.
4. Com o StackSet selecionado, escolha Adicionar pilhas ao StackSet no menu Ações.
5. Na página Definir opções de implantação, escolha Importar pilhas para o conjunto de pilhas e insira o ID da pilha que você deseja importar no campo Pilhas a importar. Por exemplo,

arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786.

(Opcional) Escolha Adicionar outro ID de pilha e insira o ID da outra pilha que você deseja importar. É possível adicionar até 10 pilhas por operação de importação.

6. Escolha Próximo.

7. Na página Especificar substituições, verifique suas seleções e escolha Avançar.

8. Na página Analisar, analise suas escolhas e as propriedades do seu StackSet. Quando estiver tudo pronto para criar seu StackSet, escolha Enviar.

Resultados: a pilha importada é agora uma instância de pilha do StackSet especificado. Para saber mais sobre o status de importação das pilhas, consulte [Códigos de status do StackSets](#).

Importar uma pilha para um StackSet (AWS CLI)

Para importar uma pilha existente para um novo StackSet

O comando `create-stack-set` a seguir cria um StackSet e importa a pilha especificada. A pilha a ser importada é identificada pelo ARN. Substitua o espaço reservado de texto por suas próprias informações.

```
aws cloudformation create-stack-set \  
  --stack-set-name MyStackSet \  
  --stack-id arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/466df9e0-0dff-08e3-8e2f-5088487c4896 \  
  --administration-role-arn arn:aws:iam::123456789012:role/AWSCloudFormationStackSetAdministrationRole \  
  --execution-role-name AWSCloudFormationStackSetExecutionRole
```

Para importar uma pilha existente para um StackSet existente

O comando `import-stacks-to-stack-sets` a seguir importa a pilha especificada para o StackSet *MyStackSet*. A pilha a ser importada é identificada pelo ARN. Substitua o espaço reservado de texto por suas próprias informações.

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set MyStackSet \  
  --stack-ids arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786
```

Para especificar mais de uma pilha, use o formato apresentado a seguir para o valor da opção `--stack-ids`.

```
--stack-ids "arn_1" "arn_2"
```

Para clonar a pilha importada em outras regiões e contas

O comando `create-stack-instances` a seguir adiciona instâncias de pilha ao StackSet. Substitua o espaço reservado de texto por suas próprias informações.

```
aws cloudformation create-stack-instances \  
  --stack-set-name MyStackSet \  
  --accounts '["account_ID_1","account_ID_2"]' \  
  --regions '["region_1","region_2"]'
```

Importação de pilhas gerenciadas para StackSets do CloudFormation

A operação importação de pilhas do CloudFormation pode importar pilhas existentes para StackSets novos ou existentes para que você possa migrar pilhas existentes para um StackSet em uma só operação. O StackSets amplia a funcionalidade das pilhas, permitindo que você crie, atualize ou exclua pilhas em várias contas e regiões com uma única operação.

Considerações sobre importações de pilhas gerenciadas pelo serviço

- A operação de importação de pilhas requer uma conta de gerenciamento ou uma conta de administrador delegada na qual você possa gerenciar o AWS Organizations associado, por exemplo, habilitando o acesso confiável com o StackSets.
- As contas de destino devem ser membros do AWS Organizations gerenciado pela conta de gerenciamento ou pela conta de administrador delegado.
- A pilha de destino existe em uma das UOs de destino.
- A conta de destino deve ser membro do AWS Organizations.
- O acesso do AWS Organizations deve estar no estado ACTIVATED para as organizações.
- As pilhas que estão sendo importadas devem estar presentes em qualquer uma das contas-membro, e não na conta de gerenciamento.

Tópicos

- [Importar uma pilha gerenciada pelo serviço para um novo StackSet \(console\)](#)

- [Criar e importar uma pilha gerenciada pelo serviço para um StackSet existente \(console\)](#)
- [Importar uma pilha gerenciada pelo serviço para um StackSet existente \(console\)](#)
- [Importar uma pilha gerenciada pelo serviço para um StackSet \(AWS CLI\)](#)

Importar uma pilha gerenciada pelo serviço para um novo StackSet (console)

Importar uma pilha para um novo StackSet usando o Console de gerenciamento da AWS

Para importar uma nova pilha para um StackSet, identifique uma pilha que contenha o recurso que você quer importar.

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets.
3. No topo da página StackSets, escolha Create StackSet (Criar StackSet).
4. Na página Escolher um modelo, faça o seguinte:
 - a. Em Modelo de permissão do StackSet, escolha Permissões gerenciadas pelo serviço.
 - b. Em Pré-requisito: preparar modelo, escolha O modelo está pronto, depois escolha o modelo usando uma das seguintes opções:
 - Em URL do Amazon S3, insira seu URL do Amazon S3 no campo URL do Amazon S3.
 - Em Carregar um arquivo de modelo, escolha um modelo do CloudFormation no computador local.

Aceite suas configurações e escolha Next (Próximo).

5. Na página Especificar detalhes do StackSet, faça o seguinte:
 - a. Insira um nome de StackSet na caixa Nome do StackSet.
 - b. (Opcional) Insira uma descrição na seção StackSet description (Descrição do StackSet).

Na página Configurar opções do StackSet, reveja suas seleções e escolha Avançar.

6. Adicione as seguintes informações à página Definir opções de implantação:
 - a. Em Adicionar pilhas ao conjunto de pilhas, escolha Importar pilhas para o conjunto de pilhas.

- b. Em Stacks to import (Pilhas para importar), escolha seu método de importação de pilhas.
 - i. Para Stack ID (ID da pilha), insira o ID da pilha.
 - ii. Em URL da pilha, insira o URL do Amazon S3.
 7. Em Associar unidades organizacionais, faça o seguinte:
 - a. Escolha Associar com organização para usar a UO-raiz.
 - b. Escolha Associar com unidades organizacionais (UOs) para inserir os IDs das UOs superiores para as pilhas a serem importadas. Por exemplo, se Stack 1 e Stack 2 estiverem em OU1 e Stack 3 estiver em OU2, insira OU1 e OU2.

Aceite suas configurações e escolha Next (Próximo).

 8. Revise suas configurações na página Review (Revisar) e selecione Submit (Enviar).

Criar e importar uma pilha gerenciada pelo serviço para um StackSet existente (console)

Para importar uma pilha existente para um novo StackSet, identifique uma pilha que contenha o recurso que você quer importar.

Para criar um StackSet e importar uma pilha

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets.
3. No topo da página StackSets, escolha Create StackSet (Criar StackSet).
4. Na página Escolher um modelo, faça o seguinte:
 - a. Em Modelo de permissão do StackSet, escolha Permissões gerenciadas pelo serviço.
 - b. Em Pré-requisito: preparar modelo, escolha O modelo está pronto, depois escolha o modelo usando uma das seguintes opções:
 - Em URL do Amazon S3, insira seu URL do Amazon S3 no campo URL do Amazon S3.
 - Em Carregar um arquivo de modelo, escolha um modelo do CloudFormation no computador local.

Aceite suas configurações e escolha Next (Próximo).

5. Na página Especificar detalhes do StackSet, faça o seguinte:
 - a. Insira um nome de StackSet na caixa Nome do StackSet.
 - b. (Opcional) Insira uma descrição na seção StackSet description (Descrição do StackSet).

Na página Configurar opções do StackSet, reveja suas seleções e escolha Avançar.

6. Adicione as seguintes informações à página Definir opções de implantação:
 - Em Add stacks to stack set (Adicionar pilhas ao conjunto de pilhas), escolha Deploy new stacks (Implantar novas pilhas).
7. Na seção Associar unidades organizacionais, faça o seguinte:
 - a. Escolha Associar com organização para usar a UO-raiz.
 - b. Escolha Associar com unidades organizacionais (UOs) para inserir os IDs das UOs superiores para as pilhas a serem importadas. Por exemplo, se Stack 1 e Stack 2 estiverem em OU1 e Stack 3 estiver em OU2, insira OU1 e OU2.
8. Para Specify regions (Especificar regiões) e Deployment options (Opções de implantação), examine suas escolhas.

Aceite suas configurações e escolha Next (Próximo).

9. Revise suas configurações na página Review (Revisar) e selecione Submit (Enviar).

Importar uma pilha gerenciada pelo serviço para um StackSet existente (console)

Escolha o StackSet e identifique a pilha que você deseja importar.

Para importar uma pilha para um StackSet existente

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. No painel de navegação, escolha StackSets.
3. Escolha o StackSet para o qual você deseja importar uma pilha e depois Adicionar pilhas ao StackSet na lista suspensa Ações.
4. Adicione as seguintes informações à página Definir opções de implantação:

- a. Em Adicionar pilhas ao conjunto de pilhas, escolha Importar pilhas para o conjunto de pilhas.
- b. Em Pilhas a importar, faça o seguinte:
 - i. Para Stack ID (ID da pilha), insira o ID da pilha.
 - ii. Em URL da pilha, insira o URL do Amazon S3.
- c. Em Associar unidades organizacionais, faça o seguinte:
 - i. Escolha Associar com organização para usar a UO-raiz.
 - ii. Escolha Associar com unidades organizacionais (UOs) para inserir os IDs das UOs superiores para as pilhas a serem importadas. Por exemplo, se Stack 1 e Stack 2 estiverem em OU1 e Stack 3 estiver em OU2, insira OU1 e OU2.

Aceite suas configurações e escolha Next (Próximo).

5. Revise a página Specify overrides (Especificar substituições) e escolha Next (Próximo).
6. Confirme e revise a página Review (Revisar) e escolha Submit (Enviar).

Importar uma pilha gerenciada pelo serviço para um StackSet (AWS CLI)

Após a criação de um StackSet, você pode importar suas pilhas passando os IDs das pilhas que estão sendo importadas. Também é possível transmitir a lista de IDs de UOs para a qual você quer mapeá-lo.

O CloudFormation importará as pilhas fornecidas pelo usuário dentro dessas UOs e as usará como destinos de implantação para o StackSet. Os IDs de pilhas apresentados na entrada serão mapeados internamente para a UO mais próxima na entrada da lista de IDs de UOs e na UO. Se uma pilha não pertencer a um ID de UO existente na lista de entrada, a AWS CLI retornará o erro `StackNotFoundException`.

A operação `import-stacks-to-stack-set` cria instâncias de pilha para as pilhas na entrada de ID de UO. Os exemplos a seguir da AWS CLI usam a operação `import-stacks-to-stack-set` para importar uma pilha para um StackSet.

- Para utilizar a operação `import-stacks-to-stack-sets`, especifique `stack-ids` ou `stack-ids-url` que você quer importar para o seu conjunto de pilhas.

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceManagedStackSet \  
  --stack-ids "arn:123456789012:us-east-1:Stack1" \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceManagedStackSet \  
  --stack-ids-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-  
name.json \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

Note

A operação `import-stacks-to-stack-sets` requer a especificação de pelo menos um ID de unidade organizacional (ID de UO) para que seja possível associar a pilha que está sendo importada para essa UO específica. Essa operação não cria instâncias de pilha para outras contas-membro nas UOs associadas. Para atualizar contas-membro para as UOs associadas, use `create-stack-instances` ou `update-stack-instances`.

O `create-stack-set` cria instâncias de pilha para todas as contas nas UOs com um modelo fornecido pelo usuário, seja por upload direto ou via Amazon S3. Os exemplos a seguir da AWS CLI usam a operação `create-stack-set` para importar uma pilha para um novo StackSet.

- Para usar a operação `create-stack-set`, especifique o nome do StackSet e importe uma pilha para um StackSet recém-criado.

```
aws cloudformation create-stack-set \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-  
name.json \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment Enabled=true
```

Reverter importações de pilhas para o CloudFormation StackSets

Se houver alterações indesejadas na instância da pilha, você poderá reverter a importação da instância da pilha.

Quando você reverte a importação de uma instância de pilha, o CloudFormation exclui a instância de pilha do StackSet, mas retém os recursos da pilha.

Para reverter uma operação de importação de pilha, realize o procedimento a seguir.

1. Especifique um atributo `DeletionPolicy` de `Retain` para cada recurso que você deseja manter após a exclusão da instância de pilha. Para obter mais informações, consulte [Reverter uma operação de importação](#).
2. Exclua as instâncias de pilha do seu StackSet. Para obter mais informações, consulte [Excluir pilhas do CloudFormation StackSets](#).
3. Exclua seu StackSet. Para obter mais informações, consulte [Excluir CloudFormation StackSets](#).

Práticas recomendadas para usar o CloudFormation StackSets

Esta seção descreve as práticas recomendadas para definir um modelo de StackSet, criar ou adicionar pilhas a um StackSet ou atualizar um StackSet.

Se você não tiver experiência com o CloudFormation, revise o tópico [Práticas recomendadas do CloudFormation](#) para obter mais recomendações que podem ajudar você a usar o CloudFormation de forma mais eficaz e segura.

Tópicos

- [Definir o modelo](#)
- [Como criar ou adicionar pilhas ao StackSet](#)
- [Como atualizar pilhas em um StackSet](#)

Definir o modelo

- Defina o modelo que você deseja padronizar em várias contas, dentro de várias regiões.
- Ao criar o modelo, certifique-se de que não há conflitos de nomenclatura nos recursos globais (como as funções do IAM e os buckets do Amazon S3) quando eles são criados em mais de uma região na mesma conta.

- Um StackSet tem um único modelo e conjunto de parâmetros. A mesma pilha é criada em todas as contas que são associadas a um StackSet. Ao criar seus modelos, faça-o em um nível de detalhes suficiente para permitir um bom equilíbrio de controle e padronização.
- Recomendamos que você armazene seu modelo em um bucket do Amazon S3.

Como criar ou adicionar pilhas ao StackSet

- Verifique se a adição de instâncias de pilha ao StackSet inicial funciona antes de adicionar quantidades maiores de instâncias de pilha ao seu StackSet.
- Escolha as opções de implantação (rollout) que funcionam para o seu caso de uso.
 - Para uma implantação mais conservadora, defina Máximo de contas concorrentes como 1 e Tolerância a falhas como 0. Defina a região de menor impacto para ser a primeira na lista Ordem das regiões. Comece com uma região.
 - Para uma implantação mais rápida, aumente os valores de Máximo de contas concorrentes e Tolerância a falhas, conforme o necessário.
- As operações em StackSets dependem de quantas instâncias de pilha estão envolvidas e podem levar bastante tempo.

Como atualizar pilhas em um StackSet

- Por padrão, a atualização de um StackSet atualiza todas as instâncias de pilha. Se você tiver vinte contas em duas regiões, terá quarenta instâncias de pilha, e todas serão atualizadas quando você atualizar o StackSet.

Para StackSets com um grande número de instâncias de pilha, recomendamos que, para testar a versão atualizada de um modelo, você atualize as instâncias de pilha seletivamente em algumas contas de teste antes de atualizar todas as instâncias de pilha.

- Para obter um controle mais granular sobre a atualização de pilhas individuais em seu StackSet, planeje criar vários StackSets.
- Atualizar um StackSet que contém um grande número de pilhas pode levar um tempo significativo. Nesta versão, apenas uma operação é permitida por vez em um StackSet. Planeje suas atualizações para não ser impedido de executar outras operações no StackSet.

Modelos de exemplo do CloudFormation StackSets

Esta seção inclui links para alguns modelos de exemplos do CloudFormation que podem ajudar você a usar o CloudFormation StackSets na sua empresa. Os modelos listados nesta seção habilitam o [AWS CloudTrail](#) ou o [AWS Config](#) e as regras dentro dele.

Important

Como prática recomendada de segurança ao permitir o acesso do AWS Config a um bucket do Amazon S3, recomendamos fortemente que você restrinja o acesso na política de bucket com a condição `AWS:SourceAccount`. Novos modelos são atualizados para terem `AWS:SourceAccount`. Se sua política de bucket atual não seguir essa prática recomendada de segurança, recomendamos fortemente que você a edite para incluir essa proteção. Isso garante que o AWS Config tenha acesso somente em nome dos usuários esperados.

Descrição	Link do S3
Habilitar AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSCloudtrail.yml
Habilitar AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfig.yml
Habilitar o AWS Config com registro central em log	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfigForOrganizations.yml
Habilitar as políticas padrão do Amazon Data Lifecycle Manager em uma organização da AWS ou em contas específicas da AWS	- Política padrão para snapshots do EBS: https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yml - Política padrão para AMIs baseadas no EBS: https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AMIDefaultPolicy.yml

Descrição	Link do S3
	stackset-sample-templates-us-east-1/Da taLifecycleManagerAMIDefaultPolicy.yaml https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml
Configurar uma regra do AWS Config para determinar se o CloudTrail está habilitado	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleCloudtrailEnabled.yml
Configure uma regra do AWS Config para determinar se o MFA raiz está habilitado	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml
Configure uma regra do AWS Config para determinar se os EIPs estão anexados	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEipAttached.yml
Configure uma regra do AWS Config para determinar se os volumes do EBS estão criptografados	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Solução de problemas de CloudFormation StackSets

Este tópico contém alguns problemas comuns do StackSets e soluções sugeridas para esses problemas.

Tópicos

- [Motivos comuns de falha na operação da pilha](#)
- [Realizar novamente operações de criação ou atualização de pilha com falha](#)
- [Falha de exclusão de instância de pilha](#)
- [Falha na operação de importação de pilhas](#)
- [Contagem de falhas de instâncias de pilha para operações do StackSets](#)

Motivos comuns de falha na operação da pilha

Problema: uma operação de pilha falhou, e o status da instância de pilha é OUTDATED.

Causa: pode haver várias causas comuns para a falha na operação da pilha.

- Permissões insuficientes em uma conta de destino para criar recursos especificados no seu modelo.
- O modelo do CloudFormation pode conter erros. Valide o modelo no CloudFormation e corrija os erros antes de tentar criar seu StackSet.
- O modelo pode estar tentando criar recursos globais que precisam ser únicos, mas não são. Por exemplo, os buckets do S3.
- Não existe um número de conta de destino especificado. Verifique os números da conta de destino que você especificou na página Definir opções de implantação do assistente.
- A conta de administrador não possui uma relação de confiança com a conta de destino.
- A quantidade máxima especificada para um recurso no seu modelo já existe na sua conta de destino. Por exemplo, você pode ter atingido o limite de permitido de funções IAM em uma conta de destino, mas o modelo cria mais funções IAM.
- Você atingiu a quantidade máxima permitida de pilhas em um StackSet. Para saber a quantidade máxima de pilhas por StackSet, consulte [Entender as cotas do CloudFormation](#).

Solução: para obter mais informações sobre as permissões necessárias das contas de destino e administrador antes de criar StackSets, consulte [Concessão de permissões para o gerenciamento de pilhas em todas as contas de destino a todos os usuários da conta do administrador](#).

Realizar novamente operações de criação ou atualização de pilha com falha

Problema: uma operação de criação ou atualização de pilha falhou, e o status da instância de pilha é OUTDATED. Para saber o motivo da falha de uma tentativa de criação ou atualização de pilha, abra o console do CloudFormation e veja os eventos da pilha, que exibirão um status DELETED (para operações de criação com falha) ou FAILED (para operações de atualização com falha). Navegue pelos eventos de pilha e localize a coluna Motivo do status. O valor de Motivo do status explica o motivo da falha na operação.

Depois de corrigir a causa subjacente da falha na criação da pilha, você poderá tentar realizar a criação da pilha novamente. Para isso, execute as etapas a seguir.

Solução: execute as seguintes etapas para tentar realizar a operação da pilha novamente.

1. No console, selecione o StackSet que contém a pilha em que a operação falhou.
2. No menu Actions (Ações), escolha Edit StackSet details (Editar detalhes do StackSet) para tentar criar ou atualizar as pilhas novamente.
3. Na página Especificar modelo, mantenha a opção padrão Usar modelo atual para usar o mesmo modelo do CloudFormation. Se sua operação de pilha falhou porque o modelo necessário foi alterado e você deseja enviar um modelo revisado, escolha Carregar um modelo no Amazon S3 e, em seguida, Procurar para selecionar o modelo atualizado. Quando você terminar de carregar do modelo revisado, escolha Próximo.
4. Na página Especificar detalhes da pilha, escolha Próximo se você não estiver alterando nenhum parâmetro específico do modelo.
5. Na página Definir opções de implantação, altere os padrões para Máximo de contas simultâneas e Tolerância a falhas, se desejar. Para ter mais informações sobre essas configurações, consulte [Opções de operação de StackSets](#).
6. Na página Revisão, revise suas seleções e marque a caixa de seleção para confirmar os recursos do IAM necessários. Selecione Enviar.
7. Se sua pilha não for atualizada com êxito, repita este procedimento depois de ter resolvido todos os problemas subjacentes que estejam impedindo a criação de pilhas.

Falha de exclusão de instância de pilha

Problema: houve uma falha de exclusão de pilha.

Causa: a exclusão de pilha falhará para as pilhas em que a proteção contra encerramento foi ativada.

Solução: determine se a proteção contra encerramento foi ativada para a pilha. Caso tenha sido, desative a proteção contra encerramento e, em seguida, execute a exclusão de instância de pilha novamente.

Falha na operação de importação de pilhas

Problema: uma operação de importação de pilhas falha ao importar pilhas existentes para StackSets novos ou existentes. A instância da pilha está em um status INOPERABLE.

Solução: reverta a operação de importação de pilhas, concluindo as seguintes tarefas.

1. Use a opção Excluir pilhas do StackSets, habilite RetainStacks durante a configuração e depois exclua instâncias de pilha do seu StackSet. Para obter mais informações, consulte [Excluir pilhas do CloudFormation StackSets](#).
2. Você verá que as instâncias de pilha do StackSet são atualizadas para remover a instância de pilha INOPERABLE.
3. Corrija as instâncias da pilha de acordo com o erro de falha na importação e repita a operação de importação da pilha.

Contagem de falhas de instâncias de pilha para operações do StackSets

A contagem de falhas de instâncias de pilha alerta você se houver falha no provisionamento ou na atualização de instâncias de pilha. Essas instâncias da pilha não foram implantados por um ou mais dos seguintes motivos:

- Recursos existentes com uma configuração semelhante
- Dependências ausentes, como perfis do AWS Identity and Access Management (IAM)
- Outros fatores conflitantes

Se você quiser implantar com o máximo de simultaneidade, a contagem máxima de simultaneidade é uma mais do que a contagem de tolerância a falhas. Por exemplo, se a contagem de tolerância a falhas for 9, a contagem máxima de simultaneidade não poderá ser maior que 10. Isso fará com que a operação retorne SUCCEEDED mesmo se houver falha algumas na atualização de algumas instâncias de pilha. A nova contagem de falhas de instâncias de pilha permite que você determine se a operação só foi bem-sucedida condicionalmente porque a contagem de tolerância a falhas está definida para permitir todas as falhas.

Você pode usar o Console de gerenciamento da AWS, o AWS SDK ou a AWS CLI para obter a contagem de falhas e filtrar as instâncias de pilha para determinar quais instâncias precisam ser reimplantadas.

Utilizar o console

Para ver o número de instâncias de pilha em que houve falha:

1. Abra o [console do CloudFormation](#) e escolha StackSets.
2. Escolha o StackSet e selecione a guia Operações.

3. Escolha um status na coluna Status para visualizar detalhes do status. Você encontrará o número de instâncias de pilha em que houve falha para uma determinada operação nos detalhes do status.

Para visualizar a conta, a região e o status das instâncias de pilha para a operação:

1. Nos detalhes do status, escolha a contagem das instâncias de pilha em que houve falha. Exemplo: instâncias de pilha: **<number of failed stack instances>**.
2. Expanda o painel lateral escolhendo o cabeçalho do painel. Os resultados no painel lateral são os status das instâncias de pilha após a conclusão da operação selecionada.

Para ver os detalhes da instância de pilha atual para uma operação:

1. Escolha a guia Stack Instances (Instâncias de pilha).
2. Filtre por Last operation ID (ID da última operação). Os resultados são os status atuais e os motivos dos status da última operação a modificar a instância. É possível usar esse filtro em combinação com Conta da AWS, Região da AWS, Status detalhado e Status de desvio para refinar ainda mais os resultados da pesquisa.

Como usar o AWS CLI

Para obter o número de instâncias de pilha em que houve falha, chame `describe-stack-set-operation` ou `list-stack-set-operations` e consulte `StatusDetails`.

```
aws cloudformation describe-stack-set-operation --stack-set-name ss1 \  
--operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60
```

```
{  
  "StackSetOperation": {  
    "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",  
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",  
    "Action": "CREATE",  
    "Status": "SUCCEEDED",  
    "OperationPreferences": {  
      "RegionOrder": [],  
      "FailureToleranceCount": 10,  
      "MaxConcurrentCount": 10  
    },  
  },  
}
```

```

    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
    "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
    "StatusDetails": {
        "FailedStackInstancesCount": 3
    }
}
}
}

```

```
aws cloudformation list-stack-set-operations --stack-set-name ss1
```

```

{
  "Summaries": [
    {
      "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",
      "Action": "CREATE",
      "Status": "SUCCEEDED",
      "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
      "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
      "StatusDetails": {
        "FailedStackInstancesCount": 3
      },
      "OperationPreferences": {
        "RegionOrder": [],
        "FailureToleranceCount": 10,
        "MaxConcurrentCount": 10
      }
    }
  ]
}

```

Para obter uma visão geral histórica de uma determinada operação, use `list-stack-set-operation-results` para visualizar o status e o motivo do status de cada instância de pilha após a conclusão da operação. Veja `Status` e `StatusReason` no seguinte exemplo:

```

aws cloudformation list-stack-set-operation-results --stack-set-name ss1 \
  --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60 --
  filters Name=OPERATION_RESULT_STATUS,Values=FAILED

```

```

{
  "Summaries": [
    {
      "Account": "123456789012",
      "Region": "us-west-2",
      "Status": "FAILED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "AccountGateResult": {
        "Status": "SKIPPED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
      },
      "OrganizationalUnitId": ""
    },
    {
      "Account": "123456789012",
      "Region": "us-west-1",
      "Status": "FAILED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "AccountGateResult": {
        "Status": "SKIPPED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
      },
      "OrganizationalUnitId": ""
    },
    {
      "Account": "123456789012",
      "Region": "us-east-1",
      "Status": "FAILED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "AccountGateResult": {
        "Status": "SKIPPED",

```

```

        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
    },
    "OrganizationalUnitId": ""
}
]
}

```

Use `list-stack-instances` com os filtros `DETAILED_STATUS` e `LAST_OPERATION_ID` para obter uma lista das instâncias de pilha em que houve falha na última operação que tentou implantar a instância de pilha. Veja o sinalizador `--filters` no exemplo com `DETAILED_STATUS` e `LAST_OPERATION_ID`:

```

aws cloudformation list-stack-instances --stack-set-name ss1 \
  --filters Name=DETAILED_STATUS,Values=FAILED Name=LAST_OPERATION_ID,Values=5550e62f-
c822-4331-88fa-21c1d7bafc60

```

```

{
  "Summaries": [
    {
      "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
      "Region": "us-east-1",
      "Account": "123456789012",
      "Status": "OUTDATED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "StackInstanceStatus": {
        "DetailedStatus": "FAILED"
      },
      "OrganizationalUnitId": "",
      "DriftStatus": "NOT_CHECKED",
      "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    },
    {
      "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
      "Region": "us-west-1",
      "Account": "123456789012",
      "Status": "OUTDATED",

```

```

        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    },
    {
        "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
        "Region": "us-west-2",
        "Account": "123456789012",
        "Status": "OUTDATED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    }
]
}

```

Para encontrar o ID da última operação para modificar uma instância de pilha, use `list-stack-instances` ou `describe-stack-instance` para obter o `LastOperationId`:

```

aws cloudformation describe-stack-instance --stack-set-name ss1 \
--stack-instance-account 123456789012 --stack-instance-region us-east-2

```

```

{
  "StackInstance": {
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Region": "us-west-2",
    "Account": "123456789012",
    "ParameterOverrides": [],
    "Status": "OUTDATED",
    "StackInstanceStatus": {
      "DetailedStatus": "FAILED"
    }
  }
}

```

```
    },  
    "StatusReason": "Account 123456789012 should have  
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role  
'AWSCloudFormationStackSetAdministrationRole'.",  
    "OrganizationalUnitId": "",  
    "DriftStatus": "NOT_CHECKED",  
    "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"  
  }  
}
```

Sincronizar as pilhas com o código-fonte armazenado em um repositório Git com a sincronização do Git

Com o Git Sync, é possível gerenciar as pilhas do CloudFormation com o controle de fonte. Você faz isso configurando o CloudFormation para monitorar um repositório Git. O repositório é monitorado quanto a alterações em dois arquivos:

- Um arquivo de modelo do CloudFormation que define uma pilha
- Um arquivo de implantação da pilha que contém os parâmetros que configuram a pilha

Com o Git sync, é possível usar pull requests e o rastreamento de versão para configurar, implantar e atualizar as pilhas do CloudFormation de um local centralizado. Quando você confirma alterações para o modelo ou para o arquivo de implantação, o CloudFormation atualiza a pilha automaticamente. Se você usa pull requests, o CloudFormation pode deixar um comentário sobre a pull request explicando quais alterações serão feitas na sua pilha antes de realmente atualizá-la. No entanto, esse recurso deverá ser habilitado primeiro.

O Git Sync fornece uma interface de console que você pode usar para vincular a um repositório, gerar um arquivo de implantação da pilha, atualizar um modelo do CloudFormation e enviar uma solicitação pull ao seu repositório. Além disso, o Git Sync fornece um painel de status que pode ser usado para monitorar, editar e solucionar problemas de implantações ativas da pilha do Git Sync. O Git Sync é acessado por meio do [console do CloudFormation](#) quando você [cria uma pilha](#). Você também pode acessar a sincronização do Git usando CodeConnections. Para obter mais informações, consulte [Como trabalhar com configurações de sincronização para repositórios vinculados](#) no Guia do usuário do Console do Developer Tools.

A sincronização do Git é compatível com os repositórios do [GitHub](#), [GitHub Enterprise](#), [GitLab](#), [Bitbucket](#) e [GitLab autogerenciado](#).

Note

O Git Sync está disponível nas seguintes regiões: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon), Canadá (Central), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Tóquio), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Europa (Irlanda), Europa (Londres), Europa (Paris), Europa (Estocolmo), Europa (Frankfurt), Europa (Milão) e América do Sul (São Paulo).

Para obter informações sobre como usar o Git sync com uma estratégia de várias contas, consulte a postagem do blog a seguir [Use CloudFormation Git sync to configure resources in customer accounts](#).

Tópicos

- [Como o Git sync funciona com o CloudFormation](#)
- [Pré-requisitos para sincronizar pilhas com um repositório Git usando o Git sync](#)
- [Criar uma pilha com base no código-fonte do repositório com a sincronização do Git](#)
- [Permitir que o CloudFormation publique um resumo das alterações de pilha em solicitações pull](#)
- [Painel de status do Git Sync](#)

Como o Git sync funciona com o CloudFormation

Este tópico descreve como a sincronização do Git funciona e apresenta os principais conceitos necessários para trabalhar com ela.

Tópicos

- [Como a sincronização do Git funciona](#)
- [Comentários sobre solicitações pull](#)
- [Arquivo de implantação da pilha](#)
- [Arquivo de modelo do CloudFormation](#)
- [Repositório de definição de modelo](#)

Como a sincronização do Git funciona

Para usar a sincronização do Git sync, primeiro você deve conectar um provedor do Git ao CloudFormation usando o serviço [CodeConnections](#). Nos procedimentos deste guia, a conexão é criada por meio do console do CodeConnections. Se preferir, crie a conexão com a AWS CLI. É possível usar os seguintes provedores do Git:

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)

- [GitLab autogerenciado](#)

A seguir, crie um modelo do CloudFormation que defina a pilha e adicione-o ao repositório. Este arquivo de modelo é monitorado. O CloudFormation atualiza a pilha automaticamente quando as alterações são confirmadas para ela.

No console do CloudFormation, você cria uma pilha e escolhe Sincronizar a partir do Git para fazer com que o CloudFormation use a sincronização do Git. Você especificará o repositório e a ramificação que deseja que o CloudFormation monitore, bem como especificará o modelo do CloudFormation que define a pilha em seu repositório.

Durante a configuração, é possível fornecer o próprio arquivo de implantação da pilha do seu repositório ou fazer com que o Git Sync gere um arquivo para você. O arquivo de implantação da pilha contém parâmetros e valores que configuram os recursos na pilha. Este arquivo de implantação da pilha é monitorado. O CloudFormation atualiza a pilha automaticamente quando as alterações são confirmadas para ela.

O Git Sync cria uma solicitação pull em seu repositório para sincronizar a pilha com o arquivo de modelo do CloudFormation e o arquivo de implantação da pilha. Se o Git Sync gerar o arquivo de implantação da pilha para você, ele será enviado ao seu repositório pelo Git Sync.

Depois, mescle a solicitação pull com o repositório para que o CloudFormation provisione a pilha, configure-a com os parâmetros de implantação e comece a monitorar as alterações no repositório.

Daí em diante, sempre que você fizer alterações no arquivo de modelo ou no arquivo de implantação de pilha e as confirmar no repositório, o CloudFormation detectará automaticamente as alterações. Se sua equipe usa pull requests, os membros da equipe podem então revisar e aprovar as alterações antes que elas sejam implantadas. Depois que a pull request for aceita, o CloudFormation implantará suas alterações.

É possível monitorar o status da configuração de sincronização do Git para a pilha e ver um histórico de confirmações aplicadas à pilha no console do CloudFormation. Além disso, o console fornece ferramentas para a reconfiguração do Git Sync e para a solução de problemas.

Comentários sobre solicitações pull

Você pode optar por fazer com que o CloudFormation crie um resumo das alterações de código nas pull requests por meio do serviço CodeConnections ativando a opção Habilitar comentário na pull request no console. Fornecer um resumo das alterações nas solicitações pull significa

que os membros da equipe podem facilmente revisar e entender o impacto das modificações propostas antes de mesclar a solicitação pull. Para obter mais informações, consulte [Permitir que o CloudFormation publique um resumo das alterações de pilha em solicitações pull](#).

Arquivo de implantação da pilha

Um arquivo de implantação da pilha corresponde a um arquivo formatado em JavaScript Object Notation (JSON) ou em YAML padrão que contém os parâmetros e os valores que gerenciam a pilha do CloudFormation. Ele é monitorado quanto a alterações. Quando alterações no arquivo são confirmadas no repositório, a pilha associada é atualizada automaticamente.

O arquivo de implantação da pilha contém um par chave/valor e dois dicionários:

- `template-file-path`

Este é o caminho completo do repositório para o arquivo de modelo do CloudFormation. O arquivo de modelo declara os recursos para a pilha do CloudFormation associada a este arquivo de implantação.

- `parameters`

O dicionário de parâmetros contém pares chave/valor que configuram os recursos na pilha. Um arquivo de implantação da pilha pode ter, no máximo, 50 parâmetros.

- `tags`

O dicionário de etiquetas contém pares chave/valor opcionais que você pode usar para identificar e categorizar os recursos na pilha. Um arquivo de implantação da pilha pode ter, no máximo, 50 etiquetas.

É possível fornecer o próprio arquivo de implantação da pilha ou fazer com que o Git Sync crie um arquivo para você e envie automaticamente uma solicitação pull para o seu repositório. Você pode gerenciar os parâmetros e as etiquetas ao editar o arquivo de implantação da pilha e ao confirmar as alterações no repositório.

Veja a seguir um exemplo de um arquivo de implantação da pilha do Git Sync:

```
template-file-path: fargate-srvs/my-stack-template.yaml

parameters:
  image: public.ecr.aws/lts/nginx:latest
```

```
task_size: x-small
max_capacity: 5
port: 8080
env: production
tags:
  cost-center: '123456'
  org: 'AWS'
```

Arquivo de modelo do CloudFormation

Um arquivo de modelo contém uma declaração dos recursos da AWS que compõem uma pilha do CloudFormation. Com o Git Sync, o arquivo de modelo é armazenado no repositório Git e referenciado pelo arquivo de implantação da pilha. É possível gerenciar a pilha ao editar o arquivo de modelo e ao confirmar as alterações para o repositório.

Para obter mais informações, consulte [Trabalhar com modelos do CloudFormation](#).

Repositório de definição de modelo

O repositório de definição de modelo corresponde ao repositório Git vinculado ao CloudFormation por meio do Git Sync. O repositório é monitorado quanto a alterações para o modelo do CloudFormation e para o arquivo de implantação da pilha. Quando você confirma as alterações para o arquivo, a pilha associada é atualizada automaticamente.

Important

Ao configurar o repositório de definição de modelo no console do Git Sync, selecione o repositório e a ramificação adequados da conexão do Git. O Git Sync monitora somente o repositório e a ramificação configurados em busca de alterações para o modelo do CloudFormation e para o arquivo de implantação da pilha.

Pré-requisitos para sincronizar pilhas com um repositório Git usando o Git sync

Antes de sincronizar uma pilha do CloudFormation com o repositório Git, verifique se os requisitos a seguir são atendidos.

Tópicos

- [Repositório Git](#)
- [Modelo do CloudFormation](#)
- [Perfil de serviço de sincronização do Git](#)
- [Permissões do IAM para usuários do console](#)

Repositório Git

É necessário ter um repositório Git hospedado em uma das plataformas a seguir.

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab autogerenciado](#)

O repositório pode ser público ou privado. Será necessário conectar esse repositório Git ao CloudFormation por meio do [console do Connections](#).

Modelo do CloudFormation

O repositório Git deve conter um [arquivo de modelo do CloudFormation](#) verificado na ramificação que você pretende conectar com a sincronização do Git. Esse modelo será referenciado pelo [arquivo de implantação da pilha](#).

Perfil de serviço de sincronização do Git

A sincronização do Git requer um perfil do IAM. É possível optar por criar um perfil do IAM para a pilha ao configurar a sincronização do Git ou usar um perfil existente.

Note

Um perfil do IAM gerado automaticamente aplica permissões somente à pilha para a qual o perfil é gerado. Para reutilizar um perfil do IAM gerado automaticamente, é necessário editar o perfil para a nova pilha.

Permissões necessárias para o perfil de serviço de sincronização do Git

O perfil do IAM fornecido para a sincronização do Git requer as permissões a seguir.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:ListChangeSets`
- `cloudformation:ValidateTemplate`
- `events:PutRule`
- `events:PutTargets`

Note

As permissões necessárias, citadas anteriormente, são adicionadas automaticamente aos perfis do IAM gerados pelo Git Sync.

O exemplo de perfil do IAM apresentado a seguir inclui as permissões de pré-requisitos para o Git Sync.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SyncToCloudFormation",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation>DeleteChangeSet",
        "cloudformation:DescribeChangeSet",
```

```

        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:GetTemplate",
        "cloudformation:ListChangeSets",
        "cloudformation:ListStacks",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{
    "Sid": "PolicyForManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy":
["cloudformation.sync.codeconnections.amazonaws.com"]
        }
    }
},
{
    "Sid": "PolicyForDescribingRule",
    "Effect": "Allow",
    "Action": "events:DescribeRule",
    "Resource": "*"
}
]
}

```

Política de confiança

Você deve fornecer a política de confiança apresentada a seguir ao criar o perfil para definir a relação de confiança.

JSON

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "CfnGitSyncTrustPolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "cloudformation.sync.codeconnections.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
}

```

Recomendamos o uso das chaves de condição `aws:SourceArn` e `aws:SourceAccount` para se proteger contra o problema `confused deputy`. A conta de origem é o ID da sua conta e o ARN de origem é o ARN da conexão no serviço [CodeConnections](#) que permite que o CloudFormation se conecte ao repositório do Git.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codeconnections:us-east-1:123456789012:connection/EXAMPLE64-8aad-4d5d-8878-dfcab0bc441f"
        }
      }
    }
  ]
}

```



```
}
```

Para obter mais informações sobre o problema confused deputy, veja [Prevenção contra o ataque do “substituto confuso” em todos os serviços](#).

Permissões do IAM para usuários do console

Para configurar com sucesso a sincronização do Git por meio do console do CloudFormation, os usuários finais também devem receber permissões por meio do IAM.

As permissões `codeconnections` a seguir são necessárias para criar e gerenciar a conexão com seu repositório do Git.

- `codeconnections:CreateRepositoryLink`
- `codeconnections:CreateSyncConfiguration`
- `codeconnections>DeleteRepositoryLink`
- `codeconnections>DeleteSyncConfiguration`
- `codeconnections:GetRepositoryLink`
- `codeconnections:GetSyncConfiguration`
- `codeconnections:ListRepositoryLinks`
- `codeconnections:ListSyncConfigurations`
- `codeconnections:ListTagsForResource`
- `codeconnections:TagResource`
- `codeconnections:UntagResource`
- `codeconnections:UpdateRepositoryLink`
- `codeconnections:UpdateSyncBlocker`
- `codeconnections:UpdateSyncConfiguration`
- `codeconnections:UseConnection`

Os usuários do console também devem ter as permissões do `cloudformation` a seguir para visualizar e gerenciar pilhas durante o processo de configuração da sincronização do Git.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`

- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:GetTemplate`
- `cloudformation:ListChangeSets`
- `cloudformation:ListStacks`
- `cloudformation:ValidateTemplate`

Note

Embora as permissões do conjunto de alterações (`cloudformation:CreateChangeSet`, `cloudformation>DeleteChangeSet`, `cloudformation:DescribeChangeSet`, `cloudformation:ExecuteChangeSet`) possam não ser estritamente necessárias para uso somente no console, elas são recomendadas para habilitar os recursos completos de inspeção e gerenciamento da pilha.

O exemplo de política do IAM a seguir inclui as permissões do usuário necessárias para configurar a sincronização do Git via console.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeConnectionsPermissions",
      "Effect": "Allow",
      "Action": [
        "codeconnections:CreateRepositoryLink",
        "codeconnections:CreateSyncConfiguration",
        "codeconnections>DeleteRepositoryLink",
        "codeconnections>DeleteSyncConfiguration",
        "codeconnections:GetRepositoryLink",
        "codeconnections:GetSyncConfiguration",
        "codeconnections:ListRepositoryLinks",

```

```

        "codeconnections:ListSyncConfigurations",
        "codeconnections:ListTagsForResource",
        "codeconnections:TagResource",
        "codeconnections:UntagResource",
        "codeconnections:UpdateRepositoryLink",
        "codeconnections:UpdateSyncBlocker",
        "codeconnections:UpdateSyncConfiguration",
        "codeconnections:UseConnection",
        "codeconnections:CreateForcedTargetSync",
        "codeconnections:CreatePullRequestForResource"
    ],
    "Resource": "*"
},
{
    "Sid": "CloudFormationConsolePermissions",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:DeleteChangeSet",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:GetTemplate",
        "cloudformation:ListChangeSets",
        "cloudformation:ListStacks",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
}
]
}

```

Note

Ao criar uma política do IAM que inclua as permissões `codeconnections:CreateForcedTargetSync` e `codeconnections:CreatePullRequestForResource`, é possível ver um aviso no console do IAM informando que essas ações não existem. Esse aviso pode ser ignorado e a política ainda será criada com sucesso. Essas permissões são necessárias para

determinadas operações de sincronização do Git, apesar de não serem reconhecidas pelo console do IAM.

Criar uma pilha com base no código-fonte do repositório com a sincronização do Git

Este tópico explica como criar uma pilha do CloudFormation que seja sincronizada com um repositório Git com o Git sync.

Important

Antes de continuar, preencha todos os [pré-requisitos](#) na seção anterior.

Criar uma pilha com base no código-fonte do repositório

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual a pilha deve ser criada.
3. Na página Pilhas, escolha Criar pilha e selecione Com novos recursos (padrão).
4. Na página Create a stack (Criar uma pilha), faça o seguinte:
 - a. Em Pré-requisito: preparar modelo, mantenha a opção Escolher um modelo existente selecionada.
 - b. Em Especificar modelo, escolha Sincronizar a partir do Git e Próximo.
5. Na página Especificar detalhes da pilha, em Nome da pilha, digite um nome para a sua pilha. Os nomes para as pilhas podem incluir letras (A-Z e a-z), números (0 a 9) e travessões (-).
6. Em Arquivo de implantação da pilha, Criação do arquivo de implantação:
 - Se você não criou um arquivo de implantação da pilha e o adicionou ao repositório, escolha Criar o arquivo usando os parâmetros a seguir e colocá-lo em meu repositório.
 - Se você tem um arquivo de implantação da pilha em seu repositório, escolha Estou fornecendo meu próprio arquivo em meu repositório.

7. Em Repositório de definição de modelo, selecione Escolher um repositório Git vinculado para escolher um repositório Git que já esteja vinculado ao CloudFormation ou Vincular um repositório Git para vincular um novo. Se você escolher Vincular um repositório Git, faça o seguinte:
 - a. Em Selecionar provedor de repositório, escolha uma destas opções:
 - GitHub
 - GitHub Enterprise Server
 - GitLab
 - Bitbucket
 - GitLab autogerenciado
 - b. Em Conexão, escolha uma conexão na lista. Se nenhuma opção aparecer na lista Conexão, selecione adicione uma nova conexão para acessar o [console do Connections](#) e crie uma conexão para o repositório.
8. Na lista Repositório, selecione o repositório Git que contém o arquivo de modelo da pilha.
9. Na lista Ramificação, selecione a ramificação que você deseja que o Git Sync monitore.

 Note

O Git Sync monitora somente a ramificação selecionada em busca de alterações no modelo do CloudFormation e nos arquivos de implantação da pilha. Quaisquer alterações que você gostaria de aplicar à sua pilha devem ser confirmadas para esta ramificação.

10. Para o Caminho do arquivo de implantação, especifique o caminho completo, incluindo o nome do arquivo de implantação da pilha na raiz da ramificação do repositório.

Se o CloudFormation estiver gerando o arquivo para você, é neste local que o arquivo será confirmado em seu repositório. Se você estiver fornecendo o arquivo, este será o local do arquivo em seu repositório.

11. Adicione um Perfil do IAM. O perfil do IAM inclui as permissões necessárias para que o CloudFormation sincronize a pilha pelo seu repositório Git. É possível escolher Novo perfil do IAM para gerar um novo perfil ou escolher Perfil do IAM existente para selecionar um perfil existente em sua Conta da AWS. Se você optar por gerar um novo perfil, as permissões necessárias serão incluídas no perfil.

12. Habilitar ou desabilitar comentários na solicitação pull:

- Para que o CloudFormation publique as informações do conjunto de alterações em solicitações pull para atualizações de pilha, mantenha a opção Habilitar comentário na solicitação pull ativada.
- Se você desativar essa opção, o CloudFormation não descreverá as diferenças entre a configuração atual da pilha e as alterações propostas nas solicitações pull quando os arquivos do repositório forem atualizados.

13. Para o Caminho do arquivo de modelo, especifique o caminho completo da raiz do seu repositório para o arquivo de modelo de pilha.

14. (Opcional) Para especificar os parâmetros da pilha, escolha Adicionar parâmetro, forneça uma chave e um valor para cada parâmetro e escolha Avançar. Para obter mais informações, consulte [Arquivo de implantação da pilha](#).

Por exemplo, para especificar um parâmetro **port=8080** no arquivo de implantação da pilha, faça o seguinte:

- a. Selecione Add parameter (Adicionar parâmetro).
- b. Em Chave, digite **port**.
- c. Em Valor, insira **8080**.

15. (Opcional) Para adicionar tags da pilha, escolha Adicionar nova tag, forneça uma chave e um valor para cada tag e, em seguida, escolha Próximo. Para obter mais informações, consulte [Arquivo de implantação da pilha](#).

16. Escolha Próximo a fim de prosseguir para Configurar opções da pilha. Para obter informações sobre como configurar as opções da pilha, consulte [Configurar opções da pilha](#).

Quando concluir a configuração da pilha, escolha Avançar para continuar.

17. Analise as configurações da pilha e confirme o seguinte:

- O modelo da pilha está configurado corretamente e definido como Sincronizar pelo Git.
- O arquivo de implantação está configurado corretamente.
- O repositório de definição de modelo está configurado corretamente e, em particular, o Repositório e o Nome da ramificação estão selecionados adequadamente.
- A pré-visualização do arquivo de implantação está correta e contém os parâmetros e os valores esperados.

18. Escolha Enviar para criar a pilha.

Após Enviar ser escolhido, uma solicitação pull é criada automaticamente em seu repositório Git. Você deve mesclar essa solicitação pull em seu repositório Git para criar sua pilha. Depois que a pilha é criada, o CloudFormation monitora as alterações no repositório Git.

Atualize sua pilha a partir do seu repositório Git

Para atualizar a pilha, realize alterações diretamente no arquivo de modelo ou no arquivo de implantação da pilha no repositório Git. Após você confirmar suas alterações na ramificação monitorada, o CloudFormation atualiza automaticamente a pilha. Se você usa pull requests, uma pull request é criada automaticamente em seu repositório Git antes que a pilha seja atualizada. É necessário mesclar essa solicitação pull em seu repositório Git para atualizar sua pilha.

No console do CloudFormation, é possível selecionar a pilha e escolher a guia Sincronização do Git para visualizar informações sobre o status da pilha e sobre os eventos de sincronização. Para obter mais informações, consulte [Painel de status do Git Sync](#).

Permitir que o CloudFormation publique um resumo das alterações de pilha em solicitações pull

Este tópico mostra como permitir que o CloudFormation publique um resumo das alterações de pilha em solicitações pull no seu repositório Git.

Ao habilitar o recurso de comentários em solicitações pull, você permite que o CloudFormation publique um comentário descrevendo as diferenças entre a configuração atual da pilha e as alterações propostas quando os arquivos do repositório são atualizados. Esse comentário fornece um resumo dos recursos que serão adicionados, modificados ou excluídos, permitindo que você faça uma revisão completa do código antes de mesclar a solicitação pull.

Para habilitar comentários em solicitações pull para uma nova pilha (console)

Ao criar a pilha, na página Especificar detalhes da pilha, em Repositório de definição de modelos, verifique se a opção Habilitar comentário na solicitação pull está ativada. Essa é a configuração padrão para novas pilhas.

Para habilitar comentários em solicitações pull para uma pilha existente (console)

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

2. Na barra de navegação na parte superior da tela, escolha a Região da AWS em que sua pilha foi criada.
3. Na página Pilhas, escolha a pilha em execução que você deseja atualizar.
4. Selecione a guia Sincronizar Git e, em seguida, escolha Editar.
5. Na página Editar configurações de sincronização do Git, em Repositório de definição de modelos, ative a opção Habilitar comentário na solicitação pull.
6. Selecione Update configuration (Atualizar configuração).

Painel de status do Git Sync

Para visualizar o status de uma implantação do AWS CloudFormation Git Sync, selecione a pilha no console do CloudFormation e escolha a guia Git Sync.

A guia Git Sync se divide em dois painéis: Status do Git Sync e Eventos de sincronização mais recentes.

Status do Git Sync

O painel superior fornece as informações apresentadas a seguir sobre a configuração do Git Sync para a pilha.

Repositório

Um link para o repositório conectado ao Git Sync

Provedor do repositório

O nome do provedor do repositório

Ramificação

O nome da ramificação que o Git Sync está monitorando

Caminho do arquivo de implantação

O caminho completo para o arquivo de implantação da pilha para esta pilha

Status de sincronização do repositório

O status da operação de sincronização mais recente

Mensagem de status de sincronização do repositório

A mensagem da operação de sincronização mais recente

Status do Git Sync

O status do Git Sync para esta pilha

Status do provisionamento

O status da operação de provisionamento

No canto superior direito do painel, use os seguintes botões para modificar ou atualizar o Git Sync:

- Editar: edite a configuração do Git Sync.
- Tentar novamente a confirmação mais recente: atualize a pilha de acordo com a confirmação mais recente para o repositório.
- Desconectar: desconecte o Git Sync da pilha.
- Atualizar: atualize o painel de status do Git Sync.

Eventos de sincronização mais recentes

O painel Eventos de sincronização mais recentes exibe uma tabela das confirmações que foram aplicadas à pilha.

É possível classificar a tabela usando as setas no cabeçalho de cada coluna. A tabela pode ser classificada em ordem crescente ou decrescente de acordo com o seguinte:

- Data
- ID de confirmação
- Evento
- Data
- Tipo de evento

Estados de pilhas com suporte

O Git Sync só poderá ser configurado para uma pilha se ela estiver em um dos seguintes estados com suporte:

- CREATE_COMPLETE
- UPDATE_COMPLETE
- UPDATE_ROLLBACK_COMPLETE
- IMPORT_COMPLETE
- IMPORT_ROLLBACK_COMPLETE

A seguinte tabela contém uma lista completa dos códigos de status da pilha com descrições:

Status da pilha e status detalhado opcional	Descrição
CREATE_COMPLETE	Criação bem-sucedida de uma ou mais pilhas.
CREATE_IN_PROGRESS	Criação contínua de uma ou mais pilhas.
CREATE_FAILED	Criação malsucedida de uma ou mais pilhas. Visualize os eventos da pilha para ver as mensagens de erro associadas. As possíveis razões para uma falha na criação incluem permissões insuficientes para trabalhar com todos os recursos na pilha, valores de parâmetro rejeitados por um produto da AWS ou um tempo limite durante a criação de recursos.
DELETE_COMPLETE	Exclusão bem-sucedida de uma ou mais pilhas. As pilhas excluídas serão retidas e poderão ser visualizadas por 90 dias.
DELETE_FAILED	Exclusão malsucedida de uma ou mais pilhas. Como a exclusão falhou, alguns recursos ainda poderão estar em execução. No entanto, não é possível trabalhar com a pilha, tampouco atualizá-la. Exclua a pilha novamente ou visualize os eventos de pilha para ver as mensagens de erro associadas.
DELETE_IN_PROGRESS	Remoção contínua de uma ou mais pilhas.
REVIEW_IN_PROGRESS	Criação contínua de uma ou mais pilhas com um StackId esperado, mas sem modelos ou recursos.

Status da pilha e status detalhado opcional	Descrição
	 Important Uma pilha com este código de status conta contra o número máximo possível de pilhas.
ROLLBACK_COMPLETE	Remoção bem-sucedida de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha. A pilha retorna ao estado operacional anterior. Quaisquer recursos criados durante a ação de criação da pilha serão excluídos. Esse status ocorre somente após uma falha de criação de uma pilha. Ele significa que todas as operações da pilha parcialmente criada foram adequadamente limpas. Ao entrar nesse estado, a única operação possível é a exclusão.
ROLLBACK_FAILED	Remoção malsucedida de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha. Exclua a pilha ou visualize os eventos de pilha para ver as mensagens de erro associadas.
ROLLBACK_IN_PROGRESS	Remoção contínua de uma ou mais pilhas após uma falha na criação de uma pilha ou após um cancelamento explícito da criação de uma pilha.
UPDATE_COMPLETE	Atualização bem-sucedida de uma ou mais pilhas.

Status da pilha e status detalhado opcional	Descrição
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Remoção contínua de recursos antigos para uma ou mais pilhas após uma atualização bem-sucedida da pilha. Para atualizações de pilha que exigem a substituição de recursos, o CloudFormation cria os novos recursos primeiro e, em seguida, exclui os recursos antigos para ajudar a reduzir as interrupções com a pilha. Nesse estado, a pilha foi atualizado e é utilizável, mas o CloudFormation ainda está excluindo os recursos antigos.
UPDATE_FAILED	Atualização bem-sucedida de uma ou mais pilhas. Visualize os eventos da pilha para ver as mensagens de erro associadas.
UPDATE_IN_PROGRESS	Atualização contínua de uma ou mais pilhas.
UPDATE_ROLLBACK_COMPLETE	Retorno bem-sucedido de uma ou mais pilhas para um estado de trabalho anterior após uma falha na atualização da pilha.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Remoção contínua de recursos novos para uma ou mais pilhas após uma atualização falha da pilha. Nesse estado, a pilha foi revertida para seu estado de trabalho anterior e é utilizável, mas o CloudFormation ainda está excluindo os recursos novos criados durante a atualização da pilha.
UPDATE_ROLLBACK_FAILED	Retorno malsucedido de uma ou mais pilhas para um estado de trabalho anterior após uma falha na atualização da pilha. Quando estiver neste estado, você pode excluir a pilha ou continuar a reversão . Talvez você precise corrigir erros para que a pilha possa retornar a um estado operacional. Se preferir, entre em contato com o Suporte para restaurar a pilha para um estado utilizável.
UPDATE_ROLLBACK_IN_PROGRESS	Retorno contínuo de uma ou mais pilhas para o estado de trabalho anterior após falha na atualização da pilha.

Status da pilha e status detalhado opcional	Descrição
IMPORT_IN_PROGRESS	A operação de importação está em andamento.
IMPORT_COMPLETE	A operação de importação foi concluída com êxito para todos os recursos da pilha que oferecem suporte a <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	A importação será revertida para a configuração do modelo anterior.
IMPORT_ROLLBACK_FAILED	Houve uma falha na operação de reversão de importação para pelo menos um recurso na pilha. Os resultados serão disponibilizados para os recursos importados com êxito pelo CloudFormation.
IMPORT_ROLLBACK_COMPLETE	A importação foi revertida com êxito para a configuração de modelo anterior.

Gerenciar extensões com o registro do CloudFormation

O registro do CloudFormation serve como um hub centralizado para gerenciar extensões que podem ser integradas aos modelos do CloudFormation em sua Conta da AWS. As extensões incluem tipos de recursos, módulos e hooks da AWS e de publicadores externos, além de suas próprias extensões personalizadas. O registro facilita a descoberta e o provisionamento de extensões nos modelos do CloudFormation da mesma maneira que você usa recursos fornecidos pela AWS.

Esta seção descreve como usar o registro do CloudFormation para ativar extensões de terceiros em sua conta, incluindo:

- Ativar extensões públicas
- Registrar e ativar extensões privadas

Tópicos

- [Documentação relacionada](#)
- [Conceitos do registro do CloudFormation](#)
- [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#)
- [Usar extensões públicas de terceiros do registro do CloudFormation](#)
- [Use extensões privadas de terceiros que foram compartilhadas com você](#)
- [Editar dados de configuração para extensões em sua conta](#)
- [Registrar tipos de recursos em AWS Config](#)

Documentação relacionada

Se você for um desenvolvedor interessado em criar suas próprias extensões, consulte a seguinte documentação:

- [Desenvolver módulos usando a CLI do CloudFormation](#) no Guia do usuário da interface de linha de comando do CloudFormation
- [Criar tipos de recursos usando a CLI do CloudFormation](#) no Guia do usuário da interface de linha de comando do CloudFormation
- [Developing custom Hooks using the CloudFormation CLI](#) no CloudFormation Hooks User Guide

Além disso, todos os tipos de recursos provisionáveis da AWS disponíveis no registro do CloudFormation podem ser usados com o AWS API Cloud Control, sendo seus atributos e propriedades definidos em um esquema JSON padrão. Para obter mais informações consulte o [Cloud Control API User Guide](#). Ao usar a Cloud Control API para realizar as operações CRUDL (criar, ler, atualizar, excluir, listar) em recursos da AWS, você pode fazer isso apenas nos recursos da AWS de sua própria Conta da AWS.

Conceitos do registro do CloudFormation

Este tópico explica os principais conceitos que ajudarão você a entender e a começar a usar o registro do CloudFormation.

Tipos de extensão

O registro do CloudFormation oferece os seguintes tipos de extensão:

Hooks

Hooks são verificações de validação que inspecionam as pilhas ou recursos específicos antes que sejam criados, atualizados ou excluídos pelo CloudFormation. Além disso, os hooks podem ser invocados durante as operações de criação de conjuntos de alterações. Eles fornecem um mecanismo para impor os padrões organizacionais e as práticas recomendadas por meio da validação das configurações de recursos em relação a requisitos específicos. Se um hook detectar alguma configuração que não siga a sua lógica, ele poderá emitir um aviso ou gerar falha no processo de provisionamento para evitar a implantação de recursos não compatíveis. Para obter mais informações, consulte o [CloudFormation Hooks User Guide](#).

Para obter documentação sobre como configurar os Hooks usando o console do CloudFormation, consulte as seguintes seções do Guia do usuário de Hooks do CloudFormation:

- [Controles proativos como Hooks do AWS Control Tower](#)
- [Guard Hooks](#)
- [Lambda Hooks](#)

Modules

Módulos são configurações de recursos reutilizáveis que podem ser incluídas em vários modelos de pilha do CloudFormation. Elas simplificam a criação e a manutenção dos modelos do CloudFormation, encapsulando as configurações de recurso complexas ou usadas com

frequência em componentes reutilizáveis. Isso promove consistência e padronização ao longo das implantações de infraestrutura da sua organização.

Resource types

Tipos de recursos permitem modelar e automatizar recursos de terceiros ou recursos personalizados que não são compatíveis nativamente com o CloudFormation. Ao desenvolver tipos de recursos, você pode estender os recursos do CloudFormation para provisionar e gerenciar recursos de vários serviços de terceiros.

Extensões públicas

Extensões públicas são extensões do CloudFormation publicadas no registro para uso de todas as Contas da AWS. Isso inclui:

- Extensões da AWS: as extensões publicadas pela AWS são sempre públicas e ativadas por padrão. Assim sendo, você não precisa tomar nenhuma medida antes de utilizá-las em sua conta. A AWS controla o versionamento dessas extensões para que você esteja sempre usando a versão mais recente disponível.
- Extensões de terceiros: essas extensões são disponibilizadas para uso geral por outras editoras, não pela AWS. Para usar uma extensão pública de terceiros, você deve primeiro ativá-la em sua conta e região.

Extensões ativadas

O registro do CloudFormation em sua Conta da AWS específica inclui três tipos de extensões ativadas:

- Extensões da AWS: todas as extensões públicas da AWS são ativadas automaticamente.
- Extensões de terceiros ativadas: são cópias locais das extensões públicas de terceiros que você ativou explicitamente para sua conta e região. Quando você ativa uma extensão pública de terceiros, o CloudFormation cria uma cópia local dela no registro da sua conta. Para obter mais informações, consulte [Usar extensões públicas de terceiros do registro do CloudFormation](#).
- Registradas privadamente: são as extensões privadas que não são listadas no registro público do CloudFormation. Elas podem ser extensões que você mesmo criou ou que foram compartilhadas com você por sua organização ou por terceiros. Para utilizar uma extensão privada na sua conta, primeiro é preciso registrá-la. Registrar a extensão carrega e ativa uma cópia dela no registro do

CloudFormation da sua conta. Para obter mais informações, consulte [Use extensões privadas de terceiros que foram compartilhadas com você](#).

Usar extensões registradas privadamente e extensões públicas ativadas de outros publicadores em sua conta é como usá-las em um ambiente de sandbox. As extensões são gerenciadas com controle de versão, assim, seu comportamento de provisionamento está vinculado a uma versão específica. Como resultado, essas extensões funcionam como as extensões públicas, seguindo as mesmas regras específicas de versão.

Note

As extensões registradas privadamente e as extensões públicas ativadas de terceiros podem implementar manipuladores de eventos que são executados durante as operações de criação, leitura, atualização, listagem e exclusão. O uso dessas extensões nas pilhas do CloudFormation pode gerar cobranças na sua conta, além das cobranças pelos recursos criados. Para obter mais informações, consulte [Preços do CloudFormation](#).

Visualizar as extensões disponíveis e ativadas no registro do CloudFormation

Para visualizar as extensões disponíveis e ativadas no registro do CloudFormation, é possível usar o Console de gerenciamento da AWS ou a AWS CLI.

Para visualizar as extensões disponíveis e ativadas (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação, em Registro, selecione a categoria da extensão que você deseja visualizar:
 - Public extensions (Extensões públicas) exibe as extensões públicas disponíveis na sua conta.
 1. Em Filtro, Tipo de extensão, escolha seu tipo de extensão: Tipos de recursos, Módulos ou Hooks.
 2. Em Filtro, Publicador, escolha um publicador: AWS ou Terceiro.

- Activated extensions (Extensões ativadas) exibe extensões públicas e privadas ativadas na sua conta.
1. Escolha o tipo de extensão: Resource types (Tipos de recursos) ,Modules (Módulos) ou Hooks (Ganchos).
 2. Use o menu suspenso Filtrar para selecionar mais extensões para visualizar:
 - AWS – lista extensões publicadas pela AWS. Extensões publicadas pela AWS são ativadas por padrão.
 - Third-party (Terceiros): lista todas as extensões públicas de editores diferentes da AWS que você ativou nesta conta.
 - Registered (Registradas): lista todas as extensões privadas que você ativou nesta conta.
 - Publisher (Editor): exibe todas as extensões públicas que você publicou utilizando essa conta.
 4. Pesquise ou escolha o nome da extensão para visualizar seus detalhes.

Para visualizar as extensões disponíveis e ativadas (AWS CLI)

Use o comando [list-types](#).

Usar extensões públicas de terceiros do registro do CloudFormation

Para utilizar uma extensão pública de terceiros no seu modelo, primeiro é necessário ativar a extensão para a conta e a região nas quais você deseja utilizá-la. A ativação de uma extensão a torna utilizável em operações de pilha na conta e região em que ela está ativada.

Quando você ativa uma extensão pública de terceiros, o CloudFormation cria uma entrada no registro de extensões da sua conta para a extensão ativada como uma extensão privada. Isso permite definir as propriedades de configuração incluídas na extensão. As propriedades de configuração definem como essa extensão é configurada para uma determinada região e Conta da AWS.

Além de definir as propriedades de configuração, também é possível personalizar a extensão das seguintes formas:

- Especifique a função de execução utilizada pelo CloudFormation para ativar a extensão, além de configurar o registro em log para essa extensão.

- Especifique se a extensão será atualizada automaticamente quando uma nova versão secundária ou uma versão de patch estiver disponível.
- Especifique um alias a ser utilizado em vez do nome da extensão pública de terceiros. Isso pode ajudar a evitar colisões de nomes entre extensões de terceiros.

Tópicos

- [Configure um perfil de execução com permissões do IAM e uma política de confiança para acesso público à extensão](#)
- [Usar novas versões de extensões automaticamente](#)
- [Usar aliases para fazer referência a extensões](#)
- [Comandos da AWS CLI comumente usados para trabalhar com extensões públicas](#)
- [Ativar uma extensão pública de terceiros em sua conta](#)
- [Atualizar uma extensão pública de terceiros em sua conta](#)
- [Desativar extensões públicas de terceiros em sua conta](#)

Configure um perfil de execução com permissões do IAM e uma política de confiança para acesso público à extensão

Ao ativar uma extensão pública via registro do CloudFormation, você pode fornecer um perfil de execução que conceda ao CloudFormation as permissões necessárias para invocar essa extensão em sua região e Conta da AWS.

As permissões necessárias para o perfil de execução são definidas na seção de manipulador do esquema da extensão. É necessário criar uma política do IAM que conceda as permissões específicas necessárias para a extensão e anexá-la à função de execução.

Além da política de permissões, o perfil de execução também deve ter uma política de confiança que permita ao CloudFormation assumir o perfil. Siga as orientações em [Criar um perfil usando políticas de confiança personalizadas](#) no Guia do usuário do IAM para criar um perfil com uma política de confiança personalizada.

Relação de confiança

A seguir são mostrados exemplos de políticas que você pode usar.

Opcionalmente, é possível restringir o escopo da permissão da prevenção do problema "confused deputy" entre serviços usando uma ou mais chaves de contexto de condição global com o campo `Condition`. Para obter mais informações, consulte [Prevenção contra o ataque do "substituto confuso" em todos os serviços](#).

- Defina o valor `aws:SourceAccount` para o ID da sua conta.
- Defina o valor `aws:SourceArn` para o ARN da sua extensão.

Exemplo de política de confiança 1

Veja a seguir um exemplo de política de confiança de perfil do IAM para uma extensão de tipo de recurso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resources.cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource"
        }
      }
    }
  ]
}
```

Exemplo de política de confiança 2

Veja a seguir um exemplo de política de confiança de perfil do IAM para a extensão de hook.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com",
          "hooks.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-west-2:123456789012:type/hook/Organization-Service-Hook"
        }
      }
    }
  ]
}
```

Usar novas versões de extensões automaticamente

Ao ativar uma extensão, você também pode especificar o tipo de extensão para utilizar a versão secundária mais recente. Seu tipo de extensão atualizará a versão secundária sempre que o editor lançar uma nova versão na sua extensão ativada.

Por exemplo, da próxima vez em que você executar uma operação de pilha, como criar ou atualizar uma pilha, utilizando um modelo que inclui essa extensão, o CloudFormation usará a nova versão secundária.

Atualizar para uma nova versão de extensão, automática ou manualmente, não afeta nenhuma instância de extensão já provisionada em pilhas.

O CloudFormation trata as atualizações de versões principais de extensões como contendo potencialmente alterações significativas e, portanto, exige que você atualize manualmente para uma nova versão principal de uma extensão.

As extensões publicadas pela AWS são ativadas por padrão para todas as contas e regiões em que estão disponíveis e sempre utilizam a versão mais recente disponível em cada Região da AWS.

Important

Como você controla se e quando as extensões são atualizadas para a versão mais recente da sua conta, é possível que versões diferentes da mesma extensão sejam implantadas em contas e regiões diferentes.

Isso pode gerar resultados inesperados ao utilizar o mesmo modelo, contendo essa extensão, entre essas contas e regiões.

Usar aliases para fazer referência a extensões

Não é possível ativar mais de uma extensão com um determinado nome em uma determinada região e Conta da AWS. Como diferentes editores podem oferecer extensões públicas com o mesmo nome, o CloudFormation permite que você especifique um alias para qualquer extensão pública de terceiros que você ativar.

Se você especificar um alias para a extensão, o CloudFormation tratará esse alias como o nome do tipo de extensão dentro da conta e da região. Você deve utilizar o alias para fazer referência à extensão nos seus modelos, chamadas de API e no console do CloudFormation.

Aliases de extensão devem ser exclusivos em uma conta e região especificadas. É possível ativar o mesmo recurso público várias vezes na mesma conta e região utilizando apelidos de nomes de tipos diferentes.

Important

Embora aliases de extensões só precisem ser exclusivos em uma determinada conta e região, sugerimos que os usuários não atribuam o mesmo alias a diferentes extensões públicas de terceiros entre contas e regiões. Isso pode gerar resultados inesperados ao utilizar um modelo que contém o alias de extensão em várias contas ou regiões.

Comandos da AWS CLI comumente usados para trabalhar com extensões públicas

Os comandos comumente usados para trabalhar com extensões públicas incluem:

- [activate-type](#) para ativar um módulo ou tipo de recurso público de terceiros em sua conta.
- [set-type-configuration](#) para especificar os dados de configuração para uma extensão na sua conta e para desabilitar e habilitar hooks.
- [list-types](#) para listar as extensões na sua conta.
- [describe-type](#) para devolver informações detalhadas sobre uma extensão específica ou uma versão específica da extensão, incluindo dados de configuração atuais.
- [set-type-default-version](#) para especificar qual versão de uma extensão é a versão padrão.
- [deactivate-type](#) para desativar um módulo ou tipo de recurso público de terceiros que havia sido ativado em sua conta.

Ativar uma extensão pública de terceiros em sua conta

O tópico a seguir mostra como ativar uma extensão pública de terceiros em sua conta, o que a torna utilizável na conta e na região em que foi ativada.

Note

Antes de continuar, confirme se você criou o [perfil do IAM](#) que utilizará com essa extensão.

Tópicos

- [Ativar uma extensão pública \(console\)](#)
- [Ativar uma extensão pública \(AWS CLI\)](#)

Ativar uma extensão pública (console)

Siga as etapas nesta seção para usar o console para:

- Ativar uma extensão pública de terceiros
- Especificar dados de configuração de extensão adicionais para sua conta

Para ativar uma extensão pública para uso em sua conta

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação, em Registro, selecione Extensões públicas.
4. Use a opção Filtrar para selecionar o tipo de extensão e selecione Terceiros. (Extensões publicadas pela AWS são ativadas por padrão.)
5. Selecione a extensão e depois Ativar.

Se várias versões de uma extensão estiverem disponíveis, você poderá usar o menu Versão para selecionar a versão da extensão que deseja ativar. A versão padrão é a mais atual.

6. Para Nome da extensão, você pode manter a opção Usar padrão selecionada ou escolher Substituir padrão e inserir o alias do tipo de extensão que deseja usar com essa extensão. O alias deve seguir o formato recomendado para o tipo de extensão. Para obter mais informações, consulte [Usar aliases para fazer referência a extensões](#).
7. Se a extensão que você está ativando for um hook ou um tipo de recurso, para ARN do perfil de execução, especifique o perfil do IAM que o CloudFormation deve assumir ao invocar a extensão. Para obter mais informações, consulte [Configure um perfil de execução com permissões do IAM e uma política de confiança para acesso público à extensão](#).
8. Em Configuração do registro em log, se desejar, especifique as informações de configuração de registro em log para uma extensão. Por exemplo:

```
{
  "logRoleArn": "arn:aws:iam::account:role/rolename",
  "logGroupName": "log-group-name"
}
```

As informações de configuração de registro em log não são necessárias, mas são recomendadas para fins de depuração. Para utilizar a configuração de registro em log com hooks, adicione a mesma política de confiança que a função de execução especificada, para que a função de registro em log possa gravar logs no seu grupo de logs.

Os nomes de chaves `logRoleArn` e `logGroupName` diferenciam maiúsculas de minúsculas.

9. Em Versionamento, Atualizações automáticas, escolha como as atualizações serão recebidas.

- On (Ativado): atualiza automaticamente para a versão secundária mais recente. As versões principais são atualizadas manualmente.
- Off (Desativado): nunca atualiza automaticamente para a versão mais recente. Todas as versões são atualizadas manualmente.

Para obter mais informações, consulte [Usar novas versões de extensões automaticamente](#).

Se a extensão exigir configuração adicional, você terá a opção de especificar os dados de configuração agora ou depois que a extensão tiver sido ativada.

Important

Se a extensão que você está ativando for um hook, esta etapa será necessária. Você deve especificar ENABLED para a propriedade HookInvocationStatus. Essa operação ativa as propriedades do hook que são definidas na seção `properties` do esquema do hook. Para obter mais informações, consulte [Hook configuration schema syntax reference](#) no CloudFormation Hooks User Guide.

Para especificar os dados de configuração.

1. Em Configuração, escolha Configurar agora e, em seguida, escolha Ativar extensão.

O CloudFormation exibe a página Configure extension (Configurar extensão). Para visualizar o esquema de configuração atual da extensão, certifique-se de a opção View configuration schema (Visualizar esquema de configuração) esteja ativada.

2. Na caixa de texto Configuration JSON (Configuração JSON), insira uma string JSON que represente os dados de configuração que você deseja especificar para essa extensão. O JSON especificado deve ser válido em relação ao esquema de configuração da extensão.
3. Escolha Configurar extensão.

Se preferir configurar a extensão após a ativação, pule essa etapa e forneça os dados de configuração posteriormente.

1. Em Configuração, escolha Configurar mais tarde e, em seguida, escolha Ativar extensão.

2. Depois que a extensão for ativada, você poderá configurá-la navegando até a extensão na página de extensões ativadas e fornecendo os dados de configuração.

Ativar uma extensão pública (AWS CLI)

Siga as etapas nesta seção para usar o AWS CLI para:

- Ativar uma extensão pública de terceiros
- Especificar dados de configuração de extensão adicionais para sua conta

Ativar hooks públicos

Ao ativar o Hooks na conta, você autoriza um hook a usar permissões definidas em sua Conta da AWS. O CloudFormation remove as permissões não necessárias antes de passar as permissões ao hook. O CloudFormation recomenda que os clientes ou usuários do hook revisem as permissões dele e estejam cientes sobre elas antes de ativar o Hooks na conta.

Para ativar um hook público para uso em sua conta (AWS CLI)

1. Obtenha o ARN para do hook e salve-o. Você pode obter o ARN de um Hook usando o Console de gerenciamento da AWS ou a AWS CLI. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Use o comando [set-type-configuration](#) para especificar os dados de configuração. O JSON transmitido por você para `--configuration` deverá ser válido em relação ao esquema de configuração do hook. Para ativar o hook para todas as operações de pilha, é necessário definir a propriedade `HookInvocationStatus` como `ENABLED` na seção `HookConfiguration`.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Para obter mais informações sobre as opções de configuração de `HookConfiguration`, consulte [Hook configuration schema syntax reference](#) no CloudFormation Hooks User Guide.

Ativar módulos e tipos de recursos públicos

Para ativar uma extensão pública para uso em sua conta (AWS CLI)

- Use [activate-type](#) para ativar a extensão e especificar se ela deve ser atualizada automaticamente sempre que uma nova versão secundária for publicada.

O exemplo abaixo especifica o nome do recurso da Amazon (ARN) público de uma extensão pública a ser ativada para essa conta. Além disso, ele especifica que o CloudFormation deve atualizar a extensão sempre que uma nova versão secundária é publicada.

```
aws cloudformation activate-type \  
  --public-type-arn public_extension_ARN \  
  --execution-role-arn arn:aws:iam::123456789012:role/my-execution-role \  
  --auto-update true --region us-west-2
```

Esse comando retorna um ARN da extensão ativada.

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-  
Resource-Example"  
}
```

Atualizar a versão de uma extensão pública utilizando (AWS CLI)

Use [activate-type](#) para ativar a extensão novamente.

Use a opção `--version-bump` para especificar se a extensão deve ser atualizada para a versão MAJOR mais recente ou para a versão MINOR mais recente.

```
aws cloudformation activate-type --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0 \  
  --type-name-alias Example::Test::Alias \  
  --version-bump MAJOR --region us-west-2
```

Atualizar uma extensão pública de terceiros em sua conta

Após ativar uma extensão pública de terceiros, você poderá atualizar a maioria dos detalhes da extensão em sua conta.

Para atualizar uma extensão pública em sua conta (console)

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação, em Registro, escolha Extensões ativadas.
4. Localize e selecione a extensão que você deseja atualizar. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).
5. No menu Ações, escolha Editar e, em seguida, escolha a opção de edição apropriada:
 - Para atualizar o esquema de configuração, consulte [Editar dados de configuração para extensões em sua conta](#).
 - Para ativar ou desativar atualizações automáticas:
 1. Escolha Editar atualizações automáticas.
 2. Escolha Ativar ou Desativar e, em seguida, escolha Salvar. Para obter mais informações, consulte [Usar novas versões de extensões automaticamente](#).
 - Para atualizar o perfil de execução:
 1. Escolha Editar perfil de execução.
 2. Especifique o ARN do perfil do IAM que você deseja que o CloudFormation utilize ao invocar essa extensão e, em seguida, escolha Salvar. Para obter mais informações, consulte [Configure um perfil de execução com permissões do IAM e uma política de confiança para acesso público à extensão](#).
 - Para atualizar a configuração de registro em log:
 1. Selecione Editar configuração do registro em log.
 2. Edite o JSON de configuração do registro em log e escolha Salvar.

Desativar extensões públicas de terceiros em sua conta

Quando você não precisar mais de uma extensão pública de terceiros ativada, use os procedimentos a seguir para desativá-la em sua conta.

Tópicos

- [Desativar uma extensão pública em sua conta \(console\)](#)
- [Desativar uma extensão pública em sua conta \(AWS CLI\)](#)

- [Desabilitar um hook de sua conta \(AWS CLI\)](#)

Desativar uma extensão pública em sua conta (console)

Para desativar uma extensão pública em sua conta

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação, em Registro, escolha Extensões ativadas.
4. Localize e selecione a extensão que você deseja desativar. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).
5. No menu Ações, escolha Desativar.
6. Selecione Desativar.

Desativar uma extensão pública em sua conta (AWS CLI)

Use o seguinte comando [deactivate-type](#):

```
aws cloudformation deactivate-type --type MODULE \  
  --type-name Example::Test::Type::MODULE \  
  --region us-west-2
```

Desabilitar um hook de sua conta (AWS CLI)

Desabilitar um hook evita que ele seja executado em sua Conta da AWS sem removê-lo.

Use o comando [set-type-configuration](#) e especifique HookInvocationStatus como DISABLED para desabilitar um hook.

O exemplo a seguir especifica a Região da AWS e o nome do recurso da Amazon (ARN) do hook que está sendo desabilitado.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "DISABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn "arn:aws:cloudformation:us-west-2:123456789012:type/hook/MyTestHook" --  
  region us-west-2
```

Para obter mais informações, consulte [Disable and enable CloudFormation Hooks](#) no CloudFormation Hooks User Guide.

Use extensões privadas de terceiros que foram compartilhadas com você

Para usar extensões privadas de terceiros que foram compartilhadas com você, primeiro você deverá registrá-las no CloudFormation, nas contas e regiões em que deseja usá-las. O registro da extensão carrega uma cópia dela no registro do CloudFormation na sua conta e a ativa. Ao registrar uma extensão privada, ela aparecerá no registro do CloudFormation para a respectiva região e Conta da AWS, e você poderá usá-la em seus modelos de pilha.

Tópicos

- [Permissões do IAM para registrar uma extensão privada de terceiros](#)
- [Comandos da AWS CLI comumente usados para trabalhar com extensões privadas](#)
- [Registrar uma extensão privada de terceiros em sua conta](#)
- [Remover extensões privadas de terceiros de sua conta](#)

Permissões do IAM para registrar uma extensão privada de terceiros

Como parte do registro de uma extensão privada, é possível especificar um bucket do Amazon S3 que contém o pacote do projeto de extensão. Esse pacote contém todos os arquivos de origem necessários para a extensão que você quer registrar. O usuário que registra a extensão deve ser capaz de acessar o pacote de projeto no bucket do Amazon S3. Para isso, o usuário precisa ter permissões [GetObject](#) para o pacote de extensão.

Isso será verdadeiro se você estiver usando o comando [register-type](#) da AWS CLI ou o comando [submit](#) da CLI do CloudFormation.

Para obter mais informações, consulte [Actions, Resources, and Condition Keys for Amazon S3](#) na Service Authorization Reference.

Comandos da AWS CLI comumente usados para trabalhar com extensões privadas

Os comandos comumente usados para trabalhar com extensões privadas incluem:

- [register-type](#) para registrar uma extensão privada em sua conta.
- [describe-type-registration](#) para retornar o status atual de uma solicitação de registro.
- [list-types](#) para listar as extensões na sua conta.
- [describe-type](#) para devolver informações detalhadas sobre uma extensão específica ou uma versão específica da extensão, incluindo dados de configuração atuais.
- [set-type-configuration](#) para especificar os dados de configuração para uma extensão na sua conta e para desabilitar e habilitar hooks.
- [set-type-default-version](#) para especificar qual versão de uma extensão é a versão padrão.
- [deregister-type](#) para remover uma extensão privada ou uma versão de extensão da sua conta.

Registrar uma extensão privada de terceiros em sua conta

Este tópico cobre as etapas para registrar uma extensão privada de terceiros que é compartilhada com você para que ela fique disponível para ser usada em sua conta.

Note

Antes de continuar, confirme se você tem as [permissões do IAM](#) necessárias para registrar uma extensão privada.

Para registrar uma extensão privada que é compartilhada com você (AWS CLI)

1. Localize o bucket do Amazon S3 que contém o pacote de projeto da extensão privada que você deseja registrar na conta.
2. Use o comando [register-type](#) para registrar uma extensão privada em sua conta.

Por exemplo, o comando a seguir registra o tipo de recurso `My::Resource::Example` na Conta da AWS especificada.

```
aws cloudformation register-type --type RESOURCE \  
  --type-name My::Resource::Example \  
  --schema-handler-package [s3 object path] --region us-west-2
```

`RegisterType` é uma operação assíncrona e retorna um token de registro que você pode usar para rastrear o progresso da solicitação de registro.

```
{
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"
}
```

Se a extensão chamar as APIs da AWS como parte de sua funcionalidade, será necessário criar um perfil de execução do IAM que inclua as permissões necessárias para chamar essas APIs da AWS e provisionar esse perfil de execução na sua conta. Em seguida, você poderá especificar esse perfil de execução usando a opção `--execution-role-arn`. Depois, o CloudFormation assumirá essa função de execução para fornecer as credenciais apropriadas para o tipo de recurso.

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (Opcional) Use o token de registro com o comando [describe-type-registration](#) para monitorar o andamento da solicitação de registro.

Quando o CloudFormation concluir a solicitação de registro, ele definirá o status do andamento da solicitação como COMPLETE.

O exemplo a seguir usa o token de registro retornado pelo comando `describe-type-registration` acima para retornar as informações do status do registro.

```
aws cloudformation describe-type-registration \
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \
  --region us-west-2
```

O comando retorna a seguinte saída.

```
{
  "ProgressStatus": "COMPLETE",
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example",
  "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example/00000001"
}
```


⚠ Important

Se a extensão sendo registrada for um hook, essa etapa será obrigatória. Você deve especificar `ENABLED` para a propriedade `HookInvocationStatus`. Essa operação ativa as propriedades do hook que são definidas na seção `properties` do esquema do hook. Para obter mais informações, consulte [Hook configuration schema syntax reference](#) no CloudFormation Hooks User Guide.

Para especificar os dados de configuração de um hook (AWS CLI)

1. Obtenha o ARN para o hook e salve-o. Você pode obter o ARN de um Hook usando o Console de gerenciamento da AWS ou a AWS CLI. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Use o comando [set-type-configuration](#) para especificar os dados de configuração. O JSON transmitido por você para `--configuration` deverá ser válido em relação ao esquema de configuração do hook. Para ativar o hook, defina a propriedade `HookInvocationStatus` como `ENABLED` na seção `HookConfiguration`.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Para obter mais informações, consulte [Hook configuration schema syntax reference](#) no CloudFormation Hooks User Guide.

Remover extensões privadas de terceiros de sua conta

Para remover uma extensão privada de terceiros ou uma versão de extensão, use o comando [deregister-type](#).

É possível cancelar o registro de uma versão específica da extensão ou da extensão como um todo. Para cancelar o registro de uma extensão, é necessário cancelar o registro individual de todas as

versões registradas dessa extensão. Se uma extensão tiver apenas uma única versão registrada, o cancelamento do registro dessa versão resultará no cancelamento do registro da extensão em si. Você não pode cancelar o registro da versão padrão de uma extensão, a menos que seja a única versão registrada dessa extensão. Nesse caso, o registro da extensão em si também será cancelado.

Warning

O cancelamento do registro de uma extensão privada não pode ser desfeito. Esta ação vai:

- Tornar a extensão inutilizável em todas as operações do CloudFormation.
- Causar falhas em futuras atualizações de pilha que usem essa extensão (para módulos e tipos de recursos). Embora você possa registrar novamente a extensão de forma privada posteriormente, isso poderá causar falhas se o CloudFormation depender de uma versão anterior.

Antes de continuar, use os comandos [list-stacks](#) e [get-template](#) para verificar se nenhuma pilha ativa está usando essa extensão.

Exemplo de comandos para cancelar o registro da extensão

Esta seção fornece exemplos que mostram as diferentes formas de cancelar o registro de extensões privadas.

Cancelar o registro por nome do tipo

Use o comando [deregister-type](#) com as opções `--type` e `--type-name` para cancelar o registro da sua extensão.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE
```

Cancelar o registro por nome do tipo e versão

Para cancelar o registro de uma versão específica da sua extensão, especifique a opção `--version-id` no comando.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --version-id 00000001
```

 Tip

Para definir uma versão diferente da extensão como padrão primeiro, use o comando [set-type-default-version](#).

Cancelar o registro pelo ARN

Use a opção `--arn` e especifique o ARN da sua extensão para cancelar o registro.

```
aws cloudformation deregister-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource
```

Editar dados de configuração para extensões em sua conta

Este tópico fornece orientação sobre como editar dados de configuração para extensões em sua conta em uma região específica. As extensões podem incluir propriedades de configuração destinadas a serem aplicadas em todas as instâncias da extensão para uma determinada conta e região. Elas são especificadas pelo autor da extensão na definição da configuração da extensão. Se houver alguma propriedade necessária na definição de configuração da extensão, você deverá especificá-la antes de poder utilizar a extensão na sua conta e região.

Para saber mais sobre como as definições de configuração são especificadas ao desenvolver uma extensão, consulte a documentação a seguir.

- [Referência de sintaxe de esquema de configuração de hook](#)
- [Definir a configuração no nível da conta de uma extensão](#)

Tópicos

- [Permissões necessárias para usar referências dinâmicas](#)
- [Editar dados de configuração para uma extensão \(console\)](#)

- [Editar dados de configuração para uma extensão \(AWS CLI\)](#)

Permissões necessárias para usar referências dinâmicas

Se os dados de configuração incluírem referências dinâmicas aos valores armazenados no AWS Systems Manager ou AWS Secrets Manager, qualquer perfil usado para provisionar o tipo (por exemplo, ao criar ou atualizar uma pilha) deverá ter as permissões adequadas para recuperar esse valor. Especificamente:

- Se os dados de configuração contiverem um parâmetro armazenado no AWS Systems Manager Parameter Store, o usuário ou perfil usado para provisionar o tipo deverá ter permissões para chamar [GetParameter](#).
- Se os dados de configuração contiverem um segredo armazenado no AWS Secrets Manager, o usuário ou perfil usado para provisionar o tipo deverá ter permissões para chamar [GetSecretValue](#).

Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

Editar dados de configuração para uma extensão (console)

Siga as etapas nesta seção para usar o console para:

- Visualizar os dados de configuração atuais de uma extensão
- Atualizar os dados de configuração de extensão para sua conta

Para visualizar os dados de configuração atuais de uma extensão

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. No painel de navegação, em Registro, escolha Extensões ativadas.
4. Encontre a extensão que você deseja visualizar. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).
5. Escolha a extensão para visualizar seus detalhes.
6. Na página de detalhes da extensão, selecione a guia Configuration (Configuração).

7. Expanda a guia Configuration schema (Esquema de configuração) para ver o esquema de configuração definido para a extensão.
8. Expanda a guia Configuration (Configuração) para ver a configuração atual que você definiu para essa extensão.

Para atualizar os dados de configuração para uma extensão

1. Na página de detalhes da extensão, na guia Configuração, escolha Editar configuração.

Como alternativa, em Ações, escolha Editar e, em seguida, escolha Editar configuração.

O CloudFormation exibe a página Configure extension (Configurar extensão). Certifique-se de que a opção View configuration schema (Visualizar esquema de configuração) esteja ativada para ver o esquema de definição de configuração atual da extensão.

2. Na caixa de texto Configuration JSON (Configuração JSON), insira uma string JSON que represente o esquema de configuração que você deseja definir para essa extensão. Ele deve ser validado com o esquema definido em Configuration schema (Esquema de configuração).
3. Escolha Configurar extensão.

Editar dados de configuração para uma extensão (AWS CLI)

Siga as etapas nesta seção para usar o AWS CLI para:

- Visualizar os dados de configuração atuais de uma extensão
- Atualizar os dados de configuração de extensão para sua conta

Para visualizar os dados de configuração atuais de uma extensão

- Use o comando [describe-type](#) para retornar informações detalhadas sobre a extensão. O elemento ConfigurationSchema da saída contém a definição de configuração atual da extensão em uma determinada região.

Como alternativa, use o comando [batch-describe-type-configurations](#) para retornar dados de configuração sobre várias extensões.

Para atualizar os dados de configuração para uma extensão

- Use o comando [set-type-configuration](#) para especificar os dados de configuração. O JSON transmitido para `--configuration` deve validar em relação ao esquema de configuração da extensão.

No exemplo a seguir, o comando `set-type-configuration` especifica os dados de configuração de `{"CredentialKey": "testUserCredential"}` para opção `--configuration`.

```
aws cloudformation set-type-configuration --type RESOURCE \  
  --type-name My::Resource::Example \  
  --configuration-alias default \  
  --configuration '{"CredentialKey": "testUserCredential"}' \  
  --region us-west-2
```

Registrar tipos de recursos em AWS Config

Você pode especificar que o AWS Config rastreie automaticamente seus tipos de recursos privados e registre alterações nesses recursos como itens de configuração. Isso permite visualizar o histórico de configuração desses tipos de recursos privados, bem como gravar regras do Regras do AWS Config para verificar as práticas recomendadas de configuração. O AWS Config é necessário para a extensão de hooks.

Para que o AWS Config rastreie automaticamente seus tipos de recursos privados:

- Gerencie os recursos por meio do CloudFormation. Isso inclui a execução de todas as operações de criação, atualização e exclusão de recursos por meio do CloudFormation.

Note

Se você usar uma função do IAM para executar suas operações de pilha, ela deverá ter permissão para chamar as seguintes ações do AWS Config:

- [PutResourceConfig](#)
- [DeleteResourceConfig](#)

- Configure o AWS Config para registrar todos os tipos de recursos. Para obter mais informações, consulte [Registrar configurações para recursos de terceiros usando a AWS CLI](#) no Guia do desenvolvedor do AWS Config.

 Note

O AWS Config não oferece suporte ao registro de recursos privados que contém propriedades definidas como obrigatórias e somente gravação.

Por design, as propriedades de recurso definidas como somente gravação não são retornadas no esquema usado para criar o item de configuração do AWS Config. Devido a isso, incluir uma propriedade definida como somente gravação e como necessária fará com que a criação do item de configuração falhe, uma vez que uma propriedade necessária não estará presente. Para visualizar o esquema que será usado para criar o item de configuração, você pode revisar a propriedade `schema` da ação [DescribeType](#).

Para mais informações sobre itens de configuração, consulte [Itens de configuração](#) no Guia do desenvolvedor do AWS Config.

Impedir que propriedades confidenciais sejam registradas em um item de configuração

Seu tipo de recurso pode conter propriedades que você considere informações confidenciais, como senhas, segredos ou outros dados confidenciais, e que não queira registrar como parte do item de configuração. Para impedir que uma propriedade seja registrada no item de configuração, inclua essa propriedade na lista `writeOnlyProperties` no esquema de tipo de recurso. Propriedades de recurso listadas como `writeOnlyProperties` podem ser especificadas pelo usuário, mas não serão retornadas por uma solicitação `read` ou `list`.

Para obter mais informações, consulte [writeOnlyProperties](#) no Guia do usuário da CLI do CloudFormation.

Entrega contínua com o CodePipeline

A entrega contínua é uma prática de liberação na qual as alterações no código são automaticamente compiladas, testadas e preparadas para liberação para produção. Com o CloudFormation e o CodePipeline, você pode usar a entrega contínua para compilar e testar automaticamente as alterações feitas nos modelos do CloudFormation antes de promovê-las para as pilhas de produção. Esse processo de liberação permite que você faça alterações de forma rápida e confiável em sua infraestrutura da AWS.

Por exemplo, é possível criar um fluxo de trabalho que compila automaticamente uma pilha de teste quando você envia um modelo atualizado para um repositório de código. Depois que o CloudFormation compila a pilha de teste, você pode testá-la e decidir se deve enviar as alterações por push para uma pilha de produção. Para obter mais informações sobre os benefícios da entrega contínua, consulte [O que é entrega contínua?](#).

Use o CodePipeline para criar um fluxo de trabalho de entrega contínua montando um pipeline para as pilhas do CloudFormation. O CodePipeline é integrado internamente ao CloudFormation, assim sendo, você pode especificar ações específicas do CloudFormation, como criar, atualizar ou excluir uma pilha, em um pipeline. Para obter mais informações sobre CodePipeline, consulte o [Guia do usuário do AWS CodePipeline](#).

Tópicos

- [Demonstração: Criar um pipeline para pilhas de produção e teste](#)
- [CloudFormationReferência de propriedades de configuração DO](#)
- [CloudFormationArtefatos do](#)
- [Usar funções de substituição de parâmetros com pipelines CodePipeline](#)

Demonstração: Criar um pipeline para pilhas de produção e teste

Imagine um processo de liberação no qual você envia um modelo do CloudFormation que o CloudFormation depois usa para criar automaticamente uma pilha de teste. Depois de verificar a pilha de teste, você pode visualizar como suas alterações modificarão sua pilha de produção e, em seguida, escolher se deseja implementá-las. Para realizar esse fluxo de trabalho, você pode usar o CloudFormation para criar a pilha de teste, excluir a pilha de teste, criar um conjunto de alterações e depois executar o conjunto de alterações. Porém, a cada ação, você precisa interagir

manualmente com o CloudFormation. Nesta demonstração, vamos criar um pipeline do CodePipeline que automatiza muitas dessas ações, ajudando você a obter um fluxo de trabalho de entrega contínuo com as pilhas do CloudFormation.

Pré-requisitos

Essa demonstração pressupõe que você já tenha usado o CodePipeline e saiba como funcionam os modelos e as pilhas do AWS CloudFormation, e os pipelines. Para obter mais informações sobre CodePipeline, consulte o [Guia do usuário do AWS CodePipeline](#). Você também precisará ter um bucket do Amazon S3 na mesma Região da AWS em que criar o pipeline.

Important

O modelo do WordPress de amostra cria uma instância do EC2 que requer uma conexão com a Internet. Verifique se você tem uma VPC padrão e sub-rede que permitem o tráfego para a Internet.

Visão geral da demonstração

Esta apresentação cria um pipeline para um site do WordPress de amostra em uma pilha. O pipeline é separado em três etapas. Cada etapa deve conter pelo menos uma ação, que é uma tarefa que o pipeline executa em seus artefatos (sua entrada). Uma etapa organiza ações em um pipeline. O CodePipeline deve concluir todas as ações de uma etapa antes que o estágio processe novos artefatos, por exemplo, se você enviou uma nova entrada para executar novamente o pipeline.

Ao final desta apresentação, você terá um pipeline que executa o fluxo de trabalho a seguir:

1. A primeira etapa do pipeline recupera um artefato de origem (um modelo do CloudFormation e seus arquivos de configuração) de um repositório.

Você preparará um artefato que inclui um modelo do WordPress de amostra e faz upload para um bucket do S3.

2. Na segunda etapa, o pipeline cria uma pilha de teste e, em seguida, aguarda a sua aprovação.

Depois de verificar a pilha de teste, você pode optar por continuar com o pipeline original ou criar e enviar outro artefato para fazer alterações. Se você aprovar, esta etapa exclui a pilha de teste e, em seguida, o pipeline continua para a próxima etapa.

3. No terceiro estágio, o pipeline cria um conjunto de alterações em uma pilha de produção e, em seguida, aguarda a sua aprovação.

Em sua execução inicial, você não terá uma pilha de produção. O conjunto de alterações mostra todos os recursos que o CloudFormation criará. Se você aprovar, esta etapa executa o conjunto de alterações e cria sua pilha de produção.

Note

O CloudFormation é um serviço gratuito. No entanto, você é cobrado pelos recursos da AWS, como a instância do EC2, que você inclui a sua pilha na taxa atual para cada um deles. Para obter mais informações sobre o preço da AWS, consulte a página de detalhes de cada produto em <http://aws.amazon.com>.

Etapa 1: Editar o artefato e fazer upload dele para um bucket do S3

Antes de criar o pipeline, você deve configurar seus arquivos e o repositório de origem. O CodePipeline copia esses arquivos de origem para o [armazenamento de artefatos](#) do pipeline e depois os utiliza para executar ações no pipeline, como criar de uma pilha do CloudFormation.

Quando você usa o Amazon Simple Storage Service (Amazon S3) como o repositório de origem, o CodePipeline requer que você compacte os arquivos de origem antes de enviá-los para um bucket do S3. O arquivo compactado é um artefato do CodePipeline que pode conter um modelo do CloudFormation, um arquivo de configuração de modelo ou ambos. Fornecemos um artefato que contém um modelo do WordPress de amostra de dois arquivos de configuração de modelo. Os dois arquivos de configuração especificam valores de parâmetro para o modelo do WordPress. O CodePipeline os utiliza quando cria pilhas do WordPress. Um arquivo contém valores de parâmetro para uma pilha de teste e o outro para uma pilha de produção. Você precisará editar os arquivos de configuração, por exemplo, para especificar um nome do par de chaves do EC2 existente que você possui. Para obter mais informações sobre artefatos, consulte [CloudFormationArtefatos do](#).

Depois de criar o artefato, você fará upload para um bucket do S3.

Como editar e fazer upload do artefato

1. Faça download e abra o artefato de amostra: <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/wordpress-single-instance.zip>.

O artefato contém três arquivos:

- O modelo do WordPress de amostra: `wordpress-single-instance.yaml`
- O arquivo de configuração do modelo para a pilha de teste: `test-stack-configuration.json`
- O arquivo de configuração do modelo da pilha de produção: `prod-stack-configuration.json`

2. Extraia todos os arquivos e, em seguida, use qualquer editor de texto para modificar os arquivos de configuração de modelo.

Abra os arquivos de configuração para verificar se eles contêm pares de chave/valor que mapeiam os parâmetros de modelo do WordPress. Os arquivos de configuração especificam os valores de parâmetro que o pipeline usa ao criar as pilhas de teste e produção.

Edite o arquivo `test-stack-configuration.json` para especificar valores de parâmetro da pilha de teste e o arquivo `prod-stack-configuration.json` da pilha de produção.

- Altere os valores das chaves de ativação `DBPassword` e `DBRootPassword` para as senhas que você pode usar para fazer login em seu banco de dados do WordPress. Conforme definido no modelo do WordPress, os valores de parâmetro devem conter apenas caracteres alfanuméricos.
 - Altere o valor da chave de ativação `KeyName` de um nome do par de chaves do EC2 na região na qual você criará o pipeline.
3. Adicione os arquivos de configuração modificados ao artefato original (`.zip`), substituindo arquivos duplicados.

Você agora tem um artefato personalizado que pode fazer upload para um bucket do S3.

4. [Faça upload do artefato para um bucket do S3 que você possui.](#)

Observe o local do arquivo. Você especificará o local desse arquivo ao criar seu pipeline.

Observações sobre o artefato e o bucket do S3:

- Use um bucket que esteja na mesma região da AWS na qual você criará o pipeline.
- O CodePipeline requer que o bucket esteja [habilitado para versionamento](#).
- Você também pode usar serviços que não exigem a compactação dos seus arquivos antes de fazer upload, como GitHub ou CodeCommit, para seu repositório de origem.

- Os artefatos podem conter informações confidenciais, como senhas. Limite o acesso para que somente os usuários permitidos possam visualizar o arquivo. Quando você fizer isso, certifique-se de que o CodePipeline ainda pode acessar o arquivo. Por exemplo, se você fizer upload do seu artefato para um bucket S3, use [políticas de bucket S3 ou políticas de usuário](#) para restringir o acesso.

Você agora tem um artefato que o CodePipeline pode extrair para o pipeline. Na próxima etapa, você especificará o local do artefato e criar o pipeline do WordPress.

Etapa 2: Criar a pilha do pipeline

Para criar o pipeline do WordPress, você usará um exemplo de modelo do CloudFormation. Além de criar o pipeline, o modelo define perfis de serviço do AWS Identity and Access Management(IAM) para o CodePipeline e o CloudFormation, um bucket do S3 para o armazenamento de artefatos do CodePipeline e um tópico do Amazon Simple Notification Service (Amazon SNS) ao qual o pipeline envia notificações, como as notificações sobre avaliações. O exemplo de modelo facilita o provisionamento e configura esses recursos em uma única pilha do CloudFormation.

Para obter mais detalhes sobre a configuração do pipeline, consulte [O que o pipeline faz](#).

Important

O modelo do WordPress de amostra cria uma instância do EC2 que requer uma conexão com a Internet. Verifique se a sua VPC padrão e sub-rede permitem o tráfego para a Internet.

Como criar a pilha do pipeline

1. Faça download do modelo de exemplo em <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic-pipeline.yml>. Salve-o em seu computador.
2. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation/>.
3. Escolha uma região da AWS que seja compatível com o CodePipeline e o CloudFormation.

Para obter mais informações, consulte [CodePipeline endpoints and quotas](#) e [CloudFormation endpoints and quotas](#) na Referência geral da AWS.

4. Selecione Criar pilha.

5. Em Specify template (Especificar modelo), escolha Upload a template file (Fazer upload de um arquivo de modelo) e escolha o modelo que você acabou de obter por download, `basic-pipeline.yml`.
6. Escolha Próximo.
7. Para Nome da pilha, digite `sample-WordPress-pipeline`.
8. Na seção Parâmetros, especifique os seguintes valores de parâmetros e, em seguida, escolha Próximo. Ao definir parâmetros da pilha, se você mantiver os mesmos nomes do modelo do WordPress e dos seus arquivos de configuração, será possível usar os valores padrão. Caso contrário, especifique os nomes dos arquivos que você usou.

PipelineName

O nome do seu pipeline, como `WordPress-test-pipeline`.

S3 Bucket

O nome do bucket do S3 onde você salvou seu artefato (arquivo `.zip`).

SourceS3Key

O nome de arquivo do artefato. Se você salvou o artefato em uma pasta, o inclua como parte do nome de arquivo, como `folder/subfolder/wordpress-single-instance.zip`.

E-mail da

O endereço de e-mail para o qual o CodePipeline envia a notificação do pipeline, como `myemail@example.com`.

9. Para esta demonstração, você não precisa adicionar tags ou especificar configurações avançadas, portanto escolha Próximo.
10. Verifique se o nome da pilha e o URL do modelo estão corretos e escolha Create stack (Criar pilha).
11. Para confirmar que você está ciente de que o CloudFormation pode criar recursos do IAM, escolha a caixa de seleção.

Pode levar vários minutos para o AWS CloudFormation criar sua pilha. Para monitorar o progresso, visualize os eventos da pilha. Para obter mais informações, consulte [Monitorar o progresso da pilha](#).

Após a criação da pilha, o CodePipeline inicia seu novo pipeline. Para visualizar seu status, consulte o [console do CodePipeline](#). Na lista de pipelines, escolha `WordPress-test-pipeline`.

O que o pipeline faz

Esta seção explica os três estágios do pipeline, usando trechos do modelo de pipeline do WordPress de amostra.

Fase 1: Fonte

A primeira etapa do pipeline é um estágio de origem no qual você especifica o local do seu código-fonte. Cada vez que você envia uma revisão para esse local, o CodePipeline executa novamente o pipeline.

O código-fonte está localizado em um bucket do S3 e é identificado por seu nome de arquivo. Você especificou estes valores como valores de parâmetros de entrada quando você criou a pilha do pipeline. Para permitir o uso do artefato de origem em estágios subsequentes, o trecho especifica a propriedade `OutputArtifacts` com o nome de `TemplateSource`. Para usar esse artefato em etapas posteriores, você especifica `TemplateSource` como um artefato de entrada.

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
        S3ObjectKey: !Ref 'SourceS3Key'
      OutputArtifacts:
        - Name: TemplateSource
```

Fase 2: TestStage

Na etapa `TestStage`, o pipeline cria a pilha de teste, espera a aprovação e, em seguida, exclui a pilha de teste.

Para a ação `CreateStack`, o pipeline usa o arquivo de configuração de teste e o modelo do WordPress para criar a pilha de teste. Ambos os arquivos estão contidos no artefato de entrada `TemplateSource`, que é obtido a partir do estágio de origem. O trecho usa o modo de ação `REPLACE_ON_FAILURE`. Se houver falha na criação da pilha, o pipeline o substitui para que você não precise limpar ou solucionar problemas na pilha antes de poder executar novamente o pipeline.

O modo de ação é útil para a iteração rápida em pilhas de teste. Para a propriedade RoleArn, o valor é uma função de serviço do AWS CloudFormation que é declarada em outro lugar no modelo.

A ação ApproveTestStack pausa o pipeline e envia uma notificação para o endereço de e-mail que você especificou quando você criou a pilha do pipeline. Embora o pipeline esteja pausado, você poderá verificar a pilha de teste do WordPress e seus recursos. Use o CodePipeline para [aprovar ou rejeitar](#) esta ação. A propriedade CustomData inclui uma descrição da ação que você está aprovando para que o pipeline adicione ao e-mail de notificação.

Depois de aprovar essa ação, o CodePipeline passa para a ação DeleteTestStack e exclui a pilha de teste do WordPress e seus recursos.

```
- Name: TestStage
  Actions:
    - Name: CreateStack
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: REPLACE_ON_FAILURE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref TestStackName
        TemplateConfiguration: !Sub "TemplateSource::${TestStackConfig}"
        TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
      RunOrder: '1'
    - Name: ApproveTestStack
      ActionTypeId:
        Category: Approval
        Owner: AWS
        Provider: Manual
        Version: '1'
      Configuration:
        NotificationArn: !Ref CodePipelineSNSTopic
        CustomData: !Sub 'Do you want to create a change set against the production
stack and delete the ${TestStackName} stack?'
      RunOrder: '2'
    - Name: DeleteTestStack
      ActionTypeId:
        Category: Deploy
```

```
Owner: AWS
Provider: CloudFormation
Version: '1'
Configuration:
  ActionMode: DELETE_ONLY
  RoleArn: !GetAtt [CFNRole, Arn]
  StackName: !Ref TestStackName
RunOrder: '3'
```

Fase 3: ProdStage

A etapa ProdStage de pipeline cria um conjunto de alterações na pilha de produção existente, aguarda a aprovação e, em seguida, executa o conjunto de alterações.

Um conjunto de alterações fornece uma pré-visualização de todas as modificações que o AWS CloudFormation fará na sua pilha de produção antes de implementá-las. Em seu primeiro pipeline executado, você não terá uma pilha de produção em execução. O conjunto de alterações mostra as ações que o AWS CloudFormation executa ao criar a pilha de teste. Para criar o conjunto de alterações, a ação CreateChangeSet usa o modelo de amostra do WordPress e a configuração de modelo de produção do artefato de entrada TemplateSource.

Semelhante à etapa anterior, a ação ApproveChangeSet pausa o pipeline e envia uma notificação por e-mail. Embora o pipeline esteja pausado, você poderá visualizar o conjunto de alterações para verificar todas as modificações propostas à pilha de produção do WordPress. Use o CodePipeline para [aprovar ou rejeitar](#) esta ação para continuar ou interromper o pipeline, respectivamente.

Depois de aprovar essa ação, a ação ExecuteChangeSet executa o conjunto de alterações para que o AWS CloudFormation execute todas as ações descritas no conjunto de alterações. Para a execução inicial, o AWS CloudFormation cria a pilha de produção do WordPress. Em execuções subsequentes, o AWS CloudFormation atualiza a pilha.

```
- Name: ProdStage
  Actions:
    - Name: CreateChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
```



```

    ActionMode: CHANGE_SET_REPLACE
    RoleArn: !GetAtt [CFNRole, Arn]
    StackName: !Ref ProdStackName
    ChangeSetName: !Ref ChangeSetName
    TemplateConfiguration: !Sub "TemplateSource::${ProdStackConfig}"
    TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
  RunOrder: '1'
- Name: ApproveChangeSet
  ActionTypeId:
    Category: Approval
    Owner: AWS
    Provider: Manual
    Version: '1'
  Configuration:
    NotificationArn: !Ref CodePipelineSNSTopic
    CustomData: !Sub 'A new change set was created for the ${ProdStackName} stack.
Do you want to implement the changes?'
  RunOrder: '2'
- Name: ExecuteChangeSet
  ActionTypeId:
    Category: Deploy
    Owner: AWS
    Provider: CloudFormation
    Version: '1'
  Configuration:
    ActionMode: CHANGE_SET_EXECUTE
    ChangeSetName: !Ref ChangeSetName
    RoleArn: !GetAtt [CFNRole, Arn]
    StackName: !Ref ProdStackName
  RunOrder: '3'

```

Etapa 3: Visualizar a pilha do WordPress

Enquanto passa pelo pipeline, o CodePipeline usa o CloudFormation para criar as pilhas de teste e de produção. Para ver o status dessas pilhas e sua saída, use o console de CloudFormation.

Para visualizar uma pilha

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation/>.
2. Dependendo se o pipeline estiver na etapa de teste ou de produção, selecione a pilha Test-MyWordPressSite ou Prod-MyWordPressSite.
3. Para verificar o status de sua pilha, visualize os [eventos](#) da pilha.

Se a pilha estiver em um estado de falha, visualize o motivo do status para encontrar o erro da pilha. Corrija o erro e, em seguida, execute novamente o pipeline. Se a pilha estiver no estado `CREATE_COMPLETE`, visualize suas saídas para obter o URL do site do WordPress.

Você usou o CodePipeline com êxito para criar um fluxo de trabalho de entrega contínuo para uma amostra do site do WordPress. Se você enviar as alterações ao bucket do S3, o CodePipeline detecta automaticamente uma nova versão e, em seguida, executa novamente o pipeline. Este fluxo de trabalho facilita o envio e o teste das alterações antes de fazer mudanças em seu site de produção.

Etapa 4: Limpar os recursos

Para ter certeza de que você não será cobrado por serviços indesejados, exclua seus recursos.

Important

Exclua as pilhas de teste e produção do WordPress antes de excluir a pilha do pipeline. A pilha do pipeline contém uma função de serviço que é necessária para excluir as pilhas do WordPress. Se você excluiu as pilhas do pipeline primeiro, você poderá associar outro nome de recurso da Amazon (ARN) de função de serviço às pilhas do WordPress e, em seguida, excluí-las.

Para excluir objetos no armazenamento de artefato

1. Abra o console do Amazon S3, em <https://console.aws.amazon.com/s3/>.
2. Selecione o bucket do S3 que o CodePipeline usou como armazenamento de artefatos do pipeline.

O nome do bucket segue o formato: *stackname-artifactstorebucket-id*. Se você seguiu esta apresentação, o nome do bucket pode ser semelhante ao exemplo a seguir: `sample-WordPress-pipeline-artifactstorebucket-12345abcd12345`.

3. Exclua todos os objetos no bucket do S3 do armazenamento de artefato.

Ao excluir a pilha do pipeline na próxima etapa, esse bucket deve estar vazio. Caso contrário, o CloudFormation não poderá excluir o bucket.

Para excluir pilhas

1. No console do CloudFormation, escolha a pilha que você deseja excluir.

Se as pilhas do WordPress que foram criadas pelo pipeline ainda estão em execução, selecione-as primeiro. Por padrão, os nomes de pilha são `Test-MyWordPressSite` e `Prod-MyWordPressSite`.

Se você já excluiu as pilhas do WordPress, selecione a pilha `sample-WordPress-pipeline`.

2. Escolha Ações e, em seguida, escolha Excluir pilha.
3. Na mensagem de confirmação, escolha Sim, excluir.

O CloudFormation excluirá a pilha de todos os recursos de pilha, como a instância do EC2, o tópico de notificação, o perfil de serviço e o pipeline.

Agora que você entende como criar um fluxo de trabalho básico do CloudFormation com o CodePipeline, pode usar o exemplo de modelo e os artefatos como ponto de partida para criar os seus próprios.

Consulte também

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com esses parâmetros.

- Para obter mais informações sobre os parâmetros de ação do CloudFormation no CodePipeline, consulte [CloudFormation deploy action configuration reference](#) no AWS CodePipeline User Guide.
- Para obter exemplos de valores do modelo por provedor de ação, como para o campo `Owner` ou para os campos `configuration`, consulte a [Referência de estruturas de ações](#) no Guia do usuário do AWS CodePipeline.
- Para baixar exemplos de modelos de pilha de pipeline no formato YAML ou JSON, consulte [Tutorial: Create a pipeline with CloudFormation](#) no AWS CodePipeline User Guide.

CloudFormationReferência de propriedades de configuração DO

Ao criar um pipeline do CodePipeline, adicione uma ação `Deploy` a esse pipeline com o CloudFormation como provedor. Depois, especifique qual ação do CloudFormation o pipeline invocará e as configurações da ação. Este tópico descreve as propriedades de configuração do CloudFormation. Para especificar propriedades, você pode usar o console do CodePipeline ou

criar um objeto JSON para ser usado na AWS CLI, na API do CodePipeline ou nos modelos do CloudFormation.

Tópicos

- [Propriedades de configuração \(console\)](#)
- [Propriedades de configuração \(objeto JSON\)](#)
- [Consulte também](#)

Propriedades de configuração (console)

O [console](#) do CodePipeline mostra as propriedades de configuração e indica as propriedades que são necessárias com base no modo de ação que você escolher.

Note

Ao criar um novo pipeline, você pode especificar somente os modos de ação Criar ou atualizar uma pilha ou Criar ou substituir um conjunto de alterações. As propriedades na seção Advanced (Avançado) estão disponíveis somente quando você edita um pipeline.

Modo de ação

A ação do CloudFormation que o CodePipeline invoca ao processar o estágio associado. Escolha um dos seguintes modos de ação:

- Criar ou substituir um conjunto de alterações cria o conjunto de alterações, se ele não existe, com base no nome da pilha e no modelo que você envia. Se o conjunto de alterações existir, o CloudFormation o exclui e, em seguida, cria um novo.
- Criar ou atualizar uma pilha cria a pilha, se a pilha especificada não existe. Se a pilha existir, o CloudFormation a atualiza. Use esta ação para atualizar pilhas existentes. O CodePipeline não substituirá a pilha.
- Excluir uma pilha exclui uma pilha. Se você especificar uma pilha que não existe, a ação será concluída com êxito sem excluir uma pilha.
- Executar um conjunto de alterações executa um conjunto de alterações.
- Substituir uma pilha com falhas cria a pilha, se a pilha especificada não existe. Se a pilha existir e estiver em estado de falha (reportado como ROLLBACK_COMPLETE, ROLLBACK_FAILED, CREATE_FAILED, DELETE_FAILED ou UPDATE_ROLLBACK_FAILED), o CloudFormation

excluirá a pilha e criará uma nova. Se a pilha não estiver em um estado de falha, o CloudFormation a atualiza. Use esta ação para substituir as pilhas com falha sem recuperá-las nem solucionar o problema delas. Geralmente, esse modo é escolhido para testes.

Nome da pilha

O nome associado a uma pilha existente ou a uma pilha que você deseja criar. O nome deve ser exclusivo na região da AWS em que você está criando a pilha.

Note

Um nome de pilha pode conter apenas caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hífens. Ele deve começar com um caractere alfabético e não pode ter mais de 128 caracteres.

Change set name (Nome do conjunto de alterações)

O nome de um conjunto de alterações existente ou um novo conjunto de alterações que você deseja criar para a pilha especificada.

Modelo

O local de um arquivo de modelo do CloudFormation, que segue o formato

ArtifactName::TemplateFileName.

Configuração do modelo

O local de um arquivo de configuração do modelo, que segue o formato

ArtifactName::TemplateConfigurationFileName. O arquivo de configuração de modelo pode conter valores de parâmetros do modelo, uma política de pilha e tags. Se você incluir informações confidenciais, como senhas, restrinja o acesso a este arquivo. Para obter mais informações, consulte [CloudFormationArtefatos do](#).

Capacidades

Para pilhas que contêm determinados recursos, confirme explicitamente que o CloudFormation pode criar ou atualizar esses recursos. Por exemplo, especifique `CAPABILITY_IAM` se o modelo da pilha contiver recursos do AWS Identity and Access Management (IAM). Para obter mais informações, consulte os parâmetros de solicitação da operação da API [CreateStack](#).

Se houver recursos do IAM no modelo de pilha, especifique essa propriedade.

Você pode especificar mais de uma funcionalidade.

Nome do perfil

O nome do perfil de serviço do IAM que o CloudFormation assume quando ele opera em recursos na pilha especificada.

Nome do arquivo de saída

Na seção Advanced (Avançado), você pode especificar um nome de arquivo de saída, como `CreateStackOutput.json`, que o CodePipeline adiciona ao [artefato de saída](#) depois de executar a ação especificada. O artefato de saída contém um arquivo JSON com o conteúdo da seção `Outputs` do modelo CloudFormation.

Se você não especificar um nome, o CodePipeline não gerará um artefato de saída.

Substituições de parâmetros

Os parâmetros são definidos em seu modelo e permitem que você insira valores personalizados ao criar ou atualizar uma pilha. É possível especificar um objeto JSON que substitui os valores dos parâmetros do modelo no arquivo de configuração do modelo. Todos os nomes de parâmetros devem estar presentes no modelo de pilha. Para obter mais informações, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Note

Há um limite de tamanho máximo de 1 KB para o objeto JSON que pode ser armazenado na propriedade `ParameterOverrides`.

Recomendamos que você use o arquivo de configuração do modelo para especificar a maioria dos valores de parâmetro. Use substituições de parâmetros para especificar somente valores de parâmetros dinâmicos. Os parâmetros dinâmicos são desconhecidos até que você execute o pipeline.

O exemplo a seguir define um valor para o parâmetro `ParameterName` usando uma função de substituição de parâmetro. A função recupera um valor de um artefato de entrada do CodePipeline. Para obter mais informações sobre funções de substituição de parâmetros, consulte [Usar funções de substituição de parâmetros com pipelines CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

```
}
```

Propriedades de configuração (objeto JSON)

Ao especificar CloudFormation como um provedor para uma ação de estágio, defina as seguintes propriedades na propriedade Configuration. Use o objeto JSON para a AWS CLI, a API do CodePipeline ou nos modelos do CloudFormation. Veja exemplos em [Demonstração: Criar um pipeline para pilhas de produção e teste](#) e [CloudFormationReferência de propriedades de configuração DO](#).

ActionMode

A ação do CloudFormation que o CodePipeline invoca ao processar o estágio associado. Especifique apenas um dos seguintes modos de ação:

- **CHANGE_SET_EXECUTE** executa o conjunto de alterações.
- **CHANGE_SET_REPLACE** cria um conjunto de alterações, se ainda não existir, com base no nome da pilha e no modelo enviado. Se o conjunto de alterações existir, o CloudFormation o exclui e, em seguida, cria um novo.
- **CREATE_UPDATE** cria a pilha se a especificada não existir. Se a pilha existir, o CloudFormation a atualiza. Use esta ação para atualizar pilhas existentes. O CodePipeline não substituirá a pilha.
- **DELETE_ONLY** exclui uma pilha. Se você especificar uma pilha que não existe, a ação será concluída com êxito sem excluir uma pilha.
- **REPLACE_ON_FAILURE** cria uma pilha quando a pilha especificada não existe. Se a pilha existir e estiver no estado de falha (reportado como **ROLLBACK_COMPLETE**, **ROLLBACK_FAILED**, **CREATE_FAILED**, **DELETE_FAILED** ou **UPDATE_ROLLBACK_FAILED**), o CloudFormation a excluirá e, em seguida, criará uma nova. Se a pilha não estiver em um estado de falha, o CloudFormation a atualiza. Use esta ação para substituir automaticamente as pilhas com falha sem recuperá-las nem solucionar o problema delas. Geralmente, esse modo é escolhido para testes.

Essa propriedade é obrigatória.

Capabilities

Para pilhas que contêm determinados recursos, confirme explicitamente que o CloudFormation pode criar ou atualizar esses recursos. Por exemplo, especifique **CAPABILITY_IAM** se o modelo

da pilha contiver recursos do AWS Identity and Access Management (IAM). Para obter mais informações, consulte os parâmetros de solicitação da operação da API [CreateStack](#).

Essa propriedade é condicional. Se houver recursos do IAM no modelo de pilha, especifique essa propriedade.

Você pode especificar várias funcionalidades. O exemplo a seguir adiciona as propriedades `CAPABILITY_IAM` e `CAPABILITY_AUTO_EXPAND` ao modelo:

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
  ChangeSetName: pipeline-changeset
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {
  "ActionMode": "CHANGE_SET_REPLACE",
  "Capabilities": "CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND",
  "ChangeSetName": "pipeline-changeset",
  "RoleArn": "CloudFormation_Role_ARN",
  "StackName": "my-pipeline-stack",
  "TemplateConfiguration": "my-pipeline-stack::template-configuration.json",
  "TemplatePath": "my-pipeline-stack::template-export.yml"
}
```

ChangeSetName

O nome de um conjunto de alterações existente ou um novo conjunto de alterações que você deseja criar para a pilha especificada.

Essa propriedade é necessária para os seguintes modos de ação: `CHANGE_SET_REPLACE` e `CHANGE_SET_EXECUTE`. Para todos os outros modos de ação, essa propriedade será ignorado.

OutputFileName

Um nome para o arquivo de saída, como `CreateStackOutput.json`. O CodePipeline adiciona o arquivo ao [artefato de saída](#) depois de executar a ação especificada. O artefato de saída contém um arquivo JSON com o conteúdo da seção `Outputs` do modelo CloudFormation.

Essa propriedade é opcional. Se você não especificar um nome, o CodePipeline não gerará um artefato de saída.

ParameterOverrides

Os parâmetros são definidos em seu modelo e permitem que você insira valores personalizados ao criar ou atualizar uma pilha. É possível especificar um objeto JSON que substitui os valores dos parâmetros do modelo no arquivo de configuração do modelo. Todos os nomes de parâmetros devem estar presentes no modelo de pilha. Para obter mais informações, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

O exemplo a seguir adiciona as substituições dos parâmetros `InstanceType` e `KeyName` ao modelo:

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_NAMED_IAM
  ChangeSetName: pipeline-changeset
  ParameterOverrides: '{"InstanceType": "t2.small", "KeyName": "my-keypair"}'
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {
  "ActionMode": "CHANGE_SET_REPLACE",
  "Capabilities": "CAPABILITY_NAMED_IAM",
  "ChangeSetName": "pipeline-changeset",
  "ParameterOverrides": '{"InstanceType": "t2.small", "KeyName": "my-keypair"}',
  "RoleArn": "CloudFormation_Role_ARN",
  "StackName": "my-pipeline-stack",
```

```
"TemplateConfiguration": "my-pipeline-stack::template-configuration.json",
"TemplatePath": "my-pipeline-stack::template-export.yml"
}
```

 Note

O tamanho máximo do objeto JSON que pode ser armazenado na propriedade `ParameterOverrides` é 1 kilobyte.

Recomendamos que você use o arquivo de configuração do modelo para especificar a maioria dos valores de parâmetro. Use substituições de parâmetros para especificar somente valores de parâmetros dinâmicos. Os valores dos parâmetros dinâmicos são desconhecidos até que você execute o pipeline.

O exemplo a seguir define um valor para o parâmetro `ParameterName` usando uma função de substituição de parâmetro. A função recupera um valor de um artefato de entrada do CodePipeline. Para obter mais informações sobre funções de substituição de parâmetros, consulte [Usar funções de substituição de parâmetros com pipelines CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Essa propriedade é opcional.

RoleArn

O nome do recurso da Amazon (ARN) do perfil de serviço do IAM que o CloudFormation assume quando ele funciona com recursos em uma pilha.

Essa propriedade é necessária para os seguintes modos de ação: `CREATE_UPDATE`, `REPLACE_ON_FAILURE`, `DELETE_ONLY` e `CHANGE_SET_REPLACE`. `RoleArn` não é aplicado ao executar um conjunto de alterações. Se você não usar o CodePipeline para criar o conjunto de alterações, verifique se o conjunto de alterações ou a pilha está associada a uma função.

StackName

O nome de uma pilha existente ou de uma pilha que você deseja criar.

Essa propriedade é necessária para todos os modos de ação.

TemplateConfiguration

TemplateConfiguration é o arquivo de configuração do modelo. Inclua o arquivo em um artefato de entrada para essa ação. O nome do arquivo de configuração de modelo segue este formato:

Artifactname::TemplateConfigurationFileName

Artifactname é o nome do artefato de entrada, da forma como aparece no CodePipeline. Por exemplo, um estágio de origem com o nome de artefato de SourceArtifact e um nome de arquivo de test-configuration.json cria um nome TemplateConfiguration, conforme mostrado neste exemplo:

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

O arquivo de configuração de modelo pode conter valores de parâmetro do modelo e uma política de pilha. Se você incluir informações confidenciais, como senhas, restrinja o acesso a este arquivo. Para obter um exemplo de arquivo de configuração de modelo, consulte [CloudFormationArtefatos do](#).

Essa propriedade é opcional.

TemplatePath

TemplatePath representa o arquivo de modelo do CloudFormation. Inclua o arquivo em um artefato de entrada para essa ação. O nome do arquivo segue este formato:

Artifactname::TemplateFileName

Artifactname é o nome do artefato de entrada, da forma como aparece no CodePipeline. Por exemplo, um estágio de origem com o nome de artefato de SourceArtifact e um nome de arquivo de template.yaml cria um nome TemplatePath, conforme mostrado neste exemplo:

```
"TemplatePath": "SourceArtifact::template.yaml"
```

Essa propriedade é necessária para os seguintes modos de ação: CREATE_UPDATE, REPLACE_ON_FAILURE e CHANGE_SET_REPLACE. Para todos os outros modos de ação, essa propriedade será ignorado.

Consulte também

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com esses parâmetros.

- Para obter mais informações sobre os parâmetros de ação do CloudFormation no CodePipeline, consulte [CloudFormation deploy action configuration reference](#) no AWS CodePipeline User Guide.
- Para obter exemplos de valores do modelo por provedor de ação, como para o campo Owner ou para os campos configuration, consulte a [Referência de estruturas de ações](#) no Guia do usuário do AWS CodePipeline.
- Para baixar exemplos de modelos de pilha de pipeline no formato YAML ou JSON, consulte [Tutorial: Create a pipeline with CloudFormation](#) no AWS CodePipeline User Guide.

CloudFormationArtefatos do

O CodePipeline realiza tarefas em artefatos enquanto o CodePipeline executa um pipeline. Para o CloudFormation, artefatos podem incluir um arquivo de modelo de pilha, um arquivo de configuração de modelo ou ambos. O CodePipeline usa esses artefatos para trabalhar com pilhas e conjuntos de alterações do CloudFormation.

Se você usar o Amazon Simple Storage Service (Amazon S3) como um repositório de origem, é necessário compactar o modelo e os arquivos de configuração de modelo em um único arquivo antes de fazer upload para um bucket do S3. Para outros repositórios, como GitHub e AWS CodeCommit, faça upload dos artefatos sem compactá-los. Para obter mais informações, consulte [Create a pipeline, stages, and actions](#) no AWS CodePipeline User Guide.

Você pode adicionar quantos arquivos precisar ao seu repositório. Por exemplo, você pode querer incluir duas configurações diferentes para o mesmo modelo: uma para uma configuração de teste e outra para uma configuração de produção.

Este tópico descreve cada tipo de artefato.

Tópicos

- [Arquivo de modelo de pilha](#)
- [Arquivo de configuração de modelo](#)
- [Consulte também](#)

Arquivo de modelo de pilha

Um arquivo de modelo de pilha define os recursos que o CloudFormation provisiona e configura. Esses arquivos são os mesmos arquivos de modelo que você usa ao criar ou atualizar pilhas usando o CloudFormation. Você pode usar os modelos formatados para YAML ou JSON. Para mais informações sobre modelos, consulte [Seções de modelos do CloudFormation](#).

Arquivo de configuração de modelo

Um arquivo de configuração de modelo é um arquivo de texto formatado para JSON que especifica os valores dos parâmetros do modelo, uma [política de pilha](#) e tags. Use esses arquivos de configuração para especificar valores de parâmetro ou uma política de pilha para uma pilha. Todos os valores de parâmetro que você especifica devem ser declarados no modelo associado.

Se você incluir informações confidenciais nesse arquivo, como senhas, restrinja o acesso a ele. Por exemplo, se você fizer upload do seu artefato para um bucket S3, use [políticas de bucket S3](#) ou [políticas de usuário](#) para restringir o acesso.

Para criar um arquivo de configuração, use o formato a seguir:

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

O exemplo a seguir especifica TestEC2Key para o parâmetro KeyName, adiciona uma tag Department com o valor Marketing e adiciona uma política de pilha que permite todas as ações de atualização, com exceção das atualizações que excluem um recurso.

```
{
```

```
"Parameters" : {
  "KeyName" : "TestEC2Key"
},
"Tags" : {
  "Department" : "Marketing"
},
"StackPolicy" : {
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal" : "*",
      "Resource" : "*"
    }
  ]
}
```

Consulte também

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com esses parâmetros.

- Para obter mais informações sobre os parâmetros de ação do CloudFormation no CodePipeline, consulte [CloudFormation deploy action configuration reference](#) no AWS CodePipeline User Guide.
- Para obter exemplos de valores do modelo por provedor de ação, como para o campo Owner ou para os campos configuration, consulte a [Referência de estruturas de ações](#) no Guia do usuário do AWS CodePipeline.
- Para baixar exemplos de modelos de pilha de pipeline no formato YAML ou JSON, consulte [Tutorial: Create a pipeline with CloudFormation](#) no AWS CodePipeline User Guide.

Usar funções de substituição de parâmetros com pipelines CodePipeline

Em um estágio do CodePipeline, você pode especificar [substituições de parâmetros](#) para ações do CloudFormation. As substituições de parâmetros permitem que você especifique valores de parâmetro do modelo que substituem os valores em um arquivo de configuração do modelo. O CloudFormation fornece funções para ajudá-lo a especificar valores dinâmicos (valores que são desconhecidos até o pipeline ser executado).

Tópicos

- [Fn::GetArtifactAtt](#)
- [Fn::GetParam](#)
- [Consulte também](#)

Fn::GetArtifactAtt

A função `Fn::GetArtifactAtt` recupera o valor de um atributo de um artefato de entrada, como o nome do bucket S3 no qual o artefato está armazenado. Use esta função para especificar atributos de um artefato, como o nome de arquivo ou nome do bucket do Amazon S3.

Quando você executa um pipeline, o CodePipeline copia e grava arquivos no armazenamento de artefatos do pipeline (um bucket do S3). O CodePipeline gera os nomes de arquivos no armazenamento de artefatos. Esses nomes de arquivos são desconhecidas antes de o pipeline ser executado.

Por exemplo, no seu pipeline, é possível ter um estágio de origem no qual o copia o código-fonte da função do AWS Lambda no armazenamento de artefatos. No próximo estágio, há um modelo do CloudFormation que cria a função do Lambda, mas o CloudFormation requer o nome de arquivo para criar a função. Use a função `Fn::GetArtifactAtt` para passar o bucket do S3 e os nomes de arquivos exatos.

Sintaxe

Use a sintaxe a seguir para recuperar um valor de atributo de um artefato.

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

`artifactName`

O nome do artefato de entrada. Declare este artefato como entrada para a ação associada.

`attributeName`

O nome do atributo do artefato com o valor você deseja recuperar. Para obter detalhes sobre cada atributo de artefato, consulte a seção Atributos a seguir.

Exemplo

As seguintes substituições de parâmetro especificam os parâmetros `BucketName` e `ObjectKey`, recuperando o nome do bucket do S3 e o nome de arquivo do artefato `LambdaFunctionSource`. Este exemplo pressupõe que o CodePipeline copiou o código-fonte da função Lambda e o salvou como um artefato, por exemplo, como parte de um estágio de origem.

```
{
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"] },
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"] }
}
```

Atributos

É possível recuperar os seguintes atributos para um artefato.

BucketName

O nome do bucket do S3 no qual o artefato está armazenado.

ObjectKey

O nome do arquivo `.zip` que contém o artefato que é gerado pelo CodePipeline, como `1ABCyZZ.zip`.

URL

O URL do Amazon Simple Storage Service (Amazon S3) do artefato, como `https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip`.

Fn::GetParam

A função `Fn::GetParam` retorna um valor de um par chave/valor em um arquivo em um formato JSON. O arquivo JSON deve estar incluído em um artefato.

Use esta função para recuperar os valores de saída de uma pilha do CloudFormation e usá-los como entrada para outra ação. Por exemplo, se você especificar um nome de arquivo de saída para uma ação do CloudFormation, o CodePipeline salvará essa saída em um arquivo JSON e a adicionará ao arquivo `.zip` do artefato de saída. Use a função `Fn::GetParam` para recuperar o valor de saída e usá-lo como entrada para outra ação.

Sintaxe

Use a sintaxe a seguir para recuperar um valor a partir de um par chave/valor.

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

artifactName

O nome do artefato, que deve ser incluído como um artefato de entrada para a ação associada.

JSONFileName

O nome de um arquivo JSON que está contido no artefato.

keyName

O nome da chave com o valor você deseja recuperar.

Exemplos

Os exemplos a seguir demonstram como usar a função `Fn::GetParam` em uma substituição de parâmetro.

Sintaxe

A substituição de parâmetro a seguir especifica o parâmetro `WebSiteURL` recuperando o valor da chave `URL` do arquivo `stack-output.json` que está no `WebStackOutput` artefato.

```
{  
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"] }  
}
```

Trechos de modelo do CloudFormation

Os trechos de modelo do CloudFormation a seguir, em um pipeline do CodePipeline, demonstram como passar saídas da pilha. Esses trechos mostram dois estágios de definição do pipeline. O primeiro estágio cria uma pilha e salva as saídas no arquivo `TestOutput.json` no artefato `StackAOutput`. Esses valores são especificados pelas propriedades `OutputFileName` e `OutputArtifacts`.

o nome do artefato de entrada de origem é `TemplateSource`. O nome do modelo de pilha é `teststackA.yaml`, e o nome do arquivo de configuração é `test-configuration.json`.

Em ambos os estágios, esses valores são especificados para as propriedades `TemplateConfiguration` e `TemplatePath`, conforme mostrado:

```
TemplateConfiguration: TemplateSource::test-configuration.json
TemplatePath: TemplateSource::teststackA.yaml
```

Exemplo Criar estágio A da pilha

```
- Name: CreateTestStackA
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        OutputFileName: TestOutput.json
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackA
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackA.yaml
      InputArtifacts:
        - Name: TemplateSource
      OutputArtifacts:
        - Name: StackAOutput
      RunOrder: '1'
```

Em um estágio posterior, a pilha B usa as saídas da pilha A. Na propriedade `ParameterOverrides`, o exemplo usa a função `Fn::GetParam` para especificar o parâmetro `StackBInputParam`. O valor resultante é o valor associado à chave `StackAOutputName`.

Exemplo Criar estágio B da pilha

```
- Name: CreateTestStackB
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
```

```
Provider: CloudFormation
Version: '1'
Configuration:
  ActionMode: CREATE_UPDATE
  Capabilities: CAPABILITY_IAM
  RoleArn: !GetAtt [CFNRole, Arn]
  StackName: StackB
  TemplateConfiguration: TemplateSource::test-configuration.json
  TemplatePath: TemplateSource::teststackB.yaml
  ParameterOverrides: |
    {
      "StackBInputParam" : { "Fn::GetParam" : ["StackAOutput", "TestOutput.json",
"StackAOutputName"]}
    }
  InputArtifacts:
    - Name: TemplateSource
    - Name: StackAOutput
  RunOrder: '1'
```

Consulte também

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com esses parâmetros.

- Para obter mais informações sobre os parâmetros de ação do CloudFormation no CodePipeline, consulte [CloudFormation deploy action configuration reference](#) no AWS CodePipeline User Guide.
- Para obter exemplos de valores do modelo por provedor de ação, como para o campo `Owner` ou para os campos `configuration`, consulte a [Referência de estruturas de ações](#) no Guia do usuário do AWS CodePipeline.
- Para baixar exemplos de modelos de pilha de pipeline no formato YAML ou JSON, consulte [Tutorial: Create a pipeline with CloudFormation](#) no AWS CodePipeline User Guide.

O Guia de referência de modelos do CloudFormation

Informações detalhadas sobre os recursos e componentes disponíveis para a criação de modelos do CloudFormation agora podem ser encontradas no [Guia de referência de modelos do CloudFormation](#). Este guia serve como referência técnica para modelos e complementa o conteúdo de introdução existente no Guia do usuário do CloudFormation.

O Guia de referência de modelos do CloudFormation fornece detalhes sobre:

- Tipos de recursos e suas propriedades
- DeletionPolicy e outros atributos em nível de recurso
- Funções intrínsecas, como Ref e Sub
- Transformações (macros do CloudFormation)
- Scripts auxiliares para configuração de instâncias do EC2
- Definições de tipos de recursos legíveis por máquina

Este guia é atualizado regularmente para refletir:

- Novos tipos de recursos
- Alterações em propriedades
- Valor(es) de retorno
- Atualizações feitas na documentação

Segurança no CloudFormation

A segurança na nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de data centers e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem:** a AWS é responsável pela proteção da infraestrutura que executa serviços AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao CloudFormation, consulte [Serviços da AWS em escopo por programa de conformidade](#).
- **Segurança na nuvem:** sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o CloudFormation. Os tópicos a seguir mostram como configurar o CloudFormation para atender aos seus objetivos de segurança e conformidade. Saiba como usar outros serviços AWS que ajudam a monitorar e proteger os recursos do CloudFormation.

Tópicos

- [Proteger pilhas do CloudFormation contra exclusão](#)
- [Impedir atualizações nos recursos de pilha](#)
- [Proteção de dados no CloudFormation](#)
- [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#)
- [Registrar em log chamadas de API do CloudFormation com o AWS CloudTrail](#)
- [Segurança da infraestrutura no CloudFormation](#)
- [Resiliência no CloudFormation](#)
- [Validação de conformidade do CloudFormation](#)
- [Análise de vulnerabilidade e configuração no CloudFormation](#)

- [Práticas recomendadas de segurança do CloudFormation](#)
- [Acessar o CloudFormation usando um endpoint de interface \(AWS PrivateLink\)](#)

Proteger pilhas do CloudFormation contra exclusão

É possível impedir que uma pilha seja excluída acidentalmente ativando a proteção contra encerramento na pilha. Se um usuário tentar excluir uma pilha com proteção contra encerramento ativada, a exclusão falhará e a pilha, incluindo o status dela, permanecerá inalterada. É possível ativar a proteção de encerramento em uma pilha ao criá-la. A proteção contra encerramento em pilhas está desativada por padrão. Você pode definir a proteção contra encerramento em uma pilha com qualquer status, exceto `DELETE_IN_PROGRESS` ou `DELETE_COMPLETE`.

Quando a proteção contra terminação é habilitada ou desabilitada em uma pilha, a mesma opção se estende a todas as pilhas aninhadas pertencentes a ela. Você não pode ativar ou desativar a proteção contra encerramento diretamente em uma pilha aninhada. Se um usuário tentar excluir diretamente uma pilha aninhada pertencente a uma pilha que tem a proteção contra encerramento ativada, a operação falhará e a pilha aninhada permanecerá inalterada.

No entanto, se um usuário executar uma atualização da pilha que exclua a pilha aninhada, o CloudFormation excluirá a pilha aninhada apropriadamente.

A proteção contra terminação é diferente da desativação de reversão. A proteção contra encerramento se aplica apenas a tentativas de excluir pilhas, enquanto a desativação de reversão se aplica à reversão automática quando a criação da pilha falha.

Para ativar a proteção contra encerramento ao criar uma pilha

Na página Specify stack options (Especificar opções da pilha) do assistente Create stack (Criar pilha), em Advanced options (Opções avançadas), expanda a seção Termination Protection (Proteção contra encerramento) e selecione Enable (Habilitar). Para obter mais informações, consulte [Configurar opções da pilha](#).

Habilitar ou desabilitar a proteção contra encerramento em uma pilha existente

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Selecione a pilha que você quiser.

 Note

Se NESTED for exibido ao lado do nome da pilha, a pilha é aninhada. Só é possível alterar a proteção contra encerramento na pilha raiz à qual a pilha aninhada pertence.

4. No painel de detalhes da pilha, selecione Stack actions (Ações da pilha) e Edit termination protection (Editar a proteção contra encerramento).

O CloudFormation exibe a caixa de diálogo Editar a proteção contra encerramento.

5. Escolha Enable (Habilitar) ou Disable (Desabilitar) e selecione Save (Salvar).

Habilitar ou desabilitar a proteção contra encerramento em uma pilha aninhada

Se NESTED for exibido ao lado do nome da pilha, a pilha é aninhada. Só é possível alterar a proteção contra encerramento na pilha raiz à qual a pilha aninhada pertence. Para alterar a proteção contra encerramento na pilha raiz:

1. Faça login no Console de gerenciamento da AWS e abra o console CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha sua Região da AWS.
3. Selecione a pilha aninhada que você quiser.
4. No painel Stack info (Informações da pilha), na seção Overview (Visão geral), selecione o nome da pilha listada como Root stack (Pilha raiz).

O CloudFormation exibe os detalhes da pilha raiz.

5. Escolha Stack actions (Ações da pilha) e selecione Edit Termination Protection (Editar a proteção contra encerramento).

O CloudFormation exibe a caixa de diálogo Editar a proteção contra encerramento.

6. Escolha Enable (Habilitar) ou Disable (Desabilitar) e selecione Save (Salvar).

Para habilitar ou desabilitar a proteção contra encerramento usando a linha de comando

Use o comando [update-termination-protection](#).

Controlar quem pode alterar a proteção contra encerramento em pilhas

Para habilitar ou desabilitar a proteção contra encerramento em pilhas, um usuário requer permissão para a ação `cloudformation:UpdateTerminationProtection`. Por exemplo, a política a seguir permite que os usuários ativem ou desativem a proteção contra encerramento em pilhas.

Para obter mais informações sobre como especificar permissões no CloudFormation, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Example Um exemplo da política que concede permissões para alterar a proteção contra encerramento de pilha

JSON

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

Impedir atualizações nos recursos de pilha

Quando você cria uma pilha, todas as ações de atualização são permitidas em todos os recursos. Por padrão, qualquer usuário com permissões de atualização da pilha pode atualizar todos os recursos na pilha. Durante uma atualização, alguns recursos podem exigir uma interrupção ou ser completamente substituídos, o que resulta em novos IDs físicos ou armazenamento completamente novo. Você pode impedir que a pilha de recursos seja involuntariamente atualizada ou excluída durante uma atualização da pilha usando uma política de pilha. Uma política de pilha é um documento JSON que define quais ações de atualização podem ser executadas nos recursos designados.

Depois de definir uma política de pilha, todos os recursos na pilha são protegidos por padrão. Para permitir atualizações em recursos específicos, você especifica uma instrução explícita `Allow` para

esses recursos em sua política de pilha. Você pode definir apenas uma política de pilha por pilha. Porém, é possível proteger vários recursos em uma única política. Uma política de pilha se aplica a todos os usuários do CloudFormation que tentarem atualizar a pilha. Você não pode associar diferentes políticas de pilha com diferentes usuários.

Uma política de pilha é aplicada somente durante as atualizações da pilha. Ela não fornece controles de acesso como uma política do AWS Identity and Access Management (IAM). Use uma política de pilha somente como um mecanismo à prova de falhas que impede atualizações acidentais em recursos de pilha específicos. Para controlar o acesso a recursos ou ações da AWS, use o IAM.

Tópicos

- [Exemplo de política de pilha](#)
- [Definir uma política de pilha](#)
- [Configurar uma política de pilha](#)
- [Atualizar recursos protegidos](#)
- [Modificar uma política de pilha](#)
- [Mais exemplos de políticas de pilha](#)

Exemplo de política de pilha

A política de pilha de exemplo a seguir impede atualizações no recurso ProductionDatabase:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Quando você define uma política de pilha, todos os recursos na pilha são protegidos por padrão. Para permitir atualizações em todos os recursos, adicionamos uma instrução `Allow` que permite que todas as ações em todos os recursos. Embora a instrução `Allow` especifique todos os recursos, a instrução explícita `Deny` a substitui para o recurso com o ID lógico `ProductionDatabase`. Essa instrução `Deny` impede todas as ações de atualização, como substituição ou exclusão, no recurso `ProductionDatabase`.

O elemento `Principal` é necessário, mas suporta apenas o caractere curinga (*), o que significa que a instrução aplica-se a todos os [principais](#).

Note

Durante uma atualização de pilha, o CloudFormation atualiza recursos que dependem de outros recursos atualizados. Por exemplo, o CloudFormation atualiza um recurso que faz referência a um recurso atualizado. O CloudFormation não faz alterações físicas, como o ID de recursos, para recursos atualizados automaticamente, mas se uma política de pilha é associada a esses recursos, você deve ter permissão para atualizá-las.

Definir uma política de pilha

Quando você cria uma pilha, nenhuma política de pilha é definida, de modo que todas as ações de atualização são permitidas em todos os recursos. Para proteger os recursos da pilha de ações de atualização, defina uma pilha de política e, em seguida, configure-a em sua pilha. Uma política de pilha é um documento JSON que define as ações de atualização de pilha do CloudFormation que os usuários do CloudFormation podem executar e os recursos que as ações se aplicam. Você pode definir a política de pilha ao criar uma pilha, especificando um arquivo de texto que contém a sua política de pilha ou digitando-a. Quando você define uma política de pilha em sua pilha, qualquer atualização não explicitamente permitida é negada por padrão.

Você define uma política de pilha com cinco elementos: `Effect`, `Action`, `Principal`, `Resource` e `Condition`. O pseudocódigo a seguir mostra a sintaxe da política de pilha.

```
{
  "Statement" : [
    {
      "Effect" : "Deny_or_Allow",
      "Action" : "update_actions",
      "Principal" : "*",
```

```
"Resource" : "LogicalResourceId/resource_logical_ID",
"Condition" : {
  "StringEquals_or_StringLike" : {
    "ResourceType" : [resource_type, ...]
  }
}
]
```

Effect

Determina se as ações que você especifica são negadas ou permitidas no(s) recurso(s) especificado(s). Você pode especificar apenas Deny ou Allow, como:

```
"Effect" : "Deny"
```

Important

Se uma política de pilha inclui instruções de sobreposição (tanto para permitir e negar atualizações em um recurso), uma instrução Deny sempre substituirá uma instrução Allow. Para garantir que um recurso está protegido, use uma instrução Deny para esse recurso.

Ação

Especifica as ações de atualização que são negadas ou permitidas:

Update:Modify

Especifica ações de atualização durante as quais os recursos poderão não ter interrupções ou algumas interrupções enquanto as alterações estão sendo aplicadas. Todos os recursos mantêm seus IDs físicos.

Update:Replace

Especifica ações de atualização durante as quais os recursos são recriados. O CloudFormation cria um novo recurso com as atualizações especificadas e, em seguida, exclui o recurso antigo. Como o recurso é recriado, o ID físico do novo recurso pode ser diferente.

Update:Delete

Especifica ações de atualização durante as quais os recursos são removidos. As atualizações que removem completamente os recursos de um modelo de pilha exigem esta ação.

Atualização:*

Especifica todas as ações de atualização. O asterisco é um caractere curinga que representa todas as ações de atualização.

O exemplo a seguir mostra como especificar apenas as ações de substituição e exclusão:

```
"Action" : ["Update:Replace", "Update:Delete"]
```

Para permitir todas as ações de atualização exceto para uma, use `NotAction`. Por exemplo, para permitir todas as ações de atualização exceto `Update:Delete`, use `NotAction`, como mostrado no exemplo a seguir:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Entidade principal

O elemento `Principal` especifica a entidade que a política se aplica. Este elemento é necessário, mas suporta apenas o caractere curinga (*), o que significa que a política aplica-se a todos [principais](#).

Recurso

Especifica os IDs lógicos dos recursos aos quais a política se aplica. Para especificar os tipos de recursos, use o elemento `Condition`.

Para especificar um único recurso, use seu ID lógico. Por exemplo:

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

Você pode usar um caractere curinga com IDs lógicos. Por exemplo, se você usar um prefixo de ID lógico comum para todos os recursos relacionados, é possível especificar todos eles com um caractere curinga:

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

Você também pode usar um elemento Not com recursos. Por exemplo, para permitir as atualizações em todos os recursos, exceto um, use um elemento NotResource para proteger esse recurso:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Quando você define uma política de pilha, qualquer atualização não explicitamente permitida é negada por padrão. Ao permitir atualizações para todos os recursos, exceto para o recurso ProductionDatabase, você nega atualizações para o recurso ProductionDatabase.

Condições

Especifica o tipo de recurso que a política se aplica. Para especificar os IDs lógicos de recursos específicos, use o elemento Resource.

Você pode especificar um tipo de recurso, como todas as instâncias de banco de dados do RDS e EC2, conforme mostrado no exemplo a seguir:

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Principal" : "*",
      "Action" : "Update:*",
```

```
"Resource" : "*",
"Condition" : {
  "StringEquals" : {
    "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
  }
},
{
  "Effect" : "Allow",
  "Principal" : "*",
  "Action" : "Update:*",
  "Resource" : "*"
}
```

A instrução `Allow` concede permissões de atualização para todos os recursos e a instrução `Deny` nega atualizações para instâncias de banco de dados do RDS e EC2. A declaração `Deny` sempre substitui as ações de permissão.

Você pode usar um caractere curinga com tipos de recursos. Por exemplo, você pode negar permissões de atualização para todos os recursos do Amazon EC2 — como instâncias, security groups e sub-redes — usando um caractere curinga, como mostrado no exemplo a seguir:

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2:*"]
  }
}
```

Você deve usar a condição `StringLike` quando ao usar caracteres curinga.

Configurar uma política de pilha

Você pode usar o console ou a AWS CLI para aplicar uma política de pilha ao criar uma pilha. Você também pode usar a AWS CLI para aplicar uma política de pilha a uma pilha existente. Depois de aplicar uma política de pilha, você não pode removê-la da pilha, mas é possível usar a AWS CLI para modificá-la.

Uma política de pilha se aplica a todos os usuários do CloudFormation que tentam atualizar a pilha. Você não pode associar diferentes políticas de pilha com diferentes usuários.

Para obter mais informações sobre como gravar políticas de pilhas, consulte [Definir uma política de pilha](#).

Para definir uma política de pilha ao criar uma pilha (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Na barra de navegação na parte superior da tela, escolha a Região da AWS na qual a pilha deve ser criada.
3. Na página CloudFormation Stacks (Pilhas do CloudFormation), escolha Criar pilha (Create stack).
4. No assistente Create Stack (Criar pilha), na página Configure stack options (Configurar opções de pilha), expanda a seção Advanced (Avançado) e escolha Stack policy (Política de pilha).
5. Especifique a política da pilha:
 - Para gravar uma política diretamente no console, escolha Enter stack policy (Inserir política de pilha) e digite a política de pilha diretamente no campo de texto.
 - Para usar uma política definida em um arquivo separado, escolha Upload a file (Fazer upload de um arquivo) e Choose file (Escolher arquivo) para selecionar o arquivo que contém a política de pilha.

Para definir uma política de pilha ao criar uma pilha (AWS CLI)

- Use o comando [create-stack](#) com a opção `--stack-policy-body` para digitar em uma política modificada ou a opção `--stack-policy-url` para especificar um arquivo que contém a política.

Para definir uma política de pilha em uma pilha existente (AWS CLI apenas)

- Use o comando [set-stack-policy](#) com a opção `--stack-policy-body` para digitar em uma política modificada ou a opção `--stack-policy-url` para especificar um arquivo que contém a política.

 Note

Para adicionar uma política a uma pilha existente, você deve ter permissão para a ação [SetStackPolicy](#) do CloudFormation.

Atualizar recursos protegidos

Para atualizar recursos protegidos, crie uma política temporária que substitui a política de pilha e permite atualizações nesses recursos. Especifique a política de substituição quando você atualizar a pilha. A política de substituição não altera permanentemente a política de pilha.

Para atualizar recursos protegidos, você deve ter permissão para usar a ação [SetStackPolicy](#) do CloudFormation. Para obter mais informações sobre configuração de permissões do CloudFormation, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Note

Durante uma atualização de pilha, o CloudFormation atualiza recursos que dependem de outros recursos atualizados. Por exemplo, o CloudFormation atualiza um recurso que faz referência a um recurso atualizado. O CloudFormation não faz alterações físicas, como o ID de recursos, para recursos atualizados automaticamente, mas se uma política de pilha é associada a esses recursos, você deve ter permissão para atualizá-las.

Para atualizar um recurso protegido (console)

1. Abra o console do CloudFormation em <https://console.aws.amazon.com/cloudformation>.
2. Selecione a pilha que deseja atualizar, escolha Stack actions (Ações de pilha) e selecione Update stack (Atualizar pilha).
3. Se você não modificou o modelo de pilha, selecione Use current template (Usar modelo atual) e clique em Next (Próximo). Se você modificou o modelo, selecione Replace current template (Substituir modelo atual) e especifique a localização do modelo atualizado na seção Specify template (Especificar modelo):
 - Para um modelo armazenado localmente em seu computador, selecione Upload a template file (Fazer upload de um arquivo de modelo). Escolha Escolher arquivo para navegar até o arquivo e selecioná-lo e, em seguida, clique em Próximo.
 - Para um modelo armazenado em um bucket do Amazon S3, selecione Amazon S3 URL (URL do Amazon S3). Insira ou cole o URL do modelo e, em seguida, clique em Próximo.

Se você tiver um modelo em um bucket com versionamento habilitado, poderá especificar uma versão específica do modelo anexando `?versionId=version-id` ao URL. Para

obter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado](#) no Guia do usuário do Amazon Simple Storage Service.

4. Se o seu modelo contém parâmetros, na página Specify stack details (Especificar detalhes da pilha), insira ou modifique os valores dos parâmetros e escolha Next (Próximo).

O CloudFormation preenche cada parâmetro com o valor que está atualmente definido na pilha, exceto os parâmetros declarados com o atributo NoEcho. É possível usar os valores atuais para esses parâmetros escolhendo Use existing value (Usar valor existente).

Para obter mais informações sobre o uso de NoEcho para ocultar informações confidenciais e sobre o uso de parâmetros dinâmicos para gerenciar segredos, consulte a melhor prática [Não incorporar credenciais em seus modelos](#).

5. Especifique uma política de pilha de substituição.
 - a. Na página Configure stack options (Configurar opções da pilha), na seção Advanced options (Opções avançadas), selecione Stack policy (Política de pilha).
 - b. Selecione Upload a file (Fazer upload de um arquivo).
 - c. Clique em Choose file (Escolher arquivo) e navegue até o arquivo que contém a política de pilha de sobreposição ou digite uma política.
 - d. Escolha Próximo.

A política de substituição deve especificar uma instrução Allow para os recursos protegidos que você deseja atualizar. Por exemplo, para atualizar todos os recursos protegidos, especifique uma política de substituição temporária que permite todas as atualizações:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

 Note

O CloudFormation aplica a política de substituição apenas durante a atualização. A política de substituição não altera permanentemente a política de pilha. Para modificar uma política de pilha, consulte [Modificar uma política de pilha](#).

6. Analise as informações e as alterações da pilha que você enviou.

Verifique se você enviou as informações corretas, como os valores de parâmetro ou modelo de URL correto. Se o seu modelo contém recursos do IAM, escolha I acknowledge that this template may create IAM resources (Eu reconheço que este modelo pode criar recursos do IAM) para especificar que você deseja usar recursos do IAM no modelo. Para obter mais informações, consulte [Confirmar recursos do IAM em modelos do CloudFormation](#).

Na seção Visualizar suas alterações, verifique se o CloudFormation fará todas as alterações que você espera. Por exemplo, verifique que o CloudFormation adiciona, remove e modifica os recursos que você pretende adicionar, remover ou modificar. O CloudFormation gera esta pré-visualização criando um conjunto de alterações para a pilha. Para obter mais informações, consulte [Atualizar pilhas do CloudFormation usando conjuntos de alterações](#).

7. Quando estiver satisfeito com as alterações, clique em Update (Atualizar).

 Note

Nesse ponto, você também tem a opção de visualizar o conjunto de alterações para rever as atualizações propostas mais cuidadosamente. Para fazer isso, clique em View change set (Visualizar conjunto de alterações) em vez de em Update (Atualizar). O CloudFormation mostra o conjunto de alterações gerado com base nas suas atualizações. Quando estiver pronto para executar a atualização da pilha, clique em Execute (Executar).

CloudFormation exibe a página Stack details (Detalhes da pilha) para sua pilha. Sua pilha agora tem um status de UPDATE_IN_PROGRESS. Depois que o CloudFormation tiver concluído com êxito a atualização da pilha, ele definirá o status da pilha como UPDATE_COMPLETE.

Caso a atualização da pilha falhe, o CloudFormation reverterá automaticamente as alterações e definirá o status como UPDATE_ROLLBACK_COMPLETE.

Para atualizar um recurso protegido (AWS CLI)

- Use o comando [update-stack](#) com a opção `--stack-policy-during-update-body` para digitar em uma política modificada ou a opção `--stack-policy-during-update-url` para especificar um arquivo que contém a política.

Note

O CloudFormation aplica a política de substituição apenas durante a atualização. A política de substituição não altera permanentemente a política de pilha. Para modificar uma política de pilha, consulte [Modificar uma política de pilha](#).

Modificar uma política de pilha

Para proteger os recursos adicionais ou para remover a proteção dos recursos, modifique a política da pilha. Por exemplo, quando você adiciona um banco de dados que você deseja proteger a sua pilha, adicione uma instrução Deny para o banco de dados à política de pilha. Para modificar a política, você deve ter permissão para usar a ação [SetStackPolicy](#).

Use a AWS CLI para modificar políticas de pilha.

Para modificar uma política de pilha (AWS CLI)

- Use o comando [set-stack-policy](#) com a opção `--stack-policy-body` para digitar em uma política modificada ou a opção `--stack-policy-url` para especificar um arquivo que contém a política.

Você não pode excluir uma política de pilha. Para remover todas as proteções de todos os recursos, você modifica a política para explicitamente permitir todas as ações em todos os recursos. A política a seguir permite todas as atualizações em todos os recursos:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}  
]  
}
```

Mais exemplos de políticas de pilha

As seguintes políticas de exemplo mostram como impedir atualizações em todos os recursos de pilha, especificar recursos e impedir determinados tipos de atualizações.

Impedir atualizações em todos os recursos de pilha

Para impedir atualizações em todos os recursos de pilha, a seguinte política especifica uma instrução Deny para todas as ações de atualização em todos os recursos.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    }  
  ]  
}
```

Impedir atualizações em um único recurso

A seguinte política nega todas as ações de atualização no banco de dados com o ID lógico MyDatabase. Isso permite todas as ações de atualização em todos os outros recursos de pilha com uma instrução Allow. A instrução Allow não se aplica ao recurso MyDatabase porque a instrução Deny sempre substitui as ações de permissão.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/MyDatabase"   
    },  
  ]  
}
```

```
{
  "Effect" : "Allow",
  "Action" : "Update:*",
  "Principal": "*",
  "Resource" : "*"
}
```

Você pode alcançar o mesmo resultado do exemplo anterior usando um padrão de negação. Quando você define uma política de pilha, o CloudFormation bloqueia qualquer atualização que não é explicitamente permitida. A seguinte política permite atualizações em todos os recursos, exceto no recurso `ProductionDatabase`, que é negado por padrão.

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Important

Não há risco em usar uma negação padrão. Se você tiver uma instrução `Allow` em outro lugar na política (como uma instrução `Allow` que usa um caractere curinga), talvez você conceda inadvertidamente permissão de atualização a recursos que você não pretendia. Como uma negação explícita substitui quaisquer ações de permissão, você pode garantir que um recurso seja protegido usando uma instrução `Deny`.

Impedir atualizações em todas as instâncias de um tipo de recurso

A seguinte política nega todas as ações de atualização no tipo de recurso de instância de banco de dados do RDS. Isso permite todas as ações de atualização em todos os outros recursos de pilha com uma instrução `Allow`. A instrução `Allow` não se aplica ao recurso de instância de banco de dados do RDS porque uma instrução `Deny` sempre substitui as ações de permissão.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Impedir atualizações de substituição em uma instância

A seguinte política nega atualizações que poderiam causar uma substituição da instância com o ID lógico MyInstance. Isso permite todas as ações de atualização em todos os outros recursos de pilha com uma instrução Allow. A instrução Allow não se aplica ao recurso MyInstance porque a instrução Deny sempre substitui as ações de permissão.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}  
]  
}
```

Impedir atualizações em pilhas aninhadas

A seguinte política nega todas as ações de atualização no tipo de recurso de pilha do CloudFormation (pilhas aninhadas). Isso permite todas as ações de atualização em todos os outros recursos de pilha com uma instrução Allow. A declaração Allow não se aplica aos recursos da pilha do CloudFormation porque a declaração Deny sempre substitui as ações permitidas.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal" : "*",  
      "Resource" : "*",  
      "Condition" : {  
        "StringEquals" : {  
          "ResourceType" : ["AWS::CloudFormation::Stack"]  
        }  
      }  
    },  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal" : "*",  
      "Resource" : "*"   
    }  
  ]  
}
```

Proteção de dados no CloudFormation

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS CloudFormation. Conforme descrito nesse modelo, AWS é responsável por proteger a infraestrutura global que executa todas as Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais

informações sobre a privacidade de dados, consulte as [Data Privacy FAQ](#). Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and RGPD](#) no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure os logs de API e atividade do usuário com AWS CloudTrail. Para obter informações sobre como usar as trilhas do CloudTrail para capturar atividades da AWS, consulte [Working with CloudTrail trails](#) no Guia do usuário do AWS CloudTrail.
- Use as soluções de criptografia AWS, juntamente com todos os controles de segurança padrão em Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar a AWS por meio de uma interface de linha de comandos ou de uma API, use um endpoint do FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui trabalhar com a CloudFormation ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia inativa

Seguindo o modelo de responsabilidade compartilhada da AWS, o CloudFormation armazena seus dados criptografados em repouso. Os clientes são responsáveis por definir políticas de criptografia

e armazenamento para dados armazenados em suas contas. Por exemplo, recomendamos habilitar a criptografia em repouso para modelos e outros dados armazenados em buckets do S3 ou tópicos do SNS. Os clientes definem de forma semelhante as configurações de criptografia para quaisquer sistemas de armazenamento de dados provisionados pelo CloudFormation.

Criptografia em trânsito

O CloudFormation usa canais criptografados para comunicações de serviço sob o modelo de responsabilidade compartilhada.

Privacidade do tráfego entre redes

As comunicações do serviço CloudFormation são criptografadas com segurança por padrão entre regiões ou zonas de disponibilidade.

Controle o acesso ao CloudFormation com o AWS Identity and Access Management

Com o AWS Identity and Access Management (IAM), você pode criar usuários do IAM e controlar seu acesso a recursos específicos da sua Conta da AWS. Usando o IAM, você pode controlar o que os usuários podem fazer com o CloudFormation, por exemplo, se podem visualizar modelos de pilhas, criar pilhas ou excluir pilhas.

Além das ações específicas do CloudFormation, você pode gerenciar quais serviços e recursos da AWS estão disponíveis para cada usuário. Assim, você pode controlar quais recursos os usuários podem acessar ao usar o CloudFormation. Por exemplo, você pode especificar quais usuários podem criar instâncias Amazon EC2, encerrar instâncias de banco de dados ou atualizar VPCs. As mesmas permissões serão aplicadas sempre que eles usam o CloudFormation para executar essas ações.

Use as informações das seções a seguir para controlar quem pode acessar o CloudFormation. Também exploraremos como autorizar a criação de recursos do IAM em modelos, como dar às aplicações executadas nas instâncias do EC2 as permissões necessárias e como usar credenciais de segurança temporárias para aumentar a segurança em seu ambiente AWS.

Definir políticas baseadas em identidade do IAM para o CloudFormation

Para dar acesso ao CloudFormation, você precisa criar e atribuir políticas do IAM que concedam às identidades do IAM (como usuários ou perfis) permissão para chamar as ações de API necessárias.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. O CloudFormation é compatível com ações, recursos e chaves de condição específicos.

Se você é novato no IAM, comece por se familiarizar com os elementos de uma política JSON do IAM. Para obter mais informações, consulte [Referência de elemento de política JSON do IAM](#) no Guia do usuário do IAM. Para aprender a criar políticas do IAM, faça o tutorial [Criar e anexar sua primeira política gerenciada pelo cliente](#) na documentação do IAM.

Tópicos

- [Ações de política para o CloudFormation](#)
- [Ações específicas do console para o CloudFormation](#)
- [Recursos de política do CloudFormation](#)
- [Chaves de condição de política do CloudFormation](#)

Ações de política para o CloudFormation

No elemento Action da instrução da política do IAM, você pode especificar qualquer ação de API oferecida pelo CloudFormation. O nome da ação deve começar com o prefixo `cloudformation:` em minúsculas. Por exemplo: `cloudformation:CreateStack`, `cloudformation:CreateChangeSet` e `cloudformation:UpdateStack`.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, da seguinte maneira:

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

Também é possível especificar várias ações usando asteriscos. Por exemplo, é possível especificar todas as ações cujos nomes começam com a palavra `Get`, da seguinte maneira:

```
"Action": "cloudformation:Get*"
```

Para ver uma lista completa das ações associadas ao prefixo de serviço `cloudformation`, consulte [Actions, resources, and condition keys for CloudFormation](#) e [Actions, resources, and condition keys for AWS API Cloud Control](#) na Service Authorization Reference.

Exemplos

O exemplo a seguir mostra uma política de permissões que concede permissões para visualizar pilhas do CloudFormation.

Example 1: um exemplo de política que concede permissões para visualizar pilhas

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStackResources"
    ],
    "Resource": "*"
  }]
}
```

Os usuários que criam ou excluem pilhas precisam ter permissões adicionais de acordo com seus modelos de pilha. Por exemplo, se o modelo descrever uma fila do Amazon SQS, os usuários deverão ter permissões para as ações do CloudFormation e do Amazon SQS, conforme mostrado no exemplo de política a seguir.

Example 2: um exemplo de política que permite as ações de criar e visualizar pilhas, e todas as ações do Amazon SQS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "sqs:*",
      "cloudformation:CreateStack",

```

```
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResources",
        "cloudformation:GetTemplate",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
}]
}
```

Ações específicas do console para o CloudFormation

Os usuários que usam o console do CloudFormation precisam ter permissões adicionais, além das necessárias para usar as APIs da AWS Command Line Interface ou do CloudFormation. Essas permissões adicionais são compatíveis com atributos específicos do console, como uploads de modelos em buckets do Amazon S3 e listas suspensas para tipos de parâmetros específicos da AWS.

Para todas as ações listadas abaixo, conceda permissões a todos os recursos, sem limitá-las a pilhas ou buckets específicos.

A ação a seguir é usada apenas pelo console do CloudFormation e não é documentada na referência de APIs. A ação permite que os usuários façam upload de modelos para buckets do Amazon S3.

- `cloudformation:CreateUploadBucket`

Para carregar modelos, os usuários também precisam ter as seguintes permissões do Amazon S3:

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

Para ver os valores nas listas suspensas de parâmetros para modelos com tipos de parâmetros específicos da AWS, os usuários precisam ter permissões para fazer as chamadas de API `describe` correspondentes. Por exemplo, as seguintes permissões são necessárias quando esses tipos de parâmetros são usados no modelo:

- `ec2:DescribeKeyPairs`: requerida para o tipo de parâmetro `AWS::EC2::KeyPair::KeyName`.
- `ec2:DescribeSecurityGroups`: requerida para o tipo de parâmetro `AWS::EC2::SecurityGroup::Id`.
- `ec2:DescribeSubnets`: requerida para o tipo de parâmetro `AWS::EC2::Subnet::Id`.
- `ec2:DescribeVpcs`: requerida para o tipo de parâmetro `AWS::EC2::VPC::Id`.

Para obter mais informações sobre os tipos de parâmetros específicos da AWS, consulte [Especificar recursos existentes no runtime com tipos de parâmetros fornecidos pelo CloudFormation](#).

Recursos de política do CloudFormation

Em uma instrução de política do IAM, o elemento `Resource` especifica o objeto ou os objetos abrangidos pela instrução. No CloudFormation, cada instrução de política do IAM se aplica aos recursos que você especifica usando os nomes dos recursos da Amazon (ARNs). O formato do ARN específico depende do recurso.

Para obter uma lista dos tipos de recursos do CloudFormation e seus ARNs, consulte [Resource types defined by CloudFormation](#) na Service Authorization Reference. Para saber com quais ações você pode especificar o ARN de cada atributo, consulte [Actions defined by CloudFormation](#).

Você pode especificar ações para uma pilha específica, como mostra o exemplo de política a seguir. Quando fornecer um ARN, substitua o *placeholder text* pelas informações específicas do recurso.

Example 1: um exemplo de política que nega as ações de excluir e atualizar pilha para a pilha especificada

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloudformation:DeleteStack",
        "cloudformation:UpdateStack"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:cloudformation:us-
east-1:111122223333:stack/MyProductionStack/*"
  }
]
}

```

A política acima usa um caractere curinga no fim do nome da pilha para que as ações de excluir e atualizar pilha sejam negadas tanto com o ID da pilha completo (como `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) e tanto com o nome da pilha (como `MyProductionStack`).

Para permitir que transformações `AWS::Serverless` criem um conjunto de alterações, a política deve incluir a permissão ao nível do recurso `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31`, como mostrado na política a seguir.

Example 2: um exemplo de política que permite a ação de criar um conjunto de alterações para a transformação especificada

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/
Serverless-2016-10-31"
    }
  ]
}

```

Chaves de condição de política do CloudFormation

Em uma instrução de política do IAM, você tem a opção de especificar as condições que controlam quando uma política entra em vigor. Por exemplo, você pode definir uma política que permita que os usuários do criem uma pilha apenas quando especificarem uma determinada URL modelo. Você pode definir condições específicas do CloudFormation e condições gerais da AWS, como `DateLessThan`, que determinam quando uma política deixa de vigorar. Para obter mais informações e uma lista de condições gerais da AWS, consulte Condição na [Referência a elementos de políticas do IAM](#), no Guia do usuário do IAM.

Note

Não use a condição `aws:SourceIp` em toda a AWS. O CloudFormation provisiona recursos usando seu próprio endereço IP, e não o endereço IP da solicitação de origem. Por exemplo, quando você cria uma pilha, o CloudFormation faz solicitações do seu endereço IP para executar uma instância do Amazon EC2 ou para criar um bucket do Amazon S3, e não do endereço IP da chamada `CreateStack` ou do comando `create-stack`.

A lista a seguir descreve as condições específicas do CloudFormation. Essas condições são aplicadas apenas quando os usuários criam ou atualizam pilhas:

`cloudformation:ChangeSetName`

O nome de um conjunto de alterações do CloudFormation que você deseja associar a uma política. Use essa condição para controlar quais conjuntos de alterações os usuários do podem executar ou excluir.

`cloudformation:ImportResourceTypes`

Os tipos de recursos do modelo que você deseja associar a uma política, como `AWS::EC2::Instance`. Use essa condição para controlar com quais tipos de recursos os usuários do podem trabalhar quando importarem recursos para uma pilha. Essa condição é verificada em relação aos tipos de recursos que os usuários declaram no parâmetro `ResourcesToImport`, que atualmente tem suporte apenas para solicitações da AWS CLI e da API. Ao usar esse parâmetro, especifique todos os tipos de recursos que deseja que os usuários controlem durante as operações de importação. Para obter mais informações sobre o parâmetro `ResourcesToImport`, consulte a ação [CreateChangeSet](#) na Referência de APIs do AWS CloudFormation.

Para obter uma lista de `ResourcesToImport` possíveis, consulte [Suporte a tipos de recursos](#).

Use a convenção de nomenclatura de recursos de três partes para especificar com quais tipos de recursos os usuários podem trabalhar, de todos os recursos de uma organização até um tipo de recurso individual.

`organization::*`

Especifique todos os tipos de recursos de uma determinada organização.

`organization::service_name::*`

Especifique todos os tipos de recurso do serviço especificado dentro de uma determinada organização.

`organization::service_name::resource_type`

Especifique um tipo de recurso específico.

Por exemplo:

`AWS::*`

Especifique todos os tipos de recursos da AWS compatíveis.

`AWS::service_name::*`

Especifique todos os recursos com suporte a um produto da AWS específico.

`AWS::service_name::resource_type`

Especifique um tipo de recurso da AWS específico, como `AWS::EC2::Instance` (todas as instâncias do EC2).

`cloudformation:ResourceTypes`

Os tipos de recursos modelo, como `AWS::EC2::Instance`, que você deseja associar a uma política. Use essa condição para controlar com quais tipos de recursos os usuários do podem trabalhar ao criar ou atualizar uma pilha. Essa condição é verificada em relação aos tipos de recursos que os usuários declaram no parâmetro `ResourceTypes`, que atualmente tem suporte apenas para solicitações da AWS CLI e da API. Ao usar esse parâmetro, os usuários devem especificar todos os tipos de recursos que estão em seu modelo. Para obter mais informações sobre o parâmetro `ResourceTypes`, consulte a ação [CreateStack](#) na Referência de APIs do AWS CloudFormation.

Para obter uma lista dos tipos de recursos, consulte o [Guia de referência de modelos do CloudFormation](#).

Use a convenção de nomenclatura de recursos de três partes para especificar com quais tipos de recursos os usuários podem trabalhar, de todos os recursos de uma organização até um tipo de recurso individual.

*organization::**

Especifique todos os tipos de recursos de uma determinada organização.

*organization::service_name::**

Especifique todos os tipos de recurso do serviço especificado dentro de uma determinada organização.

organization::service_name::resource_type

Especifique um tipo de recurso específico.

Por exemplo:

*AWS::**

Especifique todos os tipos de recursos da AWS compatíveis.

*AWS::service_name::**

Especifique todos os recursos com suporte a um produto da AWS específico.

AWS::service_name::resource_type

Especifique um tipo de recurso da AWS específico, como *AWS::EC2::Instance* (todas as instâncias do EC2).

*Alexa::ASK::**

Especifique todos os tipos de recurso no Kit de habilidades do Alexa.

Alexa::ASK::Skill

Especifique o tipo de recurso [Alexa::ASK::Skill](#) individual.

*Custom::**

Especifique todos os recursos personalizados.

Para obter mais informações, consulte [Crie uma lógica de provisionamento personalizada com recursos personalizados](#).

Custom::*resource_type*

Especifique um tipo de recurso personalizado específico.

Para obter mais informações, consulte [Crie uma lógica de provisionamento personalizada com recursos personalizados](#).

cloudformation:RoleARN

O Nome de recurso da Amazon (ARN) de uma função de serviço do IAM que você deseja associar a uma política. Use essa condição para controlar qual serviço os usuários do podem usar ao trabalhar com pilhas ou conjuntos de alterações.

cloudformation:StackPolicyUrl

O URL de uma política de pilha Amazon S3 que você deseja associar a uma política. Use essa condição para controlar quais políticas de pilha os usuários do podem associar a uma pilha durante uma ação de criação ou atualização de pilha. Para obter mais informações sobre políticas de pilhas, consulte [Impedir atualizações nos recursos de pilha](#).

 Note

Para garantir que os usuários possam apenas criar ou atualizar pilhas com as políticas de pilhas que você carregou, defina o bucket do S3 como somente leitura para esses usuários.

cloudformation:TemplateUrl

O URL de um modelo do Amazon S3 que você deseja associar a uma política. Use essa condição para controlar quais modelos os usuários do podem usar ao criar ou atualizar pilhas.

 Note

Para garantir que os usuários possam apenas criar ou atualizar pilhas com os modelos que você carregou, defina o bucket do S3 como somente leitura para esses usuários.

 Note

As seguintes condições específicas de CloudFormation se aplicam aos parâmetros da API com o mesmo nome:

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`
- `cloudformation:StackPolicyUrl`
- `cloudformation:TemplateUrl`

Por exemplo, `cloudformation:TemplateUrl` só se aplica ao parâmetro `TemplateUrl` para as APIs `CreateStack`, `UpdateStack` e `CreateChangeSet`.

Para obter exemplos de políticas do IAM que usam chaves de condição, consulte [Exemplos de políticas baseadas em identidade do IAM para o CloudFormation](#).

Confirmar recursos do IAM em modelos do CloudFormation

Antes que você possa criar uma pilha, o CloudFormation deve validar seu modelo. Durante a validação, o CloudFormation verifica os recursos do IAM que seu modelo pode criar. Os recursos do IAM, como um usuário com acesso completo, podem acessar e modificar qualquer recurso na sua . Conta da AWS. Portanto, sugerimos que você examine as permissões associadas a cada recurso do IAM antes de prosseguir, para não criar recursos acidentalmente com permissões escalonadas. Para garantir que fez isso, você deve confirmar que o modelo contém esses recursos, fornecendo ao CloudFormation as capacidades especificadas antes de criar a pilha.

Você pode confirmar as capacidades dos modelos do CloudFormation usando o console do CloudFormation, a AWS Command Line Interface (AWS CLI) ou a API:

- No console do CloudFormation, na página Configurar opções da pilha dos assistentes de criação ou de atualização de pilhas, escolha Eu reconheço que este modelo pode criar recursos do IAM.
- Na AWS CLI, ao usar os comandos [create-stack](#) e [update-stack](#), especifique o valor `CAPABILITY_IAM` ou `CAPABILITY_NAMED_IAM` para a opção `--capabilities`. Se seu modelo incluir recursos do IAM, você poderá especificar uma dessas capacidades. Se seu modelo incluir nomes personalizados para recursos do IAM, você deverá especificar `CAPABILITY_NAMED_IAM`.
- Na API, ao usar as ações [CreateStack](#) e [UpdateStack](#), especifique `Capabilities.member.1=CAPABILITY_IAM` ou `Capabilities.member.1=CAPABILITY_NAMED_IAM`. Se seu modelo incluir recursos do IAM, você poderá especificar uma dessas capacidades. Se seu modelo incluir nomes personalizados para recursos do IAM, você deverá especificar `CAPABILITY_NAMED_IAM`.

 Important

Se seu modelo contiver recursos do IAM nomeados personalizados, não crie várias pilhas reutilizando o mesmo modelo. Os recursos do IAM devem ser globalmente exclusivos na sua conta. Se você usar o mesmo modelo para criar várias pilhas em diferentes regiões, suas pilhas poderão compartilhar os mesmos recursos do IAM, em vez de cada uma ter um recurso exclusivo. Recursos compartilhados entre pilhas podem ter consequências acidentais das quais não é possível se recuperar. Por exemplo, se você excluir ou atualizar recursos compartilhados do IAM em uma pilha, você modificará acidentalmente os recursos de outras pilhas.

Gerenciar credenciais para aplicações executadas em instâncias do Amazon EC2

Se você tiver uma aplicação executada em uma instância do Amazon EC2 e precisar fazer solicitações a recursos da AWS, como buckets do Amazon S3 ou uma tabela do DynamoDB, essa aplicação precisará de credenciais de segurança da AWS. No entanto, a distribuição e a inserção de credenciais de segurança de longo prazo em cada instância que você executa é um desafio e um risco potencial à segurança. Em vez de usar credenciais de longo prazo, como credenciais de usuário do IAM, recomendamos que você crie uma função do IAM que seja associada a uma instância do Amazon EC2 quando a instância for iniciada. Um aplicativo pode obter credenciais de segurança temporárias na instância Amazon EC2. Você não precisa inserir credenciais de longo prazo na instância. Além disso, para facilitar o gerenciamento de credenciais, você pode especificar apenas uma única função para várias instâncias Amazon EC2. Você não precisa criar credenciais exclusivas para cada instância.

Para obter um trecho de modelo que mostra como executar uma instância com uma função, consulte [Exemplos de modelos de função do IAM](#).

 Note

As aplicações em instâncias que usam credenciais de segurança temporárias podem chamar qualquer ação do CloudFormation. No entanto, como o CloudFormation interage com muitos outros serviços da AWS, é necessário verificar se todos os serviços que você deseja usar oferecem suporte a credenciais de segurança temporárias. Para obter uma lista de serviços

que aceitam credenciais de segurança temporárias, consulte os [serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Conceder acesso temporário (acesso federado)

Em alguns casos, você pode desejar conceder aos usuários sem credenciais da AWS acesso temporário à sua Conta da AWS. Em vez de criar e excluir credenciais de longo prazo sempre que você quiser conceder acesso temporário, use o AWS Security Token Service (AWS STS). Por exemplo, você pode usar funções do IAM. Em uma função do IAM, você pode criar programaticamente e distribuir muitas credenciais de segurança temporárias (que incluem uma chave de acesso, uma chave de acesso secreta e um token de segurança). Essas credenciais têm vida útil limitada e, portanto, não podem ser usadas para acessar sua Conta da AWS depois que expiram. Você também pode criar vários perfis do IAM para conceder diferentes níveis de permissões a usuários individuais. Perfis do IAM são úteis para cenários como identidades federadas e logon único.

Uma identidade federada é uma identidade distinta que você pode usar entre vários sistemas. Para usuários corporativos com um sistema de identidade estabelecido on-premises (como o LDAP ou o Active Directory), você pode tratar de toda a autenticação com seu sistema de identidade local. Quando um usuário é autenticado, você fornece credenciais de segurança temporárias a partir da função ou do usuário do IAM apropriado. Por exemplo, é possível criar um perfil administrators e um perfil developers em que os administradores têm acesso total à conta da AWS e os desenvolvedores têm permissões para trabalhar apenas com pilhas do CloudFormation. Depois que um administrador é autenticado, ele é autorizado a obter credenciais de segurança temporárias do perfil administrators. No entanto, os desenvolvedores podem obter credenciais de segurança temporárias apenas do perfil developers.

Você também pode conceder aos usuários federados acesso ao Console de gerenciamento da AWS. Depois que os usuários se autenticam com o sistema de identidade on-premises, você pode construir programaticamente um URL temporário que forneça acesso direto ao Console de gerenciamento da AWS. Quando os usuários usam o URL temporário, eles não precisam fazer login na AWS porque já foram autenticados (autenticação única). Além disso, como o URL é construído a partir das credenciais de segurança temporárias dos usuários, as permissões que estão disponíveis com essas credenciais determinam quais permissões os usuários têm no Console de gerenciamento da AWS.

Você pode usar várias diferentes APIs do AWS STS para gerar credenciais de segurança temporárias. Para obter mais informações sobre qual API usar, consulte [Comparar credenciais do AWS STS](#) no Guia do usuário do IAM.

 Important

Você não pode trabalhar com o IAM ao usar credenciais de segurança temporárias que foram geradas a partir da API `GetFederationToken`. Em vez disso, se você precisar trabalhar com o IAM, use credenciais de segurança temporárias de uma função.

O CloudFormation interage com muitos outros serviços da AWS. Ao usar credenciais de segurança temporárias com o CloudFormation, verifique se todos os serviços que deseja usar oferecem suporte a credenciais de segurança temporárias. Para obter uma lista de serviços que aceitam credenciais de segurança temporárias, consulte os [serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Para obter mais informações, consulte os seguintes recursos relacionados no Guia do usuário do IAM:

- [Cenários comuns para credenciais temporárias](#)
- [Habilitar o acesso do intermediador de identidades personalizado ao console da AWS](#)

Exemplos de políticas baseadas em identidade do IAM para o CloudFormation

Por padrão, os usuários e os perfis não têm permissão para criar nem modificar recursos do CloudFormation. Eles também não podem executar tarefas usando o Console de gerenciamento da AWS, a AWS Command Line Interface (AWS CLI) ou a API da AWS. Para conceder aos usuários permissões para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis. Para obter mais informações, consulte [Definir políticas baseadas em identidade do IAM para o CloudFormation](#).

Os exemplos a seguir mostram as instruções de política que você pode usar para conceder ou negar permissões para usar uma ou mais ações do CloudFormation.

Tópicos

- [Exigir um URL de modelo específico](#)
- [Negar todas as operações de importação do CloudFormation](#)
- [Permitir operações de importação para tipos de recursos específicos](#)
- [Negar recursos do IAM em modelos de pilha](#)
- [Permitir a criação de pilhas com tipos de recursos específicos](#)
- [Controlar o acesso com base nas ações de API de mutação de recurso](#)
- [Restringir operações de conjunto de pilhas com base na região e nos tipos de recursos](#)
- [Permitir todas as operações do gerador de IaC](#)

Exigir um URL de modelo específico

O exemplo de política a seguir concede aos usuários permissões para usar apenas o URL do modelo *<https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template>* para criar ou atualizar uma pilha.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudformation:TemplateUrl": [
            "https://s3.amazonaws.com/amzn-s3-demo-bucket/
test.template"
          ]
        }
      }
    }
  ]
}
```

Negar todas as operações de importação do CloudFormation

O exemplo de política a seguir concede permissões para realizar todas as operações do CloudFormation, exceto as operações de importação.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllStackOperations",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    },
    {
      "Sid": "DenyImport",
      "Effect": "Deny",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    }
  ]
}
```

Permitir operações de importação para tipos de recursos específicos

O exemplo de política a seguir concede permissões para todas as operações de pilha, e também para operações de importação apenas nos recursos especificados (neste exemplo, `AWS::S3::Bucket`).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImport",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:ImportResourceTypes": [
            "AWS::S3::Bucket"
          ]
        }
      }
    }
  ]
}
```

Negar recursos do IAM em modelos de pilha

O exemplo de política a seguir permite que os usuários criem pilhas, mas negará as solicitações se o modelo da pilha incluir algum recurso do serviço IAM. A política também requer que os usuários especifiquem o parâmetro `ResourceTypes`, que está disponível apenas para solicitações de AWS CLI e de API. Essa política usa instruções de negação explícita de forma que se qualquer outra política conceder permissões adicionais, essa política sempre permanecerá em vigor (uma instrução de negação explícita sempre substitui uma instrução de permissão explícita).

JSON

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

```

    },
    {
      "Effect" : "Deny",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "ForAnyValue:StringLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        }
      }
    },
  ],
  {
    "Effect": "Deny",
    "Action" : [ "cloudformation:CreateStack" ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "cloudformation:ResourceTypes": "true"
      }
    }
  }
]
}

```

Permitir a criação de pilhas com tipos de recursos específicos

A política a seguir é semelhante ao exemplo anterior. A política concede aos usuários permissões para criar uma pilha, a menos que o modelo da pilha inclua algum recurso do serviço IAM. Ela também requer que os usuários especifiquem o parâmetro `ResourceTypes`, que está disponível apenas para solicitações de AWS CLI e de API. Essa política é mais simples, mas não usa instruções de negação explícita. Outras políticas, que concedem permissões adicionais, podem substituir essa política.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],

```

```

    "Resource" : "*",
    "Condition" : {
      "ForAllValues:StringNotLikeIfExists" : {
        "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
      },
      "Null":{
        "cloudformation:ResourceTypes": "false"
      }
    }
  }
]
}

```

Controlar o acesso com base nas ações de API de mutação de recurso

O exemplo de política a seguir concede permissões para filtrar o acesso pelo nome de uma ação de API de mutação de recurso. Isso é usado para controlar quais APIs os usuários do IAM podem usar para adicionar ou remover tags de uma pilha ou um conjunto de pilhas. A operação usada para adicionar ou remover tags deve ser adicionada como valor para a chave de condição. A política a seguir concede as permissões de TagResource e UntagResource à operação de mutação CreateStack.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  ]
}

```

```
}
  }
}
```

Restringir operações de conjunto de pilhas com base na região e nos tipos de recursos

O exemplo de política a seguir concede permissões para conjuntos de pilhas gerenciados pelo serviço. Um usuário com essa política só pode executar operações em conjuntos de pilhas com modelos que contêm tipos de recurso do Amazon S3 (`AWS::S3::*`) ou o tipo de recurso `AWS::SES::ConfigurationSet`. Se o usuário entrou na conta de gerenciamento da organização com o ID 123456789012, ele também poderá executar operações apenas em conjuntos de pilhas que tenham como destino a UO com o ID `ou-1fsfsrsdsfrewr` e apenas no conjunto de pilhas com o ID `stack-set-id` que tenha como destino a Conta da AWS com o ID `987654321012`.

As operações do conjunto de pilha falharão se o modelo de conjunto de pilhas contiver tipos de recursos diferentes dos especificados na política ou se os destinos de implementação forem OU ou IDs de conta diferentes dos especificados na política para as contas de gerenciamento e conjuntos de pilhas correspondentes.

Essas restrições de política só se aplicam quando as operações de conjunto de pilha têm como destino as regiões `us-east-1`, `us-west-2` ou `eu-west-2` Regiões da AWS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/*",
        "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
        "arn:aws:cloudformation:us-west-2:111122223333:type/resource/AWS-SES-ConfigurationSet",
        "arn:aws:cloudformation:*:111122223333:stackset-target/*ou-1fsfsrsdsfrewr",

```

```

        "arn:aws:cloudformation:*:111122223333:stackset-target/stack-set-id/444455556666"
      ],
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:TargetRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    }
  ]
}

```

Permitir todas as operações do gerador de IaC

A política a seguir permite acesso às ações do CloudFormation relacionadas à varredura de recursos e ao gerenciamento de modelos do gerador de IaC. A primeira instrução concede permissões para descrever, listar e iniciar varreduras de recursos. Também permite acesso às permissões adicionais necessárias (`cloudformation:GetResource`, `cloudformation:ListResources` e `cloudformation:ListTypes`) que permitem que o gerador de IaC recupere informações sobre os recursos e os tipos de recursos disponíveis. A segunda instrução concede permissões totais para criar, excluir, descrever, listar e atualizar os modelos gerados.

Você também deve conceder permissões de leitura nos serviços da AWS de destino a qualquer pessoa que vá fazer varreduras de recursos com o gerador de IaC. Para obter mais informações, consulte [Permissões do IAM necessárias para varrer recursos](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ResourceScanningOperations",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeResourceScan",

```

```

        "cloudformation:GetResource",
        "cloudformation:ListResources",
        "cloudformation:ListResourceScanRelatedResources",
        "cloudformation:ListResourceScanResources",
        "cloudformation:ListResourceScans",
        "cloudformation:ListTypes",
        "cloudformation:StartResourceScan"
    ],
    "Resource": "*"
},
{
    "Sid": "TemplateGeneration",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateGeneratedTemplate",
        "cloudformation>DeleteGeneratedTemplate",
        "cloudformation:DescribeGeneratedTemplate",
        "cloudformation:GetResource",
        "cloudformation:GetGeneratedTemplate",
        "cloudformation:ListGeneratedTemplates",
        "cloudformation:UpdateGeneratedTemplate"
    ],
    "Resource": "*"
}
]
}

```

AWS Políticas gerenciadas pela do AWS CloudFormation

Uma política gerenciada pela AWS é uma política autônoma criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns e permitir a atribuição de permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para casos de uso específicos, por estarem disponíveis para uso por todos os clientes da AWS. Recomendamos que você reduza ainda mais as permissões definindo as [políticas gerenciadas pelo cliente](#) que são específicas para seus casos de uso.

Não é possível alterar as permissões definidas em políticas gerenciadas pela AWS. Caso a AWS faça atualizações nas permissões estabelecidas em uma política gerenciada pela AWS, estas mudanças impactarão todas as identidades das entidades principais (usuários, grupos e funções)

vinculadas à esta política. A AWS é mais propensa a atualizar uma política gerenciada pela AWS durante o lançamento de um novo serviço da AWS ou quando novas operações de API estiverem disponíveis para serviços existentes.

Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Manual do usuário do IAM.

Política gerenciada pela AWS: AWSCloudFormationFullAccess

Você pode anexar `AWSCloudFormationFullAccess` aos seus usuários, grupos e funções.

Esta política concede permissões para ter acesso total aos recursos e às ações do CloudFormation.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `cloudformation`: permite que as entidades principais executem todas as ações do CloudFormation em todos os recursos.

Para visualizar as permissões dessa política, consulte [AWSCloudFormationFullAccess](#) no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AWSCloudFormationReadOnlyAccess

Você pode anexar `AWSCloudFormationReadOnlyAccess` aos seus usuários, grupos e funções.

Essa política concede permissões que oferecem acesso somente leitura aos recursos e ações do CloudFormation.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `cloudformation`: permite que as entidades principais executem ações somente leitura do CloudFormation, como descrever pilhas, listar recursos e visualizar modelos, mas não permite criar, atualizar ou excluir pilhas.

Para visualizar as permissões dessa política, consulte [AWSCloudFormationReadOnlyAccess](#) no Guia de referência de políticas gerenciadas pela AWS.

Atualizações do CloudFormation nas políticas gerenciadas pela AWS

Confira os detalhes sobre as atualizações em políticas gerenciadas pela AWS do CloudFormation desde que esse serviço começou a monitorar essas alterações. Para obter alertas automáticos sobre as alterações feitas nesta página, inscreva-se no feed RSS na página de histórico de documentos do CloudFormation.

Alteração	Descrição	Data
AWSCloudFormationReadOnlyAccess : atualização de uma política existente	O CloudFormation adicionou novas permissões para permitir ações de <code>cloudformation:BatchDescribe*</code> para operações de descrição em lote.	30 de janeiro de 2026
AWSCloudFormationReadOnlyAccess : atualização de uma política existente	O CloudFormation adicionou novas permissões para permitir ações de <code>cloudformation:Detect*</code> para recursos de detecção de desvio de pilha.	13 de novembro de 2019
AWSCloudFormationReadOnlyAccess : atualização de uma política existente	O CloudFormation adicionou novas permissões para permitir ações de <code>cloudformation:EstimateTemplateCost</code> , <code>cloudformation:Get*</code> , e <code>cloudformation:ValidateTemplate</code> .	2 de novembro de 2017
AWSCloudFormationFullAccess : nova política	O CloudFormation adicionou uma nova política gerenciada pela AWS que concede acesso total às ações e aos recursos do CloudFormation.	26 de julho de 2019

Alteração	Descrição	Data
AWSCloudFormationReadOnlyAccess : atualização de uma política existente	O CloudFormation adicionou novas permissões para permitir ações de <code>cloudformation:DetectStackDrift</code> e <code>cloudformation:DetectStackResourceDrift</code> para detecção de desvio de pilha.	6 de fevereiro de 2019
AWSCloudFormationReadOnlyAccess : nova política	O CloudFormation adicionou uma nova política gerenciada pela AWS que concede acesso somente leitura às ações e aos recursos do CloudFormation.	6 de fevereiro de 2015
O CloudFormation começou a rastrear alterações	O CloudFormation começou a rastrear alterações em suas políticas gerenciadas pela AWS.	6 de fevereiro de 2015

Função de serviço do CloudFormation

Um perfil de serviço é um perfil do AWS Identity and Access Management (IAM) que permite que o CloudFormation faça chamadas a recursos em uma pilha em seu nome. Você pode especificar um perfil do IAM que permita que o CloudFormation crie, atualize ou exclua seus recursos da pilha. Por padrão, o CloudFormation usa uma sessão temporária que ele gera a partir de suas credenciais de usuário para operações de pilha. Se você especificar uma função de serviço, o CloudFormation usará as credenciais da função.

Use uma função de serviço para especificar explicitamente as ações que o CloudFormation pode realizar, que podem nem sempre ser sempre as mesmas ações que você ou outros usuários podem realizar. Por exemplo, você pode ter privilégios administrativos, mas pode limitar o acesso do CloudFormation a apenas ações do Amazon EC2.

Você cria o perfil de serviço e a respectiva política de permissão com o serviço do IAM. Para obter mais informações sobre a criação de um perfil de serviço, consulte [Criar um perfil para delegar permissões a um serviço da AWS](#) no Guia do usuário do IAM. Especifique o CloudFormation (`cloudformation.amazonaws.com`) como o serviço que pode assumir a função.

Para associar uma função de serviço a uma pilha, especifique a função ao criar a pilha. Para obter detalhes, consulte [Configurar opções da pilha](#). Você também pode alterar o perfil de serviço ao atualizar a pilha no console ou [DeleteStack](#) a pilha via API. Antes de especificar um perfil de serviço, certifique-se de ter permissão para aprová-la (`iam:PassRole`). A permissão `iam:PassRole` especifica quais perfis você pode usar. Para obter mais informações, consulte [Conceder permissões ao usuário para passar um perfil para um serviço da AWS](#) no Guia do usuário do IAM.

Important

Quando você especifica uma função de serviço, o CloudFormation sempre usa essa função para todas as operações que serão executadas nessa pilha. Não é possível remover uma função de serviço associada a uma pilha depois que ela é criada. Outros usuários com permissão para executar operações nessa pilha podem usar esse perfil, independentemente de terem a permissão `iam:PassRole` ou não. Se o perfil inclui permissões que o usuário não precisa, você pode ampliar involuntariamente as permissões de um usuário. Certifique-se de que a função concede o menor privilégio. Para obter mais informações, consulte [Aplicar permissões de privilégio mínimo](#), no Guia do usuário do IAM.

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos o uso das chaves de contexto de condição global [aws:SourceArn](#) e [aws:SourceAccount](#) em políticas de recursos para limitar as permissões que o CloudFormation concede a outro serviço para um recurso específico, como uma extensão do CloudFormation. Use

`aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Certifique-se de que o valor de `aws:SourceArn` seja um ARN do recurso que o CloudFormation armazena.

A maneira mais eficaz de se proteger do problema 'confused deputy' é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:cloudformation:*:123456789012:*`.

Se o valor `aws:SourceArn` não contiver o ID da conta, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` no CloudFormation para evitar o problema confused deputy.

Exemplo de política de confiança que usa as chaves de condição **aws:SourceArn** e **aws:SourceAccount**

Para serviços de registro, o CloudFormation faz chamadas para o AWS Security Token Service (AWS STS) para assumir um perfil de serviço em sua conta. Este perfil é configurado para `ExecutionRoleArn` na operação [RegisterType](#) e definido como `LogRoleArn` na operação [LoggingConfig](#). Para obter mais informações, consulte [Configure um perfil de execução com permissões do IAM e uma política de confiança para acesso público à extensão](#).

Este exemplo de política de confiança de perfil usa instruções de condição para limitar o recurso `AssumeRole` no perfil de serviço somente para as ações na extensão do CloudFormation especificada na conta indicada. As condições `aws:SourceArn` e `aws:SourceAccount` são avaliadas de forma independente. Qualquer solicitação para usar o perfil de serviço deve atender às duas condições.

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "resources.cloudformation.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/
resource/Organization-Service-Resource"
      }
    }
  }
]
```

Mais informações

Para obter exemplos de políticas que usam as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` para um perfil de serviço usado pelo StackSets, consulte [Configurar chaves globais para atenuar problemas de confused deputy](#).

Para obter mais informações, consulte [Atualizar uma política de confiança de perfil](#) no Guia do usuário do IAM.

Solicitações de sessões de acesso direto (FAS) e avaliação de permissões

Ao criar, atualizar e excluir pilhas do CloudFormation, os usuários têm a opção de especificar um ARN de perfil do IAM. Se nenhum perfil for fornecido, o CloudFormation usará seu mecanismo de serviço padrão para interagir com outros serviços da AWS. Nesse cenário, o chamador deve ter as permissões necessárias para os recursos sendo gerenciados. Alternativamente, quando um usuário fornecer seu próprio perfil do IAM, o CloudFormation assumirá esse perfil para realizar interações de serviço em nome dele.

Quer o usuário usuário forneça ou não um perfil do IAM, o CloudFormation gerará um novo token de FAS com escopo reduzido para cada operação de recurso. Portanto, [as chaves de condição relacionadas a FAS](#), incluindo `aws:ViaAWSService`, são preenchidas em ambos os cenários.

O uso de FAS afeta o modo como as políticas do IAM são avaliadas durante as operações do CloudFormation. Ao criar uma pilha com um modelo que inclui recursos afetados pelas chaves de condição relacionadas a FAS, podem ocorrer negações de permissão.

Exemplo de política do IAM

Considere a política do IAM a seguir. O Statement2 impedirá consistentemente a criação de um recurso `AWS::KMS::Key` no CloudFormation. A restrição será aplicada de modo consistente, seja ou não fornecido o perfil do IAM durante a operação da pilha. Isso ocorre porque a chave de condição `aws:ViaAWSService` é sempre definida como `true` devido ao uso de FAS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Statement2",
      "Effect": "Deny",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    }
  ]
}
```

```

    }
  }
}

```

Exemplo de modelo de pilha

Por exemplo, quando um usuário cria uma pilha com o exemplo de modelo a seguir, `aws:ViaAWSService` é definido como `true`, e as permissões do perfil serão substituídas pela política de FAS. A criação da pilha será afetada pelo `Statement2` da política do IAM, que nega a ação `CreateKey`. Isso resulta em um erro de permissão negada.

```

Resources:
  myPrimaryKey:
    Type: AWS::KMS::Key
    Properties:
      Description: An example multi-Region primary key
      KeyPolicy:
        Version: '2012-10-17'
        Id: key-default-1
        Statement:
          - Sid: Enable IAM User Permissions
            Effect: Allow
            Principal:
              AWS: !Join
                - ''
                - - 'arn:aws:iam::'
                  - !Ref AWS::AccountId
                  - ':root'
            Action: kms:*
            Resource: '*'

```

Para obter mais informações sobre FAS, consulte [Sessões de acesso direto](#) no Guia do usuário do IAM.

Note

A maioria dos recursos adere a esse comportamento. Porém, se você tiver um sucesso ou uma falha inesperada ao criar, atualizar ou excluir um recurso, e a política do IAM incluir

chaves de condição relacionadas a FAS, provavelmente o recurso em questão pertencerá a um pequeno subconjunto de recursos que não seguem esse padrão.

Registrar em log chamadas de API do CloudFormation com o AWS CloudTrail

O CloudFormation é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, perfil ou serviço da AWS no CloudFormation. O CloudTrail captura todas as chamadas de API para o CloudFormation como eventos, incluindo as chamadas do console do CloudFormation e chamadas de código para APIs do CloudFormation. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail a um bucket do Amazon S3, incluindo eventos do CloudFormation. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Ao usar as informações coletadas pelo CloudTrail, é possível determinar a solicitação que foi feita ao CloudFormation, o endereço IP do qual a solicitação foi feita, quem a fez e quando ela foi feita, além de outros detalhes.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

Tópicos

- [Informações do CloudFormation no CloudTrail](#)
- [Noções básicas sobre entradas de arquivos de log do CloudFormation](#)

Informações do CloudFormation no CloudTrail

O CloudTrail é habilitado em sua conta AWS ao criá-la. Quando ocorre atividade no CloudFormation, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS no Event history (Histórico de eventos). É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para saber mais, consulte [Viewing events with CloudTrail event history](#).

Para obter um registro de eventos em andamento na sua conta da AWS, incluindo eventos do CloudFormation, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da . A trilha registra logs de eventos de todas as Regiões na AWS divisória e entrega os arquivos do log para o bucket Amazon S3 especificado. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configuração notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [Receber arquivos de log do CloudTrail de várias contas](#)

Todas as ações do CloudFormation são registradas em log pelo CloudTrail e estão documentadas na [Referência de APIs do AWS CloudFormation](#). Por exemplo, as chamadas para as seções `CreateStack`, `DeleteStack` e `ListStacks` geram entradas nos arquivos de log do CloudTrail.

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 com credenciais de usuário-raiz ou usuário do 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 serviço da AWS.

Para saber mais, consulte [Elemento `userIdentity` do CloudTrail](#).

Noções básicas sobre entradas de arquivos de log do CloudFormation

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a operação solicitada, a data e a hora da operação, os parâmetros de solicitação etc. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública, portanto, não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a operação `CreateStack`. A operação foi feita por um usuário do IAM denominado Alice.

Note

Apenas os nomes de chaves de parâmetros de entrada são registrados no log. Valores de parâmetros não são registrados no log.


```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:02:43Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "CreateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "tags": [
      {
        "key": "test",
        "value": "tag"
      }
    ],
    "stackName": "my-test-stack",
    "disableRollback": true,
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ]
  },
  "responseElements": {
    "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/a38e6a60-b397-11e3-b0fc-08002755629e"
  },
  "requestID": "9f960720-b397-11e3-bb75-a5b75389b02d",
  "eventID": "9bf6cfb8-83e1-4589-9a70-b971e727099b"
}
```

O exemplo a seguir mostra que Alice chamou a operação `UpdateStack` na pilha `my-test-stack`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmnopq",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:04:29Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "UpdateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ]
  },
  "stackName": "my-test-stack"
},
"responseElements": {
  "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/a38e6a60-b397-11e3-b0fc-08002755629e"
},
"requestID": "def0bf5a-b397-11e3-bb75-a5b75389b02d",
"eventID": "637707ce-e4a3-4af1-8edc-16e37e851b17"
}
```

O exemplo a seguir mostra que Alice chamou a operação `ListStacks`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
```

```

    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "ListStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
  "eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}

```

O exemplo a seguir mostra que Alice chamou a operação `DescribeStacks` na pilha `my-test-stack`.

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
}

```

```
"eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

O exemplo a seguir mostra que Alice chamou a operação DeleteStack na pilha my-test-stack.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DeleteStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "42dae739-b398-11e3-bb75-a5b75389b02d",
  "eventID": "4965eb38-5705-4942-bb7f-20ebe79aa9aa"
}
```

Segurança da infraestrutura no CloudFormation

Por ser um serviço gerenciado, o AWS CloudFormation é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte [Segurança na Nuvem AWS](#). Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança da infraestrutura, consulte [Proteção de Infraestrutura](#) em Pilar de Segurança: AWS Well-Architected Framework.

Você usa CloudFormation chamadas de API publicadas pela para acessar AWS o por meio da rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.

- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Você chamar essas operações de API de qualquer local da rede, mas o CloudFormation não é suportado por políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar políticas do CloudFormation para controlar o acesso de Amazon Virtual Private Cloud (Amazon VPC) endpoints ou de VPCs específicas. Efetivamente, isso isola o acesso à rede para um determinado recurso do CloudFormation apenas da VPC específica dentro da rede da AWS.

Resiliência no CloudFormation

A infraestrutura global da AWS é criada com base em AWS Regiões e Zonas de Disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, conectadas com baixa latência, throughput elevado 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 altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte [Infraestrutura global da AWS](#).

Validação de conformidade do CloudFormation

Para saber se um AWS service (Serviço da AWS) está no escopo de programas de conformidade específicos, consulte [Serviços da AWS no escopo por programa de conformidade](#) e selecione o programa de conformidade em que você está interessado. Para obter informações gerais, consulte [Programas de Conformidade da AWS](#).

É possível baixar relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte [Baixar relatórios no AWS Artifact](#).

Sua responsabilidade de conformidade ao usar o Serviços da AWS é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. Para ter mais informações sobre sua responsabilidade pela conformidade ao usar Serviços da AWS, consulte a [documentação da AWS sobre segurança](#).

Análise de vulnerabilidade e configuração no CloudFormation

A configuração e os controles de TI são uma responsabilidade compartilhada entre a AWS e você, nosso cliente. Para obter mais informações, consulte o AWS [modelo de responsabilidade compartilhada da](#) .

Práticas recomendadas de segurança do CloudFormation

O CloudFormation oferece uma série de recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As práticas recomendadas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Tópicos

- [Usar o IAM para controlar o acesso](#)
- [Não incorporar credenciais em seus modelos](#)
- [Use o AWS CloudTrail para registrar chamadas do CloudFormation](#)

Usar o IAM para controlar o acesso

O IAM é um serviço da AWS que você pode usar para gerenciar usuários e suas respectivas permissões na AWS. É possível usar o IAM com o CloudFormation para especificar quais ações do CloudFormation os usuários podem executar, como visualizar os modelos de pilha, criar pilhas ou excluir pilhas. Além disso, qualquer pessoa que gerencie as pilhas do CloudFormation precisará de permissões para os recursos dentro dessas pilhas. Por exemplo, se os usuários quiserem usar o CloudFormation para executar, atualizar ou encerrar instâncias do Amazon EC2, eles precisarão ter permissão para chamar as ações relevantes do Amazon EC2.

Na maioria dos casos, os usuários precisam de acesso total para gerenciar todos os recursos em um modelo. O CloudFormation faz chamadas para criar, modificar e excluir esses recursos em nome dele. Para permissões separadas entre um usuário e o serviço do CloudFormation, use um perfil de serviço. O CloudFormation usa a política do perfil de serviço para fazer chamadas em vez da política do usuário. Para obter mais informações, consulte [Função de serviço do CloudFormation](#).

Não incorporar credenciais em seus modelos

Em vez de incorporar informações confidenciais nos modelos do CloudFormation, é recomendável usar referências dinâmicas no modelo de pilha.

As referências dinâmicas fornecem uma maneira compacta e avançada para você fazer referência a valores externos que são armazenados e gerenciados em outros serviços, como o Repositório de parâmetros do AWS Systems Manager ou o AWS Secrets Manager. Quando você usa uma referência dinâmica, o CloudFormation recupera o valor da referência especificada, se necessário, durante operações de pilha e conjunto de alterações e passa o valor para o recurso apropriado. No entanto, o CloudFormation nunca armazena o valor real do parâmetro. Para obter mais informações, consulte [Obter valores armazenados em outros serviços usando referências dinâmicas](#).

O [AWS Secrets Manager](#) ajuda você a criptografar, armazenar e recuperar credenciais com segurança para bancos de dados e outros serviços. O [Repositório de parâmetros do AWS Systems Manager](#) fornece armazenamento hierárquico seguro para o gerenciamento de dados de configuração.

Para obter mais informações sobre a definição de parâmetros do modelo, consulte [Sintaxe de Parameters de modelo do CloudFormation](#).

Use o AWS CloudTrail para registrar chamadas do CloudFormation

O AWS CloudTrail rastreia qualquer pessoa que faça chamadas de API do CloudFormation na Conta da AWS. As chamadas de API são registradas em log sempre que alguém usa a API do CloudFormation, o console do CloudFormation, um console de back-end ou comandos da AWS CLI do CloudFormation. Ative o registro em log e especifique um bucket do Amazon S3 para armazenar os logs. Dessa forma, se você precisar, poderá auditar quem fez qual chamada do CloudFormation em sua conta. Para obter mais informações, consulte [Registrar em log chamadas de API do CloudFormation com o AWS CloudTrail](#).

Acessar o CloudFormation usando um endpoint de interface (AWS PrivateLink)

É possível usar o AWS PrivateLink para criar uma conectividade privada entre a sua VPC e o CloudFormation. Você pode acessar o CloudFormation como se estivesse em sua VPC, sem usar um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão Direct Connect. As instâncias na sua VPC não precisam de endereços IP públicos para acessar o CloudFormation.

Você estabelece essa conectividade privada criando um endpoint de interface, desenvolvido pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Estas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao CloudFormation.

CloudFormation suporta fazer chamadas para todas as suas ações de API por meio do endpoint da interface.

Considerações sobre endpoints da VPC do CloudFormation

Antes de configurar um endpoint de interface, verifique se você atendeu aos pré-requisitos no tópico [Access an AWS service using an interface VPC endpoint](#) no AWS PrivateLink Guide.

Os seguintes pré-requisitos e considerações adicionais se aplicam ao configurar um endpoint de interface para o CloudFormation:

- Se você tiver recursos na VPC que devam responder a uma solicitação de recurso personalizado ou a uma condição de espera, certifique-se de que eles tenham acesso aos buckets do Amazon S3 específicos do CloudFormation. O CloudFormation tem buckets do S3 em cada região para monitorar respostas a uma solicitação de [recurso personalizado](#) ou a uma [condição de espera](#). Caso um modelo inclua recursos personalizados ou condições de espera em uma VPC, a política de endpoint da VPC deve permitir que usuários enviem respostas para os seguintes buckets:
 - Para recursos personalizados, permita o tráfego para o bucket `cloudformation-custom-resource-response-region`. Ao usar recursos personalizados, os nomes de Região da AWS não contêm traços. Por exemplo, `uswest2`.
 - Para condições de espera, permita o tráfego para o bucket `cloudformation-waitcondition-region`. Ao usar condições de espera, os nomes de Região da AWS contêm traços. Por exemplo, `us-west-2`.

Se a política de endpoint bloquear tráfego para esses buckets, o CloudFormation não receberá respostas, e a operação de pilha falhará. Por exemplo, se você tiver um recurso em uma VPC na região `us-west-2` que deva responder a uma condição de espera, o recurso deverá poder enviar uma resposta para o bucket `cloudformation-waitcondition-us-west-2`.

Para obter uma lista de todas as Regiões da AWS em que o CloudFormation está disponível atualmente, consulte a página [CloudFormation endpoints and quotas](#) na Referência geral da Amazon Web Services.

- No momento, os VPC endpoints não são compatíveis com solicitações entre regiões: crie o endpoint na mesma região em que planeja emitir suas chamadas de API para o CloudFormation.
- Endpoints de VPC oferecem suporte somente a DNS fornecido pela Amazon por meio do Amazon Route 53. Se quiser usar seu próprio DNS, poderá usar o encaminhamento de DNS condicional. Para obter mais informações, consulte [Conjuntos de opções DHCP no Amazon VPC](#) no Manual do usuário da Amazon VPC.
- O grupo de segurança anexado ao endpoint da VPC deve permitir conexões de entrada na porta 443 na sub-rede privada da VPC.

Criar um endpoint da VPC de interface para o CloudFormation

É possível criar um endpoint da VPC para o CloudFormation usando a console da Amazon VPC ou a AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte [Create a VPC endpoint](#) (Criar um endpoint da VPC) no Guia do AWS PrivateLink.

Como criar um endpoint de interface para o CloudFormation usando o seguinte nome de serviço:

- `com.amazonaws.region.cloudformation`: cria um endpoint para as operações de API do CloudFormation.

Se você habilitar o DNS privado para o endpoint da interface, poderá fazer solicitações de API a CloudFormation usando seu nome DNS regional padrão. Por exemplo, `cloudformation.us-east-1.amazonaws.com`.

Nas Regiões da AWS em que os endpoints específicos do FIPS são compatíveis, você também pode criar um endpoint de interface para o CloudFormation usando o seguinte nome de serviço:

- `com.amazonaws.region.cloudformation-fips`: cria um endpoint para a API do CloudFormation que está em conformidade com [Federal Information Processing Standard \(FIPS\) 140-2](#).

Para obter uma lista completa de endpoints do CloudFormation, consulte [endpoints e cotas do CloudFormation](#) no Referência geral da Amazon Web Services. .

Criando uma política de endpoint da VPC para o CloudFormation

Uma política de endpoint é um recurso do IAM que pode ser anexado ao endpoint de interface. A política de endpoint padrão permite acesso total ao CloudFormation por meio de endpoint de

interface. Para controlar o acesso permitido ao CloudFormation pela VPC, anexe uma política de endpoint personalizada ao endpoint de interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem realizar ações (Contas da AWS, usuários do IAM e perfis do IAM).
- As ações que podem ser executadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte [Controlar o acesso aos endpoints da VPC usando políticas de endpoint](#) no Guia AWS PrivateLink.

Exemplo: política de endpoint da VPC para ações do CloudFormation

Veja a seguir um exemplo de uma política de endpoint para o CloudFormation. Quando anexada a um endpoint, essa política concede acesso às ações indicadas do CloudFormation para todos os principais em todos os recursos. O exemplo a seguir nega a todos os usuários a permissão para criar pilhas por meio do endpoint da VPC e permite acesso total a todas as outras ações no serviço do CloudFormation.

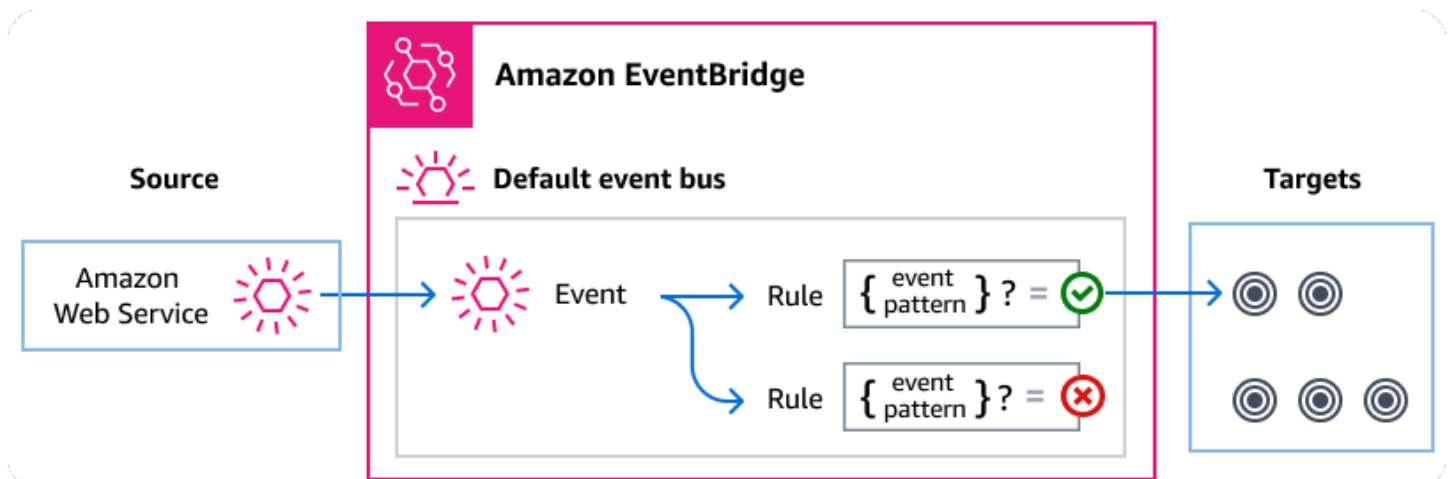
```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Monitorar eventos de sincronização do CloudFormation e do Git com o EventBridge

O Amazon EventBridge é um serviço sem servidor que usa eventos para conectar os componentes da aplicação, facilitando a criação de aplicações escaláveis orientadas por eventos. A arquitetura orientada por eventos é um estilo de criação de sistemas de software com acoplamento fraco que funcionam juntos emitindo e respondendo a eventos. Os eventos representam uma mudança em um recurso ou ambiente.

Como ocorre em muitos serviços da AWS, o CloudFormation gera e envia eventos para o barramento de eventos padrão do EventBridge, que é provisionado automaticamente em cada Conta da AWS. Um barramento de eventos é um roteador que recebe eventos e os entrega a zero ou mais destinos, ou alvos. As regras especificadas para o barramento de eventos avaliam os eventos à medida que eles chegam. Cada regra verifica se um evento corresponde ao padrão do evento. Se o evento corresponder, o barramento de eventos enviará o evento para os destinos especificados.

Para obter mais informações, consulte [Começar a usar o Amazon EventBridge](#) no Manual do usuário do Amazon EventBridge.



Tópicos

- [Visão geral dos eventos de sincronização do CloudFormation e do Git](#)
- [Permissões do Amazon EventBridge](#)
- [Criar um padrão de evento personalizado para uma regra do EventBridge](#)
- [Referência detalhada de eventos do CloudFormation](#)

Visão geral dos eventos de sincronização do CloudFormation e do Git

O CloudFormation envia eventos ao EventBridge sempre que uma operação de criação, atualização, exclusão ou detecção de desvio é executada em uma pilha. O CloudFormation também envia eventos ao EventBridge por alterações de status em conjuntos de pilhas e instâncias de conjuntos de pilhas. Você pode usar regras do EventBridge para rotear eventos para outros destinos. A entrega desses eventos é garantida, mas eles podem ser entregues fora da ordem.

Como os eventos do CloudFormation representam alterações em pilhas ou conjuntos de pilhas e seus recursos, você pode usá-los para iniciar fluxos de trabalho associados aos respectivos eventos. Por exemplo:

- Crie tags específicas de pilhas ou de conjuntos de pilhas em todos os recursos provisionados por meio do CloudFormation.
- Estabeleça uma associação entre uma pilha ou um conjunto de pilhas do CloudFormation e um Amazon WorkSpaces Application Manager (Amazon WAM).
- Especifique uma associação com um AppRegistry para a pilha ou o conjunto de pilhas criado.

Os eventos a seguir são gerados pelo CloudFormation e enviados para o barramento de eventos padrão no EventBridge. Para obter mais informações, consulte [???](#).

Tipo de evento	Descrição	Origem do evento
Alteração de status de recursos	Qualquer atualização realizada em uma pilha que altera as propriedades do recurso subjacente. Para obter uma lista completa dos tipos de recursos da AWS compatíveis, consulte a Referência de tipos de propriedades e recursos da AWS.	AWS CloudFormation
Alteração do status da pilha	Representa uma mudança de status em uma determinada pilha.	AWS CloudFormation

Tipo de evento	Descrição	Origem do evento
	Para obter detalhes do código, consulte Códigos de status da pilha .	
Alteração no status da detecção de desvios	Representa uma atualização de detecção de desvio iniciada pelo usuário em uma determinada pilha. Para obter uma lista de tipos totalmente mutáveis e imutáveis que são compatíveis com a detecção de desvios, consulte Suporte a tipos de recursos	AWS CloudFormation
Alteração do status de StackSet	Representa uma mudança de status em um determinado conjunto de pilhas.	AWS CloudFormation
Alteração no status da instância de pilha do StackSet	Representa uma mudança de status em uma instância específica de pilha do StackSet. Para obter detalhes do código, consulte Códigos de status da instância de pilha .	AWS CloudFormation
Status da operação do StackSet	Representa uma mudança de status em uma determinada operação do StackSet. Para obter detalhes do código, consulte Códigos de status do StackSets .	AWS CloudFormation

Além disso, a sincronização do Git do AWS CloudFormation envia eventos de mudanças de status para sincronizações de repositórios e de recursos ao EventBridge.

Os eventos de sincronização do Git a seguir são gerados pelo CodeConnections e enviados ao barramento de eventos padrão no EventBridge. Para obter mais informações, consulte [???](#).

Tipo de evento	Descrição	Origem do evento
Alteração do status de sincronização do repositório	Representa uma alteração do status em uma sincronização do repositório do Git.	Conexões de código da AWS
Alteração do status de sincronização de recursos	Representa uma alteração do status em uma sincronização de recursos do Git.	Conexões de código da AWS

Permissões do Amazon EventBridge

O CloudFormation não requer outras permissões para entregar eventos ao EventBridge. Os eventos contêm informações que já estão disponíveis por meio de operações de API do CloudFormation.

Talvez os destinos que você especificar precisem de determinadas permissões ou configurações. Para obter mais detalhes sobre o uso de serviços específicos para destinos, consulte [Amazon EventBridge targets](#) no Guia do usuário do Amazon EventBridge.

Criar um padrão de evento personalizado para uma regra do EventBridge

É possível encontrar vários padrões predefinidos no EventBridge para eventos de sincronização do CloudFormation e do Git. Isso simplifica a forma como um padrão de evento é criado. Em vez de escrever seus próprios padrões de eventos, é possível selecionar valores de campo em um formulário e o EventBridge gera o padrão para você. É possível criar uma regra usando um desses padrões de eventos predefinidos ou criar seu próprio padrão de evento personalizado.

Quando um serviço como o CloudFormation entrega um evento ao barramento de eventos padrão, o EventBridge usa o padrão de evento definido na regra para determinar se o evento deve ser entregue aos destinos da regra. Um padrão de evento corresponde aos dados nos eventos do CloudFormation desejados.

Cada padrão de evento é um objeto JSON que contém:

- Um atributo `source` que identifica o serviço no qual o evento é originado. Por exemplo, o `aws.cloudformation` ou o `aws.codeconnections`.

- (Opcional): um atributo `detail-type` que contém uma matriz dos tipos de eventos a serem correlacionados.
- (Opcional): um atributo `detail` que contém quaisquer outros dados relacionados aos eventos a serem correlacionados.

Por exemplo, o ID da pilha, os recursos envolvidos, o status de vários recursos e outros dados relevantes para um tipo específico de eventos.

Por exemplo, o seguinte padrão de evento corresponde a todos os eventos de alteração do status do recurso:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

O seguinte padrão de eventos usa dados detalhados dos eventos a serem correlacionados somente com os eventos de alteração de status do recurso em que o CloudFormation cria um recurso `AWS::S3::Bucket` ou `AWS::SNS::Topic`:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

Consulte mais informações sobre como escrever padrões de eventos em [Event patterns](#) no Guia do usuário do Amazon EventBridge.

Referência detalhada de eventos do CloudFormation

Todos os eventos dos serviços da AWS têm um conjunto comum de campos contendo metadados sobre o evento, como o serviço da AWS que é a origem do evento, a hora em que o evento foi gerado, a conta e a região em que o evento ocorreu, e outros. Para obter as definições desses

campos gerais, consulte [Metadados de eventos de serviço da AWS](#) na Referência de eventos da AWS.

Além disso, cada evento tem um campo de `detail` que contém dados específicos desse determinado evento. A referência abaixo define os campos de detalhes dos vários eventos do CloudFormation.

Ao usar o EventBridge para selecionar e gerenciar eventos do CloudFormation, é importante ter em mente o seguinte:

- O campo do `source` especifica a origem de evento.

Por exemplo, `aws.cloudformation` ou `aws.codeconnections`.

- O campo do `detail-type` especifica o tipo de evento.

Por exemplo, `CloudFormation Resource Status Change` ou `CloudFormation Drift Detection Status Change`.

- O campo de `detail` contém os dados específicos desse determinado evento.

Por exemplo, o ID da pilha, os recursos envolvidos, o status de vários recursos e outros dados relevantes para um tipo específico de eventos.

Consulte mais informações sobre a criação de padrões de eventos que habilitam regras para corresponder a eventos do CloudFormation em [Event patterns](#) no Guia do usuário do Amazon EventBridge.

Consulte mais informações sobre eventos e como o EventBridge os processa em [EventBridge events](#) no Guia do usuário do Amazon EventBridge.

Tópicos

- [Detalhes do evento de alteração de status de recurso](#)
- [Detalhes do evento de alteração de status de desvios](#)
- [Detalhe do evento de alteração de status de detecção de desvios](#)
- [Detalhes do evento de alteração de status do StackSet](#)
- [Detalhes do evento de alteração de status de instância de pilha do StackSet](#)
- [Detalhes do evento de alteração de status de operação do StackSet](#)
- [Detalhe do evento de alteração do status de sincronização do repositório](#)

- [Detalhe do evento de alteração do status de sincronização de recursos](#)

Detalhes do evento de alteração de status de recurso

Abaixo estão os campos de detalhes para eventos de Alteração de status de recurso.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```
{
  . . . ,
  "detail-type": "CloudFormation Resource Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-id" : "string",
    "logical-resource-id" : "string",
    "physical-resource-id": "string",
    "status-details": {
      "status": "string",
      "status-reason": "string"
    },
    "resource-type": "string",
    "client-request-token": "string"
  }
}
```

detail-type

Identifica o tipo de evento.

Para eventos de status de recursos, esse valor é `CloudFormation Resource Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status de recursos, esses dados incluem:

`stack-id`

O ID de pilha exclusivo que é associado à pilha.

`logical-resource-id`

O nome lógico do recurso, conforme definido no modelo.

`physical-resource-id`

O nome ou o identificador exclusivo que corresponde ao ID de uma instância física de um recurso compatível com o CloudFormation.

`status-details`

`status`

Status do recurso.

`status-reason`

Motivo do status do recurso.

`resource-type`

Tipo de recurso. Por exemplo, `AWS::S3::Bucket`.

`client-request-token`

Um token de acesso usado para chamar a API. Todos os eventos que são iniciados por uma determinada operação de pilha são atribuídos ao mesmo token de solicitação do cliente, que você pode usar para rastrear as operações. Operações de pilha iniciadas a partir do console usam o formato de token `Console-StackOperation-ID`, que ajuda você a identificar facilmente a operação da pilha. Por exemplo, se você criar uma pilha usando o console, cada evento de pilha resultante receberia o mesmo token no seguinte formato: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Exemplo: evento de alteração de status de recurso

O exemplo abaixo é de um evento de status de recurso. Esse evento detalha que o CloudFormation criou com êxito o recurso solicitado, um bucket do Amazon S3, na pilha especificada.

```
{
```

```

    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "CloudFormation Resource Status Change",
    "source": "aws.cloudformation",
    "account": "111122223333",
    "time": "2017-12-22T18:43:48Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
    ],
    "detail": {
        "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
        "logical-resource-id": "my-s3-bucket",
        "physical-resource-id": "arn:aws:s3:::my-s3-bucket-us-east-1",
        "status-details": {
            "status": "CREATE_COMPLETE",
            "status-reason": ""
        },
        "resource-type": "AWS::S3::Bucket",
        "client-request-token": ""
    }
}

```

Detalhes do evento de alteração de status de desvios

Abaixo estão os campos de detalhes para os eventos de alteração de status de desvios.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```

{
    . . . ,
    "detail-type": "CloudFormation Stack Status Change",
    "source": "aws.cloudformation",
    . . . ,
    "detail": {
        "stack-id": "string",
        "status-details": {
            "status": "string",
            "status-reason": "string"
        },
        "client-request-token": "string"
    }
}

```

```
}
```

detail-type

Identifica o tipo de evento.

Para eventos de status da pilha, esse valor é `CloudFormation Stack Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status da pilha, esses dados incluem:

stack-id

O ID de pilha exclusivo associado à pilha.

status-details

status

Status da pilha.

Para obter uma lista completa de códigos de status de pilhas, consulte [Códigos de status da pilha](#).

status-reason

Motivo do status do recurso.

client-request-token

Um token de acesso usado para chamar a API. Todos os eventos que são iniciados por uma determinada operação de pilha são atribuídos ao mesmo token de solicitação do cliente, que você pode usar para rastrear as operações. Operações de pilha iniciadas a partir do console usam o formato de token `Console-StackOperation-ID`, que ajuda você a identificar facilmente a operação da pilha. Por exemplo, se você criar uma pilha usando o console, cada evento de pilha resultante receberia o mesmo token no seguinte formato: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Exemplo: evento de status de pilha

O exemplo abaixo é de um evento de status da pilha, em que o CloudFormation criou com êxito a pilha solicitada, `teststack`.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Stack Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
  ],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "status-details": {
      "status": "CREATE_COMPLETE",
      "status-reason": ""
    },
    "client-request-token": ""
  }
}
```

Detalhe do evento de alteração de status de detecção de desvios

Abaixo estão os campos de detalhes para os eventos de detecção de desvios das pilhas.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```
{
  . . . ,
  "detail-type": "CloudFormation Drift Detection Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-id": "string",
    "stack-drift-detection-id": "string",
    "status-details": {
```

```
    "stack-drift-status": "string",
    "detection-status": "string"
  },
  "drift-detection-details": {
    "drifted-stack-resource-count": integer
  },
  "client-request-token": "string"
}
```

detail-type

Identifica o tipo de evento.

Para eventos de detecção de desvio de pilha, esse valor é `CloudFormation Drift Detection Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de detecção de desvio de pilha, esses dados incluem:

stack-id

O ID de pilha exclusivo que é associado à pilha.

stack-drift-detection-id

O ID de detecção de desvio de pilha.

status-details

stack-drift-status

Status de desvio da pilha.

detection-status

Status da operação de detecção de desvio.

drift-detection-details

drifted-stack-resource-count

Número de recursos com desvios. Quando o valor é -1, a detecção de desvio está em andamento. Todos os outros números inteiros não negativos representam o número real de recursos com desvios.

client-request-token

Um token de acesso usado para chamar a API. Todos os eventos que são iniciados por uma determinada operação de pilha são atribuídos ao mesmo token de solicitação do cliente, que você pode usar para rastrear as operações. Operações de pilha iniciadas a partir do console usam o formato de token Console-StackOperation-ID, que ajuda você a identificar facilmente a operação da pilha. Por exemplo, se você criar uma pilha usando o console, cada evento de pilha resultante receberia o mesmo token no seguinte formato: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Example Exemplo: evento de detecção de desvio de pilha

O exemplo abaixo é de um evento de detecção de desvio de pilha. Esse evento detalha que o CloudFormation concluiu a detecção de desvio na pilha especificada e que a pilha atualmente tem um status de desvio DRIFTED devido a um recurso desviado.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Drift Detection Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": ["string"],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "stack-drift-detection-id": "624af370-311a-11e8-b6b7-500cexample",
    "status-details": {
      "stack-drift-status": "DRIFTED",
      "detection-status": "DETECTION_COMPLETE"
    },
    "drift-detection-details": {
      "drifted-stack-resource-count": 1
    }
  }
}
```

```
    },  
    "client-request-token":""  
  }  
}
```

Detalhes do evento de alteração de status do StackSet

Abaixo estão os campos de detalhes para os eventos de alteração de status do StackSet.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```
{  
  . . . ,  
  "detail-type": "CloudFormation StackSet Status Change",  
  "source": "aws.cloudformation",  
  . . . ,  
  "detail": {  
    "stack-set-arn" : "string",  
    "status-details": {  
      "status": "string"  
    }  
  }  
}
```

detail-type

Identifica o tipo de evento.

Para eventos de evento de status do StackSet, esse valor é `CloudFormation StackSet Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de eventos de status do StackSet, esses dados incluem:

stack-set-arn

O nome do recurso da Amazon (ARN) associado ao StackSet.

status-details

status

O status do StackSet.

Valores válidos: ACTIVE | DELETED

Example Exemplo: evento de alteração de status do StackSet

O exemplo abaixo é de um evento de alteração de status do StackSet. Esse evento detalha que o CloudFormation excluiu o conjunto de pilhas especificado.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-23T17:06:18Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663"
  ],
  "detail": {
    "stack-set-arn" : "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663",
    "status-details": {
      "status": "DELETED"
    }
  }
}
```

Detalhes do evento de alteração de status de instância de pilha do StackSet

Abaixo estão os campos de detalhes para os eventos de status da instância da pilha do StackSet.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```
{
  . . . ,
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-id" : "string",
    "action" : "string",
    "status-details": {
      "status": "string",
      "status-reason": "string",
      "detailed-status": "string"
    }
  }
}
```

detail-type

Identifica o tipo de evento.

Para eventos de status da instância da pilha do StackSet, esse valor é `CloudFormation StackSet StackInstance Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status da instância da pilha do StackSet, esses dados incluem:

stack-set-arn

O nome do recurso da Amazon (ARN) associado ao StackSet.

stack-id

O ID de pilha exclusivo que é associado à instância de pilha.

action

O tipo de operação de conjunto de pilhas.

Valores válidos: CREATE | UPDATE | DELETE | DETECT_DRIFT

status-details

status

O status da instância do StackSet.

Para obter mais detalhes, consulte [Códigos de status da instância de pilha](#).

Valores válidos: CURRENT | OUTDATED | INOPERABLE

status-reason

Motivo do status da instância do StackSet.

detailed-status

O status detalhado da instância do StackSet.

Valores válidos: CANCELLED | FAILED | FAILED_IMPORT | INOPERABLE | PENDING | RUNNING | SKIPPED_SUSPENDED_ACCOUNT | SUCCEEDED

Example Exemplo: evento de alteração de status de instância de pilha do StackSet

O exemplo abaixo é de um evento de alteração de status de instância de pilha do StackSet.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xnh0934z09",
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  "account": "11112223333",
  "time": "2021-09-22T19:19:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:11112223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59"
  ]
}
```

```

    ],
    "detail": {
      "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test1234:e5f54eea-d041-44ad-94f8-b8268acale59",
      "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
      "status-details": {
        "status": "OUTDATED",
        "status-reason": "User Initiated",
        "detailed-status": "PENDING"
      }
    }
  }
}

```

Detalhes do evento de alteração de status de operação do StackSet

Abaixo estão os campos de detalhes para os eventos de alteração de status de operação do StackSet.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```

{
  . . . ,
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-set-operation-id" : "string",
    "status-details": {
      "status": "string"
    }
  }
}
}

```

detail-type

Identifica o tipo de evento.

Para eventos de status de operação do StackSet, esse valor é `CloudFormation StackSet Operation Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos do CloudFormation, esse valor é `aws.cloudformation`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status de operação do StackSet, esses dados incluem:

stack-set-arn

O nome do recurso da Amazon (ARN) associado ao StackSet.

stack-set-operation-id

O ID exclusivo que é associado à operação do StackSet.

status-details

status

O status da operação do StackSet.

Para obter mais detalhes, consulte [Códigos de status do StackSets](#).

Valores válidos: RUNNING | SUCCEEDED | FAILED | STOPPING | STOPPED | QUEUED

Example Exemplo: evento de alteração status de operação do StackSet

O exemplo abaixo é de um evento de alteração de status de operação do StackSet. O evento detalha que o CloudFormation concluiu com êxito a operação solicitada no conjunto de pilhas especificado.

```
{
  "version": "0",
  "id": "4de89905-fd92-6a6b-9509-23c04bcb6a21",
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T05:46:24Z",
  "region": "us-east-1",
```

```

"resources": [
  "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-
d041-44ad-94f8-b8268aca1e59"
],
"detail": {
  "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
  "stack-set-operation-id": "ce69adce-2221-4483-8c4b-c51f284f25e8",
  "status-details": {
    "status": "SUCCEEDED"
  }
}
}

```

Detalhe do evento de alteração do status de sincronização do repositório

Veja a seguir os campos de detalhes para eventos de alteração de status de sincronização do repositório.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```

{
  . . . ,
  "detail-type": "Git Sync Repository Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
    "connectionArn" : "string",
    "providerType" : "string",
    "repositoryName": "string",
    "providerType": "string",
    "repositoryName": "string",
    "repositoryArn": "string",
    "repositoryLinkId": "string",
    "ownerId": "string",
    "commit": "string",
    "branch": "string",
    "syncType": "string",
    "status": "string",
    "previousSync": "string"
  }
}

```

```
}
```

detail-type

Identifica o tipo de evento.

Para eventos de status de sincronização do repositório, esse valor é `Git Sync Repository Sync Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos de sincronização do Git, esse valor é `aws.codeconnections`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status de sincronização do repositório, esses dados incluem:

connectionArn

O nome do recurso da Amazon (ARN) associado ao CodeConnections.

providerType

O provedor Git conectado ao CloudFormation.

Valores válidos: `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

repositoryName

O nome do repositório Git.

repositoryArn

O ARN associado ao repositório Git.

repositoryLinkId

O ID exclusivo associado ao link do repositório.

ownerId

O ID exclusivo associado ao proprietário do repositório.

commit

O ID da conta exclusivo associado à confirmação do repositório.

branch

O ID exclusivo associado à ramificação do repositório.

syncType

O tipo de sincronização que está sendo executada.

status

O status de sincronização do repositório atual.

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

O status de sincronização anterior ao status atual.

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Exemplo: evento de alteração do status de sincronização do repositório

O exemplo a seguir é de um evento de alteração de status de sincronização do repositório. O evento detalha que o CodeConnections sincronizou com êxito o repositório.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "GitSync Repository Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000"],
  "detail": {
    "connectionArn": "arn:aws:codestar-connections:us-east-1:111122223333:connection/sample-connection-id",
    "providerType": "GitHub",
    "repositoryName": "sample-repository-name",
    "repositoryArn": "arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000",
    "repositoryLinkId": "550e8400-e29b-41d4-a716-446655440000",
    "ownerId": "sample-owner-id",
    "commit": "sample-commit-id",
```



```
"branch": "main",
"syncType": "CFN_STACK_SYNC",
"status": "SUCCEEDED",
"previousStatus": "IN_PROGRESS",
}
}
```

Detalhe do evento de alteração do status de sincronização de recursos

Veja a seguir os campos de detalhes para eventos de alteração do status de sincronização de recursos.

Os campos `source` e `detail-type` estão incluídos porque contêm valores específicos para eventos do .

```
{
  . . . ,
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
    "providerType" : "string",
    "commit" : "string",
    "repositoryName": "string",
    "branch": "string",
    "syncType": "string",
    "syncTarget": "string",
    "status": "string",
    "previousSync": "string"
  }
}
```

detail-type

Identifica o tipo de evento.

Para eventos de status de sincronização do repositório, esse valor é `Git Sync Repository Sync Status Change`.

source

Identifica o serviço que gerou o evento. Para eventos de sincronização do Git, esse valor é `aws.codeconnections`.

detail

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Para eventos de status de sincronização de recursos, esses dados incluem:

providerType

O provedor Git conectado ao CloudFormation.

Valores válidos: GitHub | GitHub Enterprise | GitLab | BitBucket

commit

O ID da conta exclusivo associado à confirmação do repositório.

repositoryName

O nome do repositório Git.

branch

O ID exclusivo associado à ramificação do repositório.

syncType

O tipo de sincronização que está sendo executada.

syncTarget

A pilha de destino da sincronização de recursos.

status

O status de sincronização do repositório atual.

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

O status de sincronização anterior ao status atual.

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Exemplo: evento de alteração do status de sincronização de recursos

O exemplo a seguir é de um evento de alteração do status de recursos. O evento detalha que o CodeConnections sincronizou com êxito o recurso.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:cloudformation:us-east-1:111122223333:stack/
targetStack1"],
  "detail": {
    "providerType": "GitHub",
    "commit": "sample-commit-id",
    "repositoryName": "sample-repository-name",
    "branch": "main",
    "syncType": "CFN_STACK_SYNC",
    "syncTarget": "arn:aws:aws:cloudformation:us-east-1:111122223333:stack/
targetStack1",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS"
  }
}
```

Entender as cotas do CloudFormation

Sua Conta da AWS tem cotas do CloudFormation que você talvez precise conhecer ao criar modelos e pilhas. Ao compreender essas cotas, você pode evitar erros de limitação que exigiriam reprojeter os modelos ou as pilhas.

A tabela a seguir mostra as cotas do CloudFormation.

Cotas	Descrição	Valor	Estratégia de ajuste
Dados da condição de espera de cfn-signal	Quantidade máxima de dados que <code>cfn-signal</code> pode passar.	4,096 bytes	Para passar uma quantidade maior, envie os dados para um bucket do Amazon S3 e, em seguida, use <code>cfn-signal</code> para passar o URL do Amazon S3 para esse bucket.
Resposta de recursos personalizados	A quantidade máxima de dados que um provedor de recursos personalizados pode passar.	4,096 bytes	
Referências dinâmicas por modelo	O número máximo de referências dinâmicas permitidas em um único modelo de pilha do CloudFormation.	60 referências dinâmicas em um modelo de pilha	
Hooks por conta	O número máximo de hooks permitidos por conta, por região.	100 hooks	
Hooks por recurso	O número máximo de hooks que podem	100 hooks	

Cotas	Descrição	Valor	Estratégia de ajuste
	ser configurados por recurso.		
Tamanho de configuração do hook	A quantidade máxima de dados de configuração que um hook pode armazenar.	204,8 KB	
Mapeamentos	O número máximo de mapeamentos que é possível declarar no modelo do CloudFormation.	200 mapeamentos	Para especificar mais mapeamentos, separe o modelo em vários modelos, usando, por exemplo, pilhas aninhadas .
Atributos de mapeamento	O número máximo de atributos de mapeamento para cada mapeamento o que é possível declarar no modelo do CloudFormation.	200 atributos	Para especificar mais atributos de mapeamento, separe os atributos em vários mapeamentos.
Nome do mapeamento e nome do atributo de mapeamento	O tamanho máximo de cada nome de mapeamento.	255 caracteres	
Modules	O número máximo de módulos que você pode registrar no registro do CloudFormation por conta e região.	100 módulos	

Cotas	Descrição	Valor	Estratégia de ajuste
Versões de módulo	O número máximo de versões que você pode registrar no registro do CloudFormation para um determinado módulo.	100 versões	Para registrar novas versões, primeiramente use DeRegisterType para cancelar o registro de versões que você não esteja mais usando.
Pilhas aninhadas	O número máximo de recursos do CloudFormation que uma pilha aninhada pode criar, atualizar ou excluir por operação. Por exemplo, você pode ter uma hierarquia de pilha aninhada com mais de 2.500 recursos no total, mas não conseguir criar, atualizar ou excluir mais de 2.500 desses recursos em uma única implantação.	2500 recursos	Divida a hierarquia da pilha em pilhas diferentes.
Saídas	O número máximo de saídas que é possível declarar no modelo do CloudFormation.	200 saídas	
Nome da saída	O tamanho máximo de um nome de saída.	255 caracteres	

Cotas	Descrição	Valor	Estratégia de ajuste
Parâmetros	O número máximo de parâmetros que é possível declarar no modelo do CloudFormation.	200 parâmetros	Para especificar mais parâmetros, você pode usar mapeamentos ou listas para atribuir vários valores a um único parâmetro.
Nome do parâmetro	O tamanho máximo de um nome de parâmetro.	255 caracteres	
Valor do parâmetro	O tamanho máximo de um valor de parâmetro.	4,096 bytes	Para usar um valor parametrizado maior, crie vários parâmetros e depois use a função Fn::Join para concatenar os vários valores em um único valor.
Recursos privados	O número máximo de recursos privados que você pode registrar no registro do CloudFormation por conta e região.	50 recursos privados	
Versões de recursos privados	O número máximo de versões que você pode registrar no registro do CloudFormation para um determinado recurso privado.	50 recursos privados	Para registrar novas versões, primeiramente use DeRegisterType para cancelar o registro de versões que você não esteja mais usando.

Cotas	Descrição	Valor	Estratégia de ajuste
Recursos	O número máximo de recursos que é possível declarar no modelo do CloudFormation.	500 recursos	Para especificar mais recursos, separe o modelo em vários modelos, usando, por exemplo, pilhas aninhadas .
Recursos em operações de pilha simultâneas	O número máximo de recursos que você pode ter envolvidos em operações de pilha (criar, atualizar ou excluir operações) na sua região em um determinado momento.	Use a API DescribeAccountLimits para determinar o limite atual para uma conta em uma região específica.	
Nome do recurso	O tamanho máximo de um nome de recurso.	255 caracteres	
Pilhas	O número máximo de pilhas do CloudFormation que você pode criar.	2000 pilhas	Para criar mais pilhas, exclua pilhas de que você não precisa ou solicite um aumento no número máximo de pilhas em sua Conta da AWS. Para obter mais informações, consulte Service Quotas do AWS em Referência geral da AWS.

Cotas	Descrição	Valor	Estratégia de ajuste
Nome da stack	O tamanho máximo do nome de uma pilha.	128 caracteres	
StackSets	O número máximo de conjuntos de pilhas do CloudFormation que é possível criar na sua conta de administrador.	1.000 conjuntos de pilhas	Para criar mais conjuntos de pilhas, exclua conjuntos de pilhas de que você não precisa ou solicite um aumento no número máximo de conjuntos de pilhas em sua Conta da AWS. Para obter mais informações, consulte Service Quotas do AWS em Referência geral da AWS.
Instâncias da pilha	O número máximo de instâncias de pilha que é possível criar por conjunto de pilhas.	100.000 instâncias de pilha por conjunto de pilhas	Para criar mais instâncias de pilha, exclua as instâncias de pilha de que você não precisa ou solicite um aumento no número máximo de instâncias de pilha em sua Conta da AWS. Para obter mais informações, consulte Service Quotas do AWS em Referência geral da AWS.

Cotas	Descrição	Valor	Estratégia de ajuste
Operações de instâncias de StackSets	O número máximo de instâncias de pilha, em todos os conjuntos de pilhas, nas quais é possível executar operações em cada região ao mesmo tempo, por conta de administrador.	10.000 operações	Esse limite se aplica a todos os conjuntos de pilhas envolvidos em uma região. Ele inclui instâncias de pilha afetadas pelas operações de criação e atualização do conjunto de pilhas, bem como a criação, a atualização ou a exclusão de instâncias de pilha diretamente.
Operações em fila de StackSets	O número máximo de operações em fila para um conjunto de pilhas em um determinado momento.	10.000 operações	
Pilhas importadas usando um objeto do S3 por operação de conjunto de pilhas	O número máximo de pilhas que você pode importar usando o objeto do S3 por operação de conjunto de pilhas.	200 pilhas	
Pilhas importadas usando IDs de pilha em linha por operação de conjunto de pilhas	O número máximo de pilhas que você pode importar usando IDs de pilha em linha por operação de conjunto de pilhas.	10 pilhas	

Cotas	Descrição	Valor	Estratégia de ajuste
Tamanho do corpo de um modelo em uma solicitação	O tamanho máximo do corpo de um modelo que você pode passar em uma solicitação <code>CreateStack</code> , <code>UpdateStack</code> ou <code>ValidateTemplate</code> .	51.200 bytes	Para usar um corpo de modelo maior, separe o modelo em vários modelos, usando, por exemplo, pilhas aninhadas . Ou faça upload do modelo para um bucket do Amazon S3.
Tamanho do corpo de modelo em um objeto do Amazon S3	O tamanho máximo do corpo de um modelo que você pode passar em um objeto do Amazon S3 para uma solicitação <code>CreateStack</code> , <code>UpdateStack</code> , <code>ValidateTemplate</code> com um URL de modelo do Amazon S3.	1 MB	Para usar um corpo de modelo maior, separe o modelo em vários modelos, usando, por exemplo, pilhas aninhadas . Ou use minificação para reduzir o tamanho do modelo do CloudFormation.
Descrição do modelo	O tamanho máximo da descrição de um modelo.	1.024 bytes	
Versões por hook	O número máximo de versões que você pode criar por hook.	100 versões	

Disponibilidade de recursos

Nem todos os recursos do CloudFormation podem estar disponíveis em todas as regiões. Para obter mais informações sobre Regiões da AWS, consulte a [Tabela de regiões da infraestrutura global](#).

- No momento, [Macros](#) não estão disponíveis na seguinte região:
 - Ásia-Pacífico (Jacarta)
- [Executar implantações azul/verde do ECS por meio do CodeDeploy usando o CloudFormation](#) não está disponível atualmente nas seguintes regiões:
 - África (Cidade do Cabo)
 - Ásia-Pacífico (Osaka)
 - Europa (Milão)

StackSets e macros

O StackSets atualmente não oferece suporte à criação ou atualização de conjuntos de pilhas com permissões gerenciadas por serviço baseadas em modelos que contêm macros. Isso inclui transformações, que são macros hospedadas pelo CloudFormation. Para mais informações sobre macros, consulte [Macros de modelo](#).

Solução de problemas do CloudFormation

Ao usar o CloudFormation, você pode encontrar problemas ao criar, atualizar ou excluir pilhas do CloudFormation. As seções a seguir podem ajudar a solucionar alguns problemas comuns que você pode encontrar.

Para perguntas gerais sobre o CloudFormation, consulte as [Perguntas frequentes sobre o CloudFormation](#). Também é possível pesquisar respostas e postar perguntas na [Comunidade do CloudFormation](#) no AWS re:Post.

Tópicos

- [Guia de solução de problemas](#)
- [Solucionar de problemas de erros](#)
- [Entrar em contato com o suporte](#)

Guia de solução de problemas

Se houver falha no CloudFormation ao criar, atualizar ou excluir a pilha, você pode visualizar mensagens de erro ou logs para ajudar a saber mais sobre o problema. As tarefas a seguir descrevem métodos gerais para a solução de um problema do CloudFormation. Para obter mais informações sobre erros específicos e soluções, consulte a seção [Solucionar de problemas de erros](#).

- Use IDs de operação e filtragem de falhas para solucionar problemas específicos. O CloudFormation atribui IDs de operação exclusivos aos eventos de implantação, facilitando o rastreamento e a correlação de eventos relacionados. Você pode filtrar eventos de pilha por ID de operação para se concentrar em fases específicas de implantação e usar filtros de falha para identificar rapidamente a causa raiz das falhas de implantação. Consulte [Visualizar eventos das pilhas por operação](#)
- Use o console do CloudFormation para visualizar o status da pilha. No console, você pode visualizar uma lista de eventos de pilha enquanto sua pilha está sendo criada, atualizada ou excluída. Os eventos de pilha incluem IDs de operação que ajudam você a correlacionar eventos relacionados e rastrear operações individuais. Você pode clicar no ID da operação para obter uma solução de problemas específica. Na lista de eventos de operação, encontre o evento de falha e, em seguida, visualize o motivo do status desse evento. O motivo do status pode conter uma mensagem de erro do CloudFormation ou de um serviço específico que pode ajudá-lo a resolver

o problema. Para obter mais informações sobre a visualização de eventos de pilha, consulte [Visualizar eventos das pilhas por operação](#).

- Para problemas do Amazon EC2, visualize os logs do `cloud-init` e do `cfn`. Esses logs são publicados na instância do Amazon EC2, no diretório `/var/log/`. Esses logs capturam processos e saídas de comandos enquanto o CloudFormation está configurando sua instância. No Windows, visualize o serviço EC2Configure em `%ProgramFiles%\Amazon\EC2ConfigService`, EC2 Launch em `%ProgramData%\Amazon\EC2-Windows\Launch\Logs`, EC2 Launch v2 em `%ProgramData%\Amazon\EC2Launch\log` e os logs do `cfn` em `C:\cfn\log`.

Você também pode configurar o modelo do CloudFormation para que os logs sejam publicados no Amazon CloudWatch que exibe os logs no Console de gerenciamento da AWS para que você não precise se conectar à sua instância do Amazon EC2. Para obter mais informações, consulte [Visualizar logs do CloudFormation no console](#) no blog de gerenciamento de aplicações.

Solucionar de problemas de erros

Ao encontrar os seguintes erros na pilha do CloudFormation, você pode usar as seguintes soluções para ajudá-lo a encontrar a origem dos problemas e corrigi-los.

Tópicos

- [Falha na exclusão da pilha](#)
- [Erro de dependência](#)
- [Conflitos do AWS Config e do AWS Systems Manager](#)
- [Erro ao analisar parâmetros ao passar uma lista](#)
- [Permissões insuficientes do IAM](#)
- [Valor inválido ou propriedade de recurso sem suporte](#)
- [Cota excedida](#)
- [As pilhas aninhadas estão paralisadas em UPDATE_COMPLETE_CLEANUP_IN_PROGRESS, UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS ou UPDATE_ROLLBACK_IN_PROGRESS](#)
- [Nenhuma atualização a ser executada](#)
- [Falha ao estabilizar recurso durante uma operação de criação, atualização ou exclusão de pilha](#)
- [O Grupo de segurança não existe na VPC](#)

- [Falha na reversão da atualização](#)
- [A condição de espera não recebeu o número necessário de sinais de uma instância do Amazon EC2](#)
- [Recurso removido da pilha, mas não excluído](#)

Falha na exclusão da pilha

Para resolver essa situação, tente o seguinte:

- Alguns recursos devem estar vazios para que possam ser excluídos. Por exemplo, você deve excluir todos os objetos de um bucket do Amazon S3 ou remover todas as instâncias de um grupo de segurança do Amazon EC2 para poder excluir o bucket ou o grupo de segurança.
- Certifique-se de que você tem as permissões necessárias do IAM para excluir os recursos da pilha. Além das permissões do CloudFormation, você deve ter permissão para usar os serviços subjacentes, como o Amazon S3 ou o Amazon EC2.
- Quando as pilhas estiverem no estado DELETE_FAILED porque o CloudFormation não conseguiu excluir um recurso, execute novamente a [exclusão](#) com o parâmetro `RetainResources` e especifique o recurso que o CloudFormation não pode excluir. O CloudFormation excluirá a pilha sem excluir o recurso retido. A retenção de recursos é útil quando você não pode excluir um recurso, como um bucket do S3 que contém objetos que você deseja manter, mas você ainda deseja excluir a pilha.

Depois de excluir a pilha, você pode excluir os recursos retidos manualmente usando o produto da AWS associado a eles.

Como alternativa, é possível considerar usar a opção `FORCE_DELETE_STACK` com o parâmetro `DeletionMode`. Para obter mais informações sobre a exclusão forçada de uma pilha, consulte [Excluir uma pilha no console do CloudFormation](#).

- Você não pode excluir pilhas que têm a proteção contra encerramento ativada. Se você tentar excluir uma pilha com proteção contra encerramento ativada, a exclusão falhará e a pilha, incluindo seu status, permanecerá inalterada. Desative a proteção contra encerramento na pilha e, em seguida, execute a operação de exclusão novamente.

Isso inclui [pilhas aninhadas](#) cujas pilhas raiz têm proteção contra encerramento ativada. Desative a proteção contra encerramento na pilha raiz e, em seguida, execute a operação de exclusão novamente. É altamente recomendável que você não exclua pilhas aninhadas diretamente, mas as exclua apenas como parte da exclusão da pilha raiz e de todos os recursos dela.

Para obter mais informações, consulte [Proteger pilhas do CloudFormation contra exclusão](#).

- Para todos os outros problemas, se você tiver AWS Support, poderá criar um caso de Suporte. Consulte [Entrar em contato com o suporte](#).

Erro de dependência

Para resolver um erro de dependência, adicione um atributo `DependsOn` aos recursos que dependem de outros recursos em seu modelo. Em alguns casos, você deve declarar dependências explicitamente para que o CloudFormation possa criar ou excluir recursos na ordem correta. Por exemplo, se você criar um IP elástico e uma VPC com um Internet gateway na mesma pilha, o IP elástico deverá depender do anexo do Internet Gateway. Para obter mais informações, consulte o atributo [DependsOn](#).

Conflitos do AWS Config e do AWS Systems Manager

O AWS Config e o AWS Systems Manager podem automatizar tarefas de gerenciamento de infraestrutura que podem causar conflitos com a implantação de uma pilha do CloudFormation. Faça o seguinte para evitar possíveis conflitos:

- Revise a configuração do AWS Config e do Systems Manager na Conta da AWS e Região da AWS associadas.
- Verifique se há regras ativas ou documentos de automação que possam ser acionados durante uma implantação do CloudFormation. Isso pode causar conflitos ou dependências de recursos que afetam sua implantação.
- Verifique seu modelo do CloudFormation para ver se há recursos gerenciados pelo AWS Config e pelo Systems Manager. Verifique possíveis sobreposições ou interdependências e considere ajustar o modelo ou a configuração de automação para evitar conflitos.
- Desative ou suspenda temporariamente quaisquer regras do AWS Config relacionadas ou automações do Systems Manager durante a implantação do CloudFormation. Lembre-se de restaurar as configurações originais após a implantação bem-sucedida para manter o nível desejado de automação e conformidade.
- Analise os registros e as mensagens de erro do CloudFormation em busca de qualquer referência ao AWS Config e problemas relacionados ao Systems Manager para ajudar a identificar a origem do conflito.

Para obter mais informações sobre regras do AWS Config, consulte [Análise de recursos com o Regras do AWS Config](#).

Para obter mais informações sobre as automações do Systems Manager, consulte [Automação do AWS Systems Manager](#).

Erro ao analisar parâmetros ao passar uma lista

Ao usar a AWS CLI ou o CloudFormation para passar uma lista, adicione o caractere de escape (\) antes de cada vírgula. O exemplo a seguir mostra como você especifica um parâmetro de entrada ao usar a AWS CLI.

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

Permissões insuficientes do IAM

Ao trabalhar com uma pilha do CloudFormation, você não precisa só de permissões para usar o CloudFormation, você também deve ter permissão para usar os serviços subjacentes que são descritos em seu modelo. Por exemplo, se estiver criando um bucket do Amazon S3 ou iniciando uma instância do Amazon EC2, você precisa de permissões para o Amazon S3 ou para o Amazon EC2. Revise sua política do IAM e verifique se você tem as permissões necessárias antes de trabalhar com pilhas do CloudFormation. Para obter mais informações, consulte [Controle o acesso ao CloudFormation com o AWS Identity and Access Management](#).

Valor inválido ou propriedade de recurso sem suporte

Ao criar ou atualizar uma pilha do CloudFormation, pode haver uma falha em sua pilha devido a parâmetros de entrada inválidos, nomes de propriedades de recursos sem suporte ou valores de propriedades de recursos sem suporte. Para parâmetros de entrada, verifique se o recurso existe. Por exemplo, quando você especifica um par de chaves do Amazon EC2 ou um ID de VPC, o recurso deve existir em sua conta e na região na qual você está criando ou atualizando a pilha. É possível usar [tipos de parâmetros específicos da AWS](#) para garantir o uso de valores válidos.

Para nomes e valores de propriedades de recursos, atualize seu modelo para usar nomes e valores válidos. Para obter mais informações, consulte [Referência de tipos de propriedade e recursos da AWS](#).

Cota excedida

Verifique se você não atingiu a cota de um recurso. Por exemplo, o número máximo padrão de instâncias do Amazon EC2 On-Demand que você pode executar é 5. Se tentar criar mais instâncias sob demanda do Amazon EC2 que a cota da sua conta, haverá falha na criação da instância e você receberá o erro `Status=start_failed`. Para visualizar as cotas padrão da AWS por serviço, consulte [Cotas de serviço da AWS](#), no Referência geral da AWS.

Para obter informações sobre cotas do CloudFormation e estratégias de ajuste, consulte [Entender as cotas do CloudFormation](#).

Além disso, durante uma atualização, se um recurso for substituído, o CloudFormation criará um novo recurso antes de excluir o antigo. Essa substituição pode colocar sua conta acima da quota de recursos, o que provoca uma falha na atualização. Você pode excluir o excesso de recursos ou solicitar um [aumento de cota](#).

As pilhas aninhadas estão paralisadas em **UPDATE_COMPLETE_CLEANUP_IN_PROGRESS**, **UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS** ou **UPDATE_ROLLBACK_IN_PROGRESS**

Falha na reversão de uma pilha aninhada. Devido às potenciais dependências de recursos entre pilhas aninhadas, o CloudFormation não inicia a limpeza de recursos de pilhas aninhadas até que todas as pilhas aninhadas tenham sido atualizadas ou revertidas. Quando há falha na reversão de uma pilha aninhada, o CloudFormation cancela todas as operações, independentemente do estado em que as outras pilhas aninhadas estão. Uma pilha aninhada que concluiu a atualização ou a reversão, mas não recebeu um sinal do CloudFormation para iniciar a limpeza porque houve uma falha na reversão de outras pilhas aninhadas, está em um estado **UPDATE_COMPLETE_CLEANUP_IN_PROGRESS** ou **UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS**. Uma pilha aninhada que não pôde ser atualizada, mas não recebeu um sinal para iniciar a reversão, está em um estado **UPDATE_ROLLBACK_IN_PROGRESS**.

Uma pilha aninhada pode não ser revertida devido a alterações que foram feitas fora do CloudFormation, quando o modelo da pilha não refletir precisamente o estado da pilha. Uma pilha aninhada também pode falhar se um grupo de Auto Scaling tinha tempo limite insuficiente para sinal de recurso quando o grupo foi criado ou atualizado.

Para corrigir a pilha, entre em contato com o [AWS Support](#).

Nenhuma atualização a ser executada

Para atualizar uma pilha do CloudFormation, você deve enviar as alterações do modelo ou do valor do parâmetro ao CloudFormation. No entanto, o CloudFormation não reconhece algumas alterações de modelo como uma atualização, como alterações em uma política de exclusão, política de atualização, declaração de condição ou declaração de saída. Se você precisar fazer essas alterações sem fazer nenhuma outra alteração, você poderá adicionar ou modificar um atributo metadata de qualquer um de seus recursos. O atributo de metadados pode ser qualquer valor arbitrário, pois o CloudFormation não interpreta seu conteúdo. Para obter mais informações, consulte o atributo [Metadata](#).

Falha ao estabilizar recurso durante uma operação de criação, atualização ou exclusão de pilha

Um recurso não respondeu porque a operação excedeu o tempo limite do CloudFormation ou um serviço da AWS foi interrompido. Para interrupções de serviços, [verifique](#) se o produto relevante da AWS está em execução e tente novamente a operação de pilha.

Se os produtos da AWS forem executados com êxito, verifique se sua pilha contém um dos seguintes recursos:

- `AWS::AutoScaling::AutoScalingGroup` para criar, atualizar e excluir operações
- `AWS::CertificateManager::Certificate` para criar operações
- `AWS::CloudFormation::Stack` para criar, atualizar e excluir operações
- `AWS::ElasticSearch::Domain` para atualizar operações
- `AWS::RDS::DBCluster` para criar e atualizar operações
- `AWS::RDS::DBInstance` para criar, atualizar e excluir operações
- `AWS::Redshift::Cluster` para atualizar operações

As operações para esses recursos podem demorar mais tempo que o tempo limite padrão. O tempo limite depende do recurso e das credenciais que você usa. Para estender o tempo limite, especifique uma [função de serviço](#) ao executar a operação de pilha. Se já estiver usando uma função de serviço, ou se a pilha contiver um recurso que não está listado, entre em contato com o [AWS Support](#).

Se a sua pilha estiver no estado `UPDATE_ROLLBACK_FAILED`, consulte [Falha na reversão da atualização](#).

O Grupo de segurança não existe na VPC

Verifique se o security group existe na VPC especificada. Se o security group existir, especifique o ID do security group e não o nome do security group. Por exemplo, o recurso `AWS::EC2::SecurityGroupIngress` tem as propriedades `SourceSecurityGroupName` e `SourceSecurityGroupId`. Para os security groups da VPC, você deve usar a propriedade `SourceSecurityGroupId` e especificar o ID do security group.

Falha na reversão da atualização

Um recurso dependente não pode retornar para o estado original, provocando falha na reversão (estado `UPDATE_ROLLBACK_FAILED`). Por exemplo, você pode ter uma pilha que está sendo revertida para uma instância de banco de dados antiga que foi excluída fora do CloudFormation. Como o CloudFormation não sabe que o banco de dados foi excluído, ele pressupõe que a instância de banco de dados ainda existe e tenta reverter, provocando falha na reversão da atualização.

Dependendo da causa da falha, você poderá corrigir o erro manualmente e continuar a reversão. Ao continuar a reversão, você pode retornar sua pilha para o estado de funcionamento (o estado `UPDATE_ROLLBACK_COMPLETE`) e, em seguida, tentar atualizar a pilha novamente. A lista a seguir descreve as soluções de erros comuns que provocam falhas na reversão da atualização:

- Falha ao receber o número necessário de sinais

Use o comando [signal-resource](#) para enviar manualmente o número necessário de sinais de êxito ao recurso que está esperando por eles e, em seguida, continue revertendo a atualização. Por exemplo, durante uma reversão de atualização, as instâncias em um grupo de Auto Scaling podem ter uma falha ao sinalizar êxito dentro do tempo limite especificado. Envie sinais de êxito manualmente ao grupo de Auto Scaling. Quando você continuar a reversão da atualização, o CloudFormation verá seus sinais e continuará com a reversão.

- As alterações em um recurso foram feitas fora do CloudFormation

Sincronize os recursos manualmente para que eles correspondam ao modelo da pilha original e continue a reversão da atualização. Por exemplo, se tiver excluído manualmente um recurso que o CloudFormation está tentando reverter, você deverá criar esse recurso manualmente com o mesmo nome e propriedades na pilha original.

- Permissões insuficientes

Verifique se você tem permissões suficientes do IAM para modificar os recursos e continue a reversão da atualização. Por exemplo, sua política de IAM pode permitir que você crie um bucket do S3, mas não que modifique o bucket. Adicione ações de modificação à sua política.

- Token de segurança inválido

O CloudFormation requer um novo conjunto de credenciais. Nenhuma alteração é necessária. Continue a reversão da atualização, o que atualiza as credenciais.

- Erro de limitação

Exclua recursos de que você não precisa ou solicite um [aumento da quota](#) e, em seguida, continue a reversão da atualização. Por exemplo, se a cota do número de instâncias sob demanda do EC2 da sua conta for 5 e a reversão da atualização exceder essa cota, haverá uma falha.

- O recurso não estabilizou

Um recurso não respondeu porque a operação excedeu o tempo limite do CloudFormation ou porque um serviço da AWS pode ter sido interrompido. Nenhuma alteração é necessária. Depois que operação do recurso for concluída ou que o produto da AWS entrar em operação novamente, continue revertendo a atualização.

Para continuar a reversão de uma atualização, você pode usar o console do CloudFormation ou a interface de linhas de comando da AWS (AWS CLI). Para obter mais informações, consulte [Continuar revertendo uma atualização](#).

Se nenhuma dessas soluções funcionar, você poderá ignorar os recursos que o CloudFormation não pode reverter com êxito. Para obter mais informações, consulte o parâmetro `ResourcesToSkip` para a operação da API [ContinueUpdateRollback](#) na Referência da API do AWS CloudFormation. O CloudFormation define o status dos recursos especificados como `UPDATE_COMPLETE` e continua a reverter a pilha. Após a reversão ser concluída, o estado dos recursos ignorados será inconsistente com o estado dos recursos no modelo de pilha. Antes de executar outra atualização da pilha, você deve modificar os recursos ou atualizar a pilha para estarem consistentes entre si. Caso contrário poderá haver falha em atualizações subsequentes da pilha, e a pilha poderá se tornar irrecoverável.

A condição de espera não recebeu o número necessário de sinais de uma instância do Amazon EC2

Para resolver essa situação, tente o seguinte:

- Verifique se a AMI que você está usando tem os scripts auxiliares do CloudFormation instalados. Se a AMI não incluir os scripts auxiliares, você também poderá baixá-los em sua instância. Para obter mais informações, consulte [Referência de scripts auxiliares do CloudFormation](#) no Guia de referência de modelos do CloudFormation.
- Verifique se o comando `cfn-signal` foi executado com êxito na instância. Você pode visualizar os logs, por exemplo, `/var/log/cloud-init.log` ou `/var/log/cfn-init.log`, para ajudá-lo a depurar a execução da instância. Você pode recuperar os logs fazendo login em sua instância, mas você deve [desabilitar a reversão em caso de falha](#) ou o CloudFormation excluirá a instância depois que houver falha na criação da pilha. Você também pode [publicar os logs](#) no Amazon CloudWatch. No Windows, você pode visualizar os logs do `cfn` em `C:\cfn\log` e os logs do serviço EC2Config em `%ProgramFiles%\Amazon\EC2ConfigService`.
- Verifique se a instância tem uma conexão com a Internet. Se estiver em uma VPC, a instância deverá poder se conectar à Internet por meio de um dispositivo NAT se ela estiver em uma sub-rede privada ou por meio de um Internet gateway se estiver em uma sub-rede pública. Para testar a conexão da instância com a Internet, tente acessar uma página da web pública, como `http://aws.amazon.com`. Por exemplo, você pode executar o seguinte comando na instância. Ele deve retornar um código de status HTTP 200.

```
curl -I https://aws.amazon.com
```

Para obter informações sobre como configurar um dispositivo NAT, consulte [NAT](#) no Guia do usuário da Amazon VPC.

Recurso removido da pilha, mas não excluído

Durante uma atualização de pilha, o CloudFormation removeu um recurso de uma pilha, mas não excluiu o recurso. O recurso ainda existe, mas não é mais acessível por meio do CloudFormation. Isso pode ocorrer durante atualizações de pilha nas quais:

- O CloudFormation precisa substituir um recurso existente, então primeiramente ele cria um novo recurso e, em seguida, tenta excluir o recurso antigo.

- Você removeu o recurso do modelo de pilha, portanto, o CloudFormation tentará excluir o recurso da pilha.

No entanto, pode haver casos em que o CloudFormation não poderá excluir o recurso. Por exemplo, se o usuário não tiver permissões para excluir um recurso de um determinado tipo.

O CloudFormation tentará excluir o recurso antigo três vezes. Se o CloudFormation não puder excluir o recurso antigo, ele removerá o recurso antigo da pilha e continuará a atualizá-la.

Quando a atualização da pilha estiver concluída, o CloudFormation emitirá um evento de pilha `UPDATE_COMPLETE`, mas incluirá um `StatusReason` que declarará que não foi possível excluir um ou mais recursos. O CloudFormation também emite um evento `DELETE_FAILED` para o recurso específico, com um `StatusReason` correspondente fornecendo mais detalhes sobre o motivo pelo qual o CloudFormation não conseguiu excluir o recurso.

Para solucionar essa situação, exclua o recurso diretamente usando o console ou a API do serviço em questão.

Entrar em contato com o suporte

Se você tiver AWS Support, poderá criar um caso de suporte técnico em <https://console.aws.amazon.com/support/home#/>. Antes de entrar em contato com o suporte, reúna as seguintes informações:

- O ID da pilha. Você pode encontrar o ID da pilha na guia Visão geral do [console do CloudFormation](#). Para obter mais informações, consulte [Visualizar informações da pilha no console do CloudFormation](#).

Important

Não faça alterações nos recursos da pilha fora do CloudFormation. Fazer alterações nos recursos da pilha fora do CloudFormation pode colocar a pilha em um estado irrecoverável.

- Todas as mensagens de erro da pilha. Para obter informações sobre como visualizar as mensagens de erro da pilha, consulte a seção [Guia de solução de problemas](#).
- Para problemas do Amazon EC2, obtenha os logs do `cloud-init` e do `cfn`. Esses logs são publicados na instância do Amazon EC2, no diretório `/var/log/`. Esses logs capturam processos e saídas de comandos enquanto sua instância está sendo configurada. No Windows, colete os

logs do serviço EC2Configure e do cfn em %ProgramFiles%\Amazon\EC2ConfigService e C:\cfn\log.

Também é possível pesquisar respostas e postar perguntas na [Comunidade do CloudFormation](#) no AWS re:Post.

Solucionar problemas de implantações malsucedidas de pilhas do CloudFormation com o Amazon Q Developer

O Amazon Q Developer é um assistente conversacional baseado em inteligência artificial (IA) generativa que pode ajudar você a entender, criar, estender e operar aplicações da AWS. Para obter informações sobre o Amazon Q Developer, consulte o [Amazon Q Developer User Guide](#).

Este documento descreve como usar o Amazon Q Developer para ajudar a solucionar problemas de implantações malsucedidas do CloudFormation. Esse atributo foi criado para ajudar a identificar e resolver rapidamente os problemas encontrados durante a implantação de pilhas.

Com tecnologia do Amazon Bedrock: o Amazon Q é baseado no Amazon Bedrock e inclui [detecção automatizada de abuso](#) implementada no Amazon Bedrock para garantir segurança, proteção e uso responsável da IA.

Tópicos

- [Recursos](#)
- [Como funciona](#)
- [Preços](#)
- [Disponibilidade regional](#)
- [Limitações](#)

Recursos

A solução para resolver problemas de implantações de pilhas com o Amazon Q Developer fornece os seguintes atributos:

- **Análise de erros:** fornece uma análise fácil de entender dos erros de implantação de pilha do CloudFormation.
- **Sugestões de resolução:** oferece instruções passo a passo para resolver os problemas identificados.

Como funciona

Quando há falha na criação ou na atualização de uma pilha do CloudFormation, uma mensagem de erro é exibida no Console de gerenciamento da AWS.

Você pode escolher o botão Diagnosticar com Q ao lado da mensagem de erro.

O Amazon Q Developer analisa o erro e fornece uma explicação do problema.

Para obter assistência adicional, escolha Ajude-me a resolver para receber as etapas detalhadas da resolução.

Preços

Para obter informações sobre preços, consulte [Amazon Q Developer pricing](#).

Disponibilidade regional

O Amazon Q Developer está disponível na maioria das Regiões da AWS. Para obter uma lista de todas as regiões em que o Amazon Q Developer está disponível atualmente, consulte [Supported Regions for Amazon Q Developer](#) no Amazon Q Developer User Guide.

Limitações

- Esse atributo pode ser acessado somente pelo Console de gerenciamento da AWS.
- Esse atributo cobre os vinte erros mais comuns. Ele não cobre todos os erros possíveis.
- Casos complexos de solução de problemas ainda podem exigir a intervenção do Suporte.

Histórico do documento para o Guia do usuário do AWS CloudFormation

A tabela a seguir descreve alterações importantes no Guia do usuário do AWS CloudFormation após maio de 2018. Para receber notificações sobre as atualizações da documentação, é possível se tornar assinante de um feed do RSS.

Important

As linhas da tabela que descrevem as atualizações no conteúdo de referência do modelo a partir de maio de 2018 foram movidas para a nova Referência de modelos do AWS CloudFormation. Para essas atualizações e quaisquer alterações futuras, consulte o [Histórico do documento da Referência de modelos do AWS CloudFormation](#) na Referência de modelos do AWS CloudFormation.

Alteração	Descrição	Data
Guia em PDF disponível	Agora é possível baixar o Guia do usuário do AWS CloudFormation em PDF.	30 de maio de 2025
O conteúdo de referência de modelos foi movido para um novo guia	O CloudFormation publicou o Guia de referência de modelos do CloudFormation. Para obter detalhes, consulte o Guia de referência de modelos do CloudFormation .	30 de maio de 2025
O gerador de IaC é compatível com a verificação parcial	Agora, você pode escolher tipos específicos de recursos para procurar, o que facilita a geração de modelos de infraestrutura como código (IaC) nos recursos existentes. Para mais informações,	27 de março de 2025

consulte [Iniciar uma verificação de recursos com o gerador de IaC do CloudFormation](#).

[Refatoração de pilhas](#)

A refatoração de pilhas simplifica a reorganização dos recursos em suas pilhas do CloudFormation e, ao mesmo tempo, preserva as propriedades e os dados dos recursos existentes. Para obter mais informações, consulte [Refatoração de pilhas](#).

6 de fevereiro de 2025

[Solucionar problemas de implantações de pilhas com o Amazon Q Developer](#)

Agora é possível usar o Amazon Q Developer para solucionar erros comuns ao implantar pilhas do CloudFormation. Para obter mais informações, consulte [Solucionar problemas de implantações malsucedidas de pilhas do CloudFormation com o Amazon Q Developer](#).

22 de novembro de 2024

[Cronograma de implantação da pilha](#)

Agora você pode ver uma representação visual da implantação da sua pilha. O gráfico do cronograma da implantação da pilha mostra o status da implantação da pilha, os status das implantações dos recursos individuais e quando os status das implantações foram alterados. Para obter mais informações, consulte [Visualizar um cronograma de implantação de uma pilha do CloudFormation](#).

11 de novembro de 2024

[Visualizar seus recursos verificados e modelos gerados](#)

Agora você pode simplificar os fluxos de trabalho do gerador de Infraestrutura como Código (IaC) visualizando os detalhes resumidos da verificação e visualizando os modelos gerados antes de implantar sua pilha de infraestrutura. Para obter mais informações, consulte [Visualizar o resumo da verificação no console do CloudFormation](#) e [Criar uma pilha do CloudFormation a partir dos recursos verificados](#).

22 de agosto de 2024

[Integração do Amazon EventBridge com a sincronização do Git do AWS CloudFormation](#)

A sincronização do Git do AWS CloudFormation agora publica alterações de status de sincronização como eventos no Amazon EventBridge. Consulte mais informações em [Repository Sync Status Change event detail](#) e [Resource Sync Status Change event detail](#).

29 de julho de 2024

[Forçar a exclusão de pilhas paralisadas](#)

Duas novas opções para forçar a exclusão de pilhas estão disponíveis para operações de exclusão de pilhas paralisadas. Agora você pode optar por forçar a exclusão da pilha, mas reter o recurso, ou forçar a exclusão da pilha inteira. Para obter mais informações, consulte [Excluir uma pilha no console do CloudFormation](#).

22 de maio de 2024

[AWS CloudTrail Causas-raiz da operação de pilha de eventos do](#)

O CloudFormation aprimora a experiência de solução de problemas para operações de pilha com uma nova integração de link direto do AWS CloudTrail. Esse recurso vincula diretamente eventos de operação de pilha no console do CloudFormation a eventos relevantes do CloudTrail. Para obter mais informações, consulte [Determinar a causa de uma falha na pilha.](#)

15 de maio de 2024

[Conjuntos de alterações no nível de propriedade](#)

Os conjuntos de alterações no nível da propriedade permitem visualizar as alterações que as implantações do CloudFormation farão nos valores das propriedades dos recursos. Para mais informações, consulte [Visualizar um conjunto de alterações para uma pilha do CloudFormation.](#)

12 de abril de 2024

[CloudFormation apresenta o evento CONFIGURATION_COMPLETE](#)

Você agora pode usar o evento CONFIGURATION_COMPLETE para permitir fluxos de trabalho mais rápidos envolvendo a criação de recursos. Para obter mais informações, consulte [Entender os eventos de criação de pilhas do CloudFormation.](#)

11 de março de 2024

[Gerar modelos do AWS CloudFormation e aplicações do AWS CDK baseados em recursos da AWS existentes](#)

Agora é possível gerar um modelo usando recursos provisionados em sua conta que ainda não são gerenciados pelo CloudFormation. Para obter mais informações, consulte [Gerar modelos com base em recursos existentes com o gerador de IaC](#).

2 de fevereiro de 2024

[Modo de simultaneidade do StackSets](#)

O Modo de simultaneidade é um parâmetro para `StackSetOperationPreferences` que permite a você escolher como o nível de simultaneidade se comporta durante as operações de conjunto de pilhas. Para obter mais informações, consulte [Escolher o modo de simultaneidade para o CloudFormation StackSets](#).

9 de novembro de 2023

[Informações detalhadas sobre desvios do StackSet](#)

As APIs a seguir permitem a você ver quais instâncias de pilha foram retiradas do modelo do StackSet e quais recursos foram desviados.

24 de julho de 2023

[ListStackInstanceResourceDrifts](#)

Retorna informações de desvio para recursos em uma instância de pilha.

[StackInstanceResourceDriftsSummary](#)

A estrutura que contém informações resumidas sobre desvios de recursos para uma instância de pilha.

[APIs do CloudFormation
StackSets para controlar o
acesso confiável ao AWS
Organizations](#)

O CloudFormation StackSets fornece aos clientes as seguintes APIs para gerenciar o acesso confiável ao AWS Organizations:

5 de junho de 2023

[ActivateOrganizationsAccess](#)

Ativa o acesso confiável com AWS Organizations. Com o acesso confiável entre o StackSets e o Organizations ativado, a conta de gerenciamento tem permissões para criar e gerenciar StackSets para sua organização.

[DeactivateOrganizationsAccess](#)

Desativa o acesso confiável com AWS Organizations. Se o acesso confiável for desativado, a conta de gerenciamento não terá permissões para criar e gerenciar StackSets gerenciados por serviços para sua organização.

[DescribeOrganizationsAccess](#)

Recupera informações sobre o status `OrganizationAccess` da conta. Esta API pode ser chamada pela conta de gerenciamento ou pelo

administrador delegado usando o parâmetro `CallAs`. Esta API também pode ser chamada sem o parâmetro `CallAs` pela conta de gerenciamento.

[DescribeStackSet API](#)

A API `DescribeStackSet` tem um novo parâmetro para a lista de regiões em que um determinado conjunto de pilhas é implantado. Para obter mais informações, consulte [DescribeStackSet](#).

1º de fevereiro de 2023

[Gerenciar eventos do StackSets com o CloudFormation e o Amazon EventBridge](#)

O CloudFormation StackSets inicia notificações de eventos via Amazon EventBridge. É possível acionar ações orientadas por eventos após criar, atualizar ou excluir seus conjuntos de pilhas do CloudFormation. Para obter mais informações, consulte [Monitorar eventos de sincronização do CloudFormation e do Git com o EventBridge](#).

16 de novembro de 2022

[Insights aprimorados sobre instâncias de pilha para operações de conjuntos de pilhas](#)

O CloudFormation StackSets fornece informações mais detalhadas sobre instâncias de pilha para operações de conjuntos de pilhas:

4 de novembro de 2022

[DescribeStackSetOperation](#)

Agora é possível usar `DescribeStackSetOperation` para fornecer a contagem de instâncias de pilha com falha para operações de conjunto de pilhas durante a implantação.

[ListStackInstances](#)

Agora é possível usar a opção de filtragem `LastOperationID` para listar instâncias de pilha para operações de conjunto de pilhas.

[Gerenciar eventos com o CloudFormation e o Amazon EventBridge](#)

Receba notificações quando eventos específicos do CloudFormation ocorrerem. Para obter mais informações, consulte [Monitorar eventos de sincronização do CloudFormation e do Git com o EventBridge](#).

20 de julho de 2022

Nível de conta

O CloudFormation anuncia a disponibilidade geral do tipo de filtro de conta, um recurso que permite que os clientes limitem os destinos da implantação a contas individuais ou incluam contas adicionais nas UOs fornecidas. Para obter mais informações, consulte [Destinos em nível de conta para StackSets gerenciados pelo serviço](#).

7 de julho de 2022

Registro do CloudFormation

O CloudFormation anuncia a disponibilidade geral dos Hooks, um recurso que permite que os clientes invoquem uma lógica personalizada a fim de automatizar ações ou inspecionar configurações de recursos antes de uma operação para criar, atualizar ou excluir pilhas. Para obter mais informações, consulte o [CloudFormation Hooks User Guide](#).

10 de fevereiro de 2022

[Stack failure options \(Opções de falha da pilha\)](#)

Você pode desenvolver suas aplicações de maneira iterativa quando falhas de provisionamento são detectadas, iniciando desde o ponto de falha sem precisar reverter os recursos provisionados com êxito. Especificando opções de falha de pilha, você pode solucionar problemas de recursos em um status `CREATE_FAILED` ou `UPDATE_FAILED`. É possível especificar opções de falha de provisionamento de todas as implantações de pilha e operações de conjunto de alterações. Para obter mais informações, consulte [Escolher como lidar com falhas ao provisionar recursos](#).

30 de agosto de 2021

[Importar pilhas para o conjunto de pilhas](#)

Agora é possível importar pilhas existentes para conjuntos de pilhas novos ou existentes. Para obter mais informações, consulte [Importar pilhas para o CloudFormation StackSets](#).

28 de julho de 2021

[Aumento da cota](#)

Agora, é possível declarar um máximo de 2000 pilhas padrão na sua Conta da AWS. Para obter mais informações, consulte [Entender as cotas do CloudFormation](#).

15 de julho de 2021

[Publicar extensões públicas de terceiros](#)

Agora é possível usar extensões públicas fornecidas por editores terceirizados da mesma forma que você faria com extensões da AWS. Para obter mais informações, consulte [Usar extensões públicas de terceiros do registro do CloudFormation](#).

21 de junho de 2021

[Macros de referência em modelos de conjunto de pilha](#)

O StackSets agora oferece suporte à criação ou atualização de conjuntos de pilhas com permissões autogerenciadas de modelos que fazem referência a macros. Para obter mais informações sobre macros, consulte [Executar processamento personalizado em modelos do CloudFormation com macros de modelo](#).

14 de abril de 2021

[Usar o valor mais recente de um parâmetro do Systems Manager em uma referência dinâmica](#)

Agora é possível fazer o CloudFormation usar a versão mais recente de um parâmetro do Systems Manager sempre que você criar ou atualizar uma pilha. Você não precisa mais especificar uma versão específica. Para obter mais detalhes, consulte [Obter um valor em texto simples do Systems Manager Parameter Store](#).

13 de abril de 2021

[Módulos aceitam o uso de delimitadores de período em nomes de recursos](#)

Agora você pode usar um período como um delimitador ao especificar o nome lógico totalmente qualificado para um recurso contido em um módulo. Para obter mais informações, consulte [Fazer referência a recursos de módulos em modelos do CloudFormation](#).

8 de abril de 2021

[O CloudFormation StackSets agora oferece suporte à implantação em regiões em paralelo](#)

Agora você pode optar por implantar o StackSets em regiões sequencialmente ou em paralelo. Para obter mais informações, consulte [Opções de operação do conjunto de pilha](#).

6 de abril de 2021

[O CloudFormation StackSets agora oferece suporte a administrador delegado com o AWS Organizations](#)

Além da conta de gerenciamento da organização, as contas de administrador delegadas podem criar e gerenciar conjuntos de pilha com permissões gerenciadas por serviço para sua organização. Para obter mais informações, consulte [Registrar uma conta de membro de administrador delegado](#) e [Criar CloudFormation StackSets com permissões gerenciadas pelo serviço](#).

18 de fevereiro de 2021

[Disponibilidade do CloudFormation StackSets nas regiões](#)

O CloudFormation StackSets agora está disponível na região Ásia-Pacífico (Osaka). Para obter mais informações, consulte [Gerenciar pilhas em todas as contas e regiões com o StackSets](#).

10 de fevereiro de 2021

[Módulos](#)

Os módulos são uma maneira de empacotar configurações de recursos para inclusão entre modelos de pilha, de modo transparente, gerenciável e repetível. Os módulos podem encapsular configurações comuns de serviço e práticas recomendadas como blocos de construção modulares e personalizáveis para você incluir em seus modelos de pilha. Para obter mais informações, consulte [Criar configurações de recursos reutilizáveis que podem ser incluídas em modelos diferentes com os módulos do CloudFormation](#).

24 de novembro de 2020

[Conjuntos de alteração para pilhas aninhadas](#)

Com os conjuntos de alterações para pilhas aninhadas, você pode visualizar as alterações nos recursos de suas aplicações e infraestrutura em toda a hierarquia de pilha aninhada, dando continuidade às atualizações após confirmar que todas as alterações estejam conforme pretendido. Para mais informações, consulte [Conjuntos de alterações para pilhas aninhadas](#).

18 de novembro de 2020

Aumento das cotas

As cotas do AWS CloudFormation a seguir foram atualizadas.

22 de outubro de 2020

- Agora é possível declarar um máximo de 200 mapeamentos no modelo do AWS CloudFormation.
- Agora você pode declarar um máximo de 200 atributos de mapeamento para cada mapeamento no modelo do AWS CloudFormation.
- Agora é possível declarar um máximo de 200 saídas no modelo do AWS CloudFormation
- Agora você pode declarar um máximo de parâmetros 200 no modelo do AWS CloudFormation.
- Agora é possível declarar um máximo de 500 recursos no modelo do AWS CloudFormation.
- Agora você pode transmitir um corpo de modelo com um tamanho máximo de 1 MB em um objeto do Amazon S3.

[Detecção de desvio para recursos privados](#)

O CloudFormation agora oferece suporte a operações de detecção de desvios em uma lista expandida de recursos da AWS, bem como recursos privados definidos como provisionáveis no registro do CloudFormation. Para obter mais informações, consulte [Suporte a tipos de recursos](#).

1.º de outubro de 2020

[Permissões atualizadas necessárias para registrar provedores de recursos](#)

Registrar um provedor de recursos em sua conta agora requer permissão para acessar o pacote do manipulador de esquema carregado em um bucket do S3 para esse provedor de recursos. Para obter mais informações, consulte [Permissões do IAM para registrar uma extensão privada de terceiros](#).

7 de agosto de 2020

[A importação de recursos oferece suporte a tipos de recursos privados provisionáveis](#)

As operações de importação agora oferecem suporte a tipos de recursos privados que são provisionáveis, ou seja, com tipo de provisionamento FULLY_MUTABLE ou IMMUTABLE . Para obter mais informações, consulte [Recursos que oferecem suporte a operações de importação](#).

3 de junho de 2020

[Implantações azul/verde do ECS por meio do CodeDeploy](#)

Agora é possível usar o CloudFormation para executar implantações azul/verde do ECS por meio do CodeDeploy. As implantações azul/verde são uma estratégia de implantação segura fornecida pelo AWS CodeDeploy para minimizar interrupções causadas pela alteração das versões da aplicação. Para obter mais informações, consulte [Executar implantações azul/verde do ECS por meio do CodeDeploy usando o CloudFormation](#).

19 de maio de 2020

[Disponibilidade do CloudFormation StackSets nas regiões](#)

O CloudFormation StackSets agora está disponível na região GovCloud (EUA-Oeste) da AWS.

18 de maio de 2020

[O AWS CloudFormation StackSets se integra a AWS Organizations](#)

Agora é possível usar o StackSets para gerenciar centralmente implantações em todas as contas em sua organização ou unidades organizacionais (UOs) específicas no AWS Organizations. Você também pode habilitar implantações automáticas em quaisquer novas contas adicionadas à sua organização ou às suas UOs. As permissões necessárias para implantação entre contas serão automaticamente tratadas pelo StackSets. Para obter mais informações, consulte [Gerenciar pilhas em todas as contas e regiões com o StackSets](#).

11 de fevereiro de 2020

[Deteção de desvios para StackSets](#)

Agora é possível executar a deteção de desvio em um conjunto de pilha e todas as instâncias de pilha incluídas nele. Para obter mais informações, consulte [Detectar desvios no CloudFormation StackSets](#).

19 de novembro de 2019

[Registro do CloudFormation agora disponível](#)

Agora é possível usar o console do CloudFormation para visualizar recursos públicos e privados que estão disponíveis para uso em sua conta. Para obter mais informações, consulte [Visualizar as extensões disponíveis e ativadas no registro do CloudFormation](#).

18 de novembro de 2019

[Ações da API de registro do CloudFormation](#)

18 de novembro de 2019

As ações de API para gerenciamento de tipos no registro do CloudFormation a seguir estão agora disponíveis.

[DeregisterType](#)

Remove um tipo ou uma versão de tipo do uso ativo no registro do CloudFormation.

[DescribeType](#)

Retorna informações detalhadas sobre um tipo registrado.

[DescribeTypeRegistration](#)

Retorna informações sobre o registro de um tipo, incluindo seu status atual e identificadores de tipo e versão.

[ListTypeRegistrations](#)

Retorna uma lista de identificadores de solicitação de registro para o tipo especificado.

[ListTypes](#)

Retorna informações resumidas sobre os tipos que foram registrados no CloudFormation.

[ListTypeVersions](#)

Retorna informações resumidas sobre as versões de um tipo.

[RegisterType](#)

Registra um tipo no registro do CloudFormation. Registrar um tipo o torna disponível para uso em modelos do CloudFormation em sua Conta da AWS.

[SetTypeDefaultVersion](#)

Especifique a versão padrão de um tipo. A versão padrão de um tipo será usada nas operações do CloudFormation.

Para obter mais informações sobre o registro do CloudFormation, consulte [Gerenciar extensões com o registro do CloudFormation](#)

[Importação de recursos adicionada](#)

Se você criou um recurso da AWS fora do gerenciamento do CloudFormation, é possível trazer esse recurso existente para o gerenciamento do CloudFormation usando o `resource import`. Para obter mais informações, consulte [Importar recursos da AWS para uma pilha do CloudFormation com uma importação de recursos](#).

11 de novembro de 2019

[Aumentos de limite do conjunto de pilhas](#)

Agora você pode criar um máximo de 100 conjuntos de pilhas em sua conta de administrador, criar um máximo de 2.000 instâncias de pilha por conjunto de pilhas e executar um máximo de 3.500 operações de instância de pilha em cada região ao mesmo tempo, por conta de administrador. Para obter mais informações, consulte [Entender as cotas do CloudFormation](#).

2 de agosto de 2019

Limite de recursos nas operações de pilha simultâneas

O CloudFormation agora impõe um limite para o número de recursos em operações de pilha simultâneas. Esse limite é determinado pela região. Para obter mais informações, consulte [Entender as cotas do CloudFormation](#).

30 de abril de 2019

Limite de operação de instâncias de pilha

Para StackSets, agora é possível ter no máximo 1500 operações de instância de pilha em execução em uma determinada região ao mesmo tempo, por conta de administrador. Para obter mais informações, consulte [Entender as cotas do CloudFormation](#).

13 de dezembro de 2018

A funcionalidade CAPABILITY_AUTO_EXPAND agora está disponível

Agora é possível usar a funcionalidade CAPABILITY_AUTO_EXPAND para criar ou atualizar uma pilha diretamente de um modelo de pilha que contém macros, sem primeiro analisar as alterações resultantes em um conjunto de alterações. Para obter mais informações, consulte [CreateStack](#) ou [UpdateStack](#) na Referência da API do AWS CloudFormation.

7 de dezembro de 2018

[Detecção de oscilação de pilha adicionada](#)

Agora é possível detectar se a configuração real de uma pilha se desviou da configuração de modelo esperada, conforme definido no CloudFormation. É possível detectar desvios em uma pilha inteira ou em recursos de pilha individuais. Para obter mais informações, consulte [Detectar alterações de configuração não gerenciadas em pilhas e recursos com a detecção de desvios](#).

13 de novembro de 2018

[A referência dinâmica secretsmanager agora está disponível](#)

É possível usar a referência a dinâmica `secretsmanager` para recuperar todos os segredos ou valores secretos armazenados no AWS Secrets Manager para uso em modelos. Os segredos podem ser credenciais de banco de dados, senhas, chaves de API de terceiros e até mesmo texto arbitrário. Usar a referência dinâmica `secretsmanager` garante que nem o Secrets Manager nem o CloudFormation registre ou persista qualquer valor secreto resolvido. Para obter mais informações, consulte [Obter um segredo ou valor de segredo do Secrets Manager](#).

9 de novembro de 2018

[Macros agora disponíveis](#)

É possível usar macros para executar o processamento personalizado em modelos, desde ações simples, como operações de localizar e substituir, até transformações extensas de modelos inteiros. Para obter mais informações, consulte [Executar processamento personalizado em modelos do CloudFormation com macros de modelo](#).

6 de setembro de 2018

[O CloudFormation agora oferece suporte a endpoints da VPC baseados em PrivateLink](#)

Você pode usar um endpoint da VPC para criar uma conexão privada entre sua VPC e o CloudFormation sem exigir acesso pela Internet, por meio de uma instância NAT, uma conexão VPN ou do Direct Connect. Para obter mais informações, consulte [Acessar o CloudFormation por meio de um endpoint de interface \(AWS PrivateLink\)](#).

22 de agosto de 2018

[Referências dinâmicas oferecem suporte a strings seguras](#)

Agora é possível usar novas referências dinâmicas para especificar valores que são armazenados e gerenciados em outros serviços, incluindo parâmetros do tipo `SecureString` do Systems Manager Parameter Store, em seus modelos de pilha. Para obter mais informações, consulte [Obter um valor de string segura do Systems Manager Parameter Store](#).

16 de agosto de 2018

[Os conjuntos de pilha agora oferecem suporte a funções de execução personalizadas](#)

Agora é possível usar funções de execução personalizadas em contas de destino para controlar os recursos de pilha que os usuários ou grupos podem incluir em seus conjuntos de pilhas. Para obter mais informações, consulte [Definir opções de permissões avançadas para operações de conjuntos de pilhas](#).

30 de maio de 2018

[Atualizações seletivas de instâncias de pilha](#)

Agora é possível usar os parâmetros opcionais `Accounts` e `Regions` para especificar as contas e regiões para atualizar instâncias de pilha durante uma operação de atualização do conjunto de pilhas. Para obter mais informações, consulte [UpdateStackSet](#) na Referência de APIs do CloudFormation.

30 de maio de 2018

[O CloudFormation agora cria buckets do S3 com criptografia habilitada](#)

Para buckets do Amazon S3 criados pelo AWS CloudFormation para armazenar modelos de pilha carregados, a criptografia no lado do servidor agora está ativada por padrão, criptografando assim todos os objetos armazenados nesses buckets. Para obter mais informações, consulte [Criar uma pilha via console do CloudFormation](#).

24 de maio de 2018

[Adição de endpoints do FIPS](#)

O CloudFormation passou a oferecer novos endpoints que usam módulos de criptografia com validação FIPS 140-2 nas seguintes regiões públicas dos EUA: Leste dos EUA 1, Leste dos EUA 2, Oeste dos EUA 1 e Oeste dos EUA 2. Consulte [Endpoints e cotas do CloudFormation](#) no Referência geral da Amazon Web Services para obter os novos URLs de endpoints compatíveis com FIPS.

17 de maio de 2018

Para atualizações do Guia do usuário do CloudFormation Hooks, consulte o [Histórico do documentos do Guia do usuário do CloudFormation Hooks](#) no Guia do usuário do CloudFormation Hooks.

Atualizações arquivadas

A tabela a seguir descreve alterações importantes em cada versão do Guia do usuário do AWS CloudFormation antes de maio de 2018.

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	22 de julho de 2019	Use a propriedade <code>encryptionOptions</code> para especificar uma Chave pertencente à AWS ou uma chave gerenciada pelo cliente para operadores do Amazon MQ.	15/05/2010
Convenção de nomenclatura do	10 de abril de 2018	As pilhas do CloudFormation criadas usando conjuntos de pilhas agora seguem uma nova convenção de nomenclatura em que o nome da pilha contém o nome do conjunto de pilhas.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
conjunto de pilhas			
Novos recursos	10 de abril de 2018	AWS::AppSync::ApiKey Use o recurso <code>AWS::AppSync::ApiKey</code> para criar uma chave exclusiva que pode ser distribuída aos clientes que estão executando operações em GraphQL com o AWS AppSync. AWS::AppSync::DataSource Use o recurso <code>AWS::AppSync::DataSource</code> para criar fontes de dados para os resolvedores no AWS AppSync. AWS::AppSync::GraphQLApi Use o recurso <code>AWS::AppSync::GraphQLApi</code> para criar uma nova API GraphQL do AWS AppSync. AWS::AppSync::GraphQLSchema Use o recurso <code>AWS::AppSync::GraphQLSchema</code> para criar o modelo de dados para a API GraphQL do AWS AppSync. AWS::AppSync::Resolver Use o recurso <code>AWS::AppSync::Resolver</code> para definir o resolvedor GraphQL lógico que você anexará aos campos em um esquema.	15/05/2018

Alteração	Data de lançamento	Descrição	Versão da API
Recurso atualizado	10 de abril de 2018	AWS::Config::ConfigurationAggregator Use o tipo de propriedade <code>OrganizationAggregationSource</code> para especificar as regiões dos dados do AWS Config a serem agregadas em um agregador de configuração do AWS Config e a função do IAM a ser usada para recuperar os detalhes do AWS Organizations.	15/05/2010
Novos recursos	4 de abril de 2018	AWS::Config::AggregationAuthorization Use o recurso <code>AWS::Config::AggregationAuthorization</code> para conceder a permissão a uma conta de agregador para coletar os dados do AWS Config. AWS::Config::ConfigurationAggregator Use o recurso <code>AWS::Config::ConfigurationAggregator</code> para criar um agregador de configuração para o AWS Config.	15/05/2010
Os conjuntos de pilha agora oferecem suporte a funções de administrador personalizadas	29 de março de 2018	Use as funções de administrador personalizadas para controlar quais usuários ou grupos podem gerenciar conjuntos de pilhas específicas dentro da mesma conta de administrador. Para obter mais informações, consulte Definir opções de permissões avançadas para operações de conjuntos de pilhas .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	29 de março de 2018	AWS::EC2::LaunchTemplate Use o recurso <code>AWS::EC2::LaunchTemplate</code> para criar um modelo de execução para uma instância do Amazon EC2.	15/05/2010
Recursos atualizados	29 de março de 2018	AWS::AutoScaling::AutoScalingGroup Use a propriedade <code>LaunchTemplate</code> para especificar o modelo de execução a ser usado para executar as instâncias. AWS::EC2::SpotFleet No tipo de propriedade SpotFleetRequestConfigData , use a propriedade <code>LaunchTemplateConfigs</code> para descrever um modelo de execução e substituições.	15/05/2010
Nova função intrínseca <code>Fn::Cidr</code>	6 de março de 2018	Devolve o bloco do endereço Cidr especificado. Para obter mais informações, consulte Fn::Cidr .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	6 de março de 2018	AWS::ApiGateway::VpcLink Use o recurso <code>AWS::ApiGateway::VpcLink</code> para especificar um link de VPC da API Gateway para um <code>AWS::ApiGateway::RestApi</code> acessar recursos em uma Amazon Virtual Private Cloud (VPC). AWS::GuardDuty::Master Use o recurso <code>AWS::GuardDuty::Master</code> para criar uma conta primária do GuardDuty. AWS::GuardDuty::Member Use o recurso <code>AWS::GuardDuty::Member</code> para criar uma conta-membro do GuardDuty. AWS::SES::ConfigurationSet Use o recurso <code>AWS::SES::ConfigurationSet</code> para criar grupos de regras que você pode aplicar aos e-mails que envia. AWS::SES::ConfigurationSetEventDestination Use o recurso <code>AWS::SES::ConfigurationSetEventDestination</code> para especificar um destino de evento do conjunto de configurações. AWS::SES::ReceiptFilter Use o recurso <code>AWS::SES::ReceiptFilter</code> para especificar se você deseja aceitar ou rejeitar e-mails provenientes de um endereço IP ou um intervalo de endereços IP.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::SES::ReceiptRule Use o recurso <code>AWS::SES::ReceiptRule</code> para especificar quais ações o Amazon SES deve realizar ao receber e-mails em nome de um ou mais endereços ou domínios de e-mail que você possui. AWS::SES::ReceiptRuleSet Use o recurso <code>AWS::SES::ReceiptRuleSet</code> para especificar um conjunto de regra vazio para o Amazon SES. AWS::SES::Template Use o recurso <code>AWS::SES::Template</code> para especificar o conteúdo do e-mail (composto de uma linha de assunto, uma parte em HTML e uma parte somente texto).	

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	6 de março de 2018	AWS::AutoScaling::AutoScalingGroup Use a propriedade <code>AutoScalingGroupName</code> para especificar o nome do grupo de Auto Scaling. AWS::ApiGateway::RestApi Use a propriedade <code>ApiKeySourceType</code> para especificar a origem da chave de API para as solicitações de medição de acordo com um plano de uso. Use a propriedade <code>MinimumCompressionSize</code> para especificar um valor nulo inteiro que é usado para habilitar ou desabilitar a compactação em uma API. AWS::ApplicationAutoScaling::ScalingPolicy No tipo de propriedade TargetTrackingScalingPolicyConfiguration, use a propriedade <code>DisableScaleIn</code> para especificar se a escala pela política de rastreamento de destino está desativada. AWS::EC2::SpotFleet No tipo de propriedade LaunchSpecifications, use a propriedade <code>TagSpecifications</code> para especificar as tags a serem aplicadas durante a criação da <code>SpotFleet</code>. AWS::Elasticsearch::Domain Use o atributo <code>Arn</code> para que <code>Fn::GetAtt</code> devolva o Nome de recurso da Amazon (ARN) do domínio. O atributo <code>DomainArn</code> do <code>Fn::GetAtt</code> está obsoleto.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::RDS::DBCluster Use a propriedade <code>DBClusterIdentifier</code> para especificar o identificador do cluster de banco de dados. AWS::RDS::DBCluster Use a propriedade <code>DBClusterIdentifier</code> para especificar o identificador do cluster de banco de dados. AWS::Redshift::Cluster Use a propriedade <code>ClusterIdentifier</code> para especificar o identificador exclusivo do cluster. AWS::Route53::HealthCheck No tipo de propriedade HealthCheckConfig , use a propriedade <code>Regions</code> para especificar as regiões a partir das quais você deseja que os verificadores de integridade do Route 53 analisem o endpoint especificado. AWS::SSM::Document Use a propriedade <code>Tags</code> para especificar as tags de recurso do CloudFormation a serem aplicadas ao documento.	

Alteração	Data de lançamento	Descrição	Versão da API
Recurso atualizado	19 de fevereiro de 2018	AWS::CodeBuild::Project Use a propriedade <code>Triggers</code> para configurar um webhook para o projeto e começar a reconstruir o código-fonte automaticamente sempre que uma alteração de código for enviada ao repositório. Ela está disponível apenas para projetos do GitHub no CloudFormation. Ele não está disponível para os projetos do GitHub Enterprise.	15/05/2010
Recurso atualizado	8 de fevereiro de 2018	AWS::DynamoDB::Table Use a propriedade <code>SSESpecification</code> para especificar as configurações a fim de habilitar a criptografia no lado do servidor.	15/05/2010
Recurso atualizado	5 de fevereiro de 2018	AWS::CodeBuild::Project No tipo de propriedade Source CodeBuild Project Source: • Use a propriedade <code>GitCloneDepth</code> para especificar a profundidade do histórico para download. • Use a propriedade <code>InsecureSsl</code> para especificar se os avisos do SSL serão ignorados quando se conectar ao repositório do projeto do GitHub Enterprise.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	23 de janeiro de 2018	AWS::AutoScaling::LifecycleHook Use a propriedade <code>LifecycleHookName</code> para especificar o nome do hook de ciclo de vida. AWS::DynamoDB::Table A propriedade <code>AttributeDefinitions</code> agora requer substituição quando atualizada. AWS::EC2::Instance Use a propriedade <code>CreditSpecification</code> para especificar a opção de crédito para uso de CPU de uma instância T2. Use a propriedade <code>ElasticGpuSpecifications</code> para especificar a GPUs elásticas, os recursos de GPU que você pode anexar a sua instância para acelerar o desempenho gráfico de seus aplicativos. AWS::EC2::VPC A propriedade <code>InstanceTenancy</code> agora não requer mais interrupção quando é atualizada de "dedicated" para "default" . AWS::ECS::Service Use a propriedade <code>HealthCheckGracePeriodSeconds</code> para especificar o período, em segundos, durante o qual o programador de serviços do Amazon ECS ignorará as verificações de integridade de destino do Elastic Load Balancing com problemas de integridade depois que uma tarefa tiver sido iniciada pela primeira vez.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::IoT::TopicRule No tipo de propriedade DynamoDBAction, as propriedades <code>RangeKeyField</code> e <code>RangeKeyValue</code> não são mais necessárias. AWS::KinesisAnalytics::ApplicationOutput No tipo de propriedade ApplicationOutput, use a propriedade <code>LambdaOutput</code> para identificar uma função Lambda como o destino ao configurar a saída do aplicativo. AWS::Kinesis::Stream Use a propriedade <code>StreamEncryption</code> para habilitar ou atualizar a criptografia no lado do servidor usando uma chave do AWS KMS key para um fluxo especificado. AWS::Lambda::Function Use a propriedade <code>ReservedConcurrentExecutions</code> para especificar o valor máximo de execuções simultâneas que você deseja reservar para a função. AWS::RDS::DBSubnetGroup Use a propriedade <code>DBSubnetGroupName</code> para especificar o nome do grupo de sub-redes de banco de dados. AWS::S3::Bucket Use a propriedade <code>BucketEncryption</code> para especificar a criptografia padrão para um bucket usando a criptografia no lado do servidor com chaves	

Alteração	Data de lançamento	Descrição	Versão da API
		SSE-S3 gerenciadas pelo Amazon S3 ou um bucket de AWS KMS keys (SSE-KMS). No tipo de propriedade ReplicationRule, use a propriedade <code>SourceSelectionCriteria</code> para especificar filtros adicionais para identificar os objetos de origem que você deseja replicar. No tipo de propriedade ReplicationDestination: • Use a propriedade <code>AccessControlTranslation</code> para especificar a propriedade da réplica da conta da AWS que possui o bucket de destino. • Use a propriedade <code>Account</code> para especificar o ID da conta do proprietário do bucket de destino. • Use a propriedade <code>EncryptionConfiguration</code> para especificar informações de criptografia para um bucket que é um destino para objetos replicados. AWS::SSM::Association Use a propriedade <code>AssociationName</code> para especificar o nome da associação entre um documento do SSM e as instâncias do EC2 que contêm um agente de configuração para processar o documento.	
Acionadores de reversão adicionados ao console do CloudFormation.	15 de janeiro de 2018	Os acionadores de reversão permitem utilizar o CloudFormation para monitorar o estado da sua aplicação durante a criação e atualização de pilhas, e que você reverta essa operação se o aplicativo violar os limites de qualquer um dos alarmes especificados. Para obter mais informações, consulte Monitorar e reverter operações de pilha .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recurso atualizado	12 de janeiro de 2018	AWS::SSM::Parameter Use a propriedade <code>AllowedPattern</code> para especificar uma expressão regular usada para validar o valor do parâmetro.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	5 de dezembro de 2017	AWS::Inspector::AssessmentTarget Use o recurso <code>AWS::Inspector::AssessmentTarget</code> para criar um destino de avaliação para o Amazon Inspector. AWS::Inspector::AssessmentTemplate Use o recurso <code>AWS::Inspector::AssessmentTemplate</code> para criar um modelo de avaliação para o Amazon Inspector. AWS::Inspector::ResourceGroup Use o recurso <code>AWS::Inspector::ResourceGroup</code> para criar um grupo de recursos do Amazon Inspector, que define as tags que identificam os recursos da AWS que fazem parte de um destino de avaliação do Amazon Inspector. AWS::ServiceDiscovery::Instance Use o recurso <code>AWS::ServiceDiscovery::Instance</code> para especificar as informações sobre uma instância criada pelo Amazon Route 53. AWS::ServiceDiscovery::PrivateDnsNamespace Use o recurso <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> para especificar as informações sobre um namespace privado para o Amazon Route 53. AWS::ServiceDiscovery::PublicDnsNamespace Use o recurso <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> para especificar informação	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		es sobre um namespace público para o Amazon Route 53. AWS::ServiceDiscovery::Service Use o recurso <code>AWS::ServiceDiscovery::Service</code> para definir um modelo para até cinco registros e uma verificação de integridade opcional que você deseja que o Amazon Route 53 crie quando você registrar uma instância.	
Recurso atualizado	5 de dezembro de 2017	AWS::KinesisAnalytics::Application No tipo de propriedade Input, use a propriedade <code>InputProcessingConfiguration</code> para transformar registros à medida que são recebidos do fluxo.	15/05/2010
Recurso atualizado	1.º de dezembro de 2017	AWS::CodeBuild::Project Use a propriedade <code>BadgeEnabled</code> para gerar um URL acessível publicamente para um distintivo de compilação do projeto. Use a propriedade <code>Cache</code> para definir configurações de cache para dependências de compilação. Use a propriedade <code>VpcConfig</code> para possibilitar que o CodeBuild acesse recursos em uma Amazon VPC. No tipo de propriedade EnvironmentVariable, use a propriedade <code>Type</code> para especificar o tipo de variável de ambiente.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	30 de novembro de 2017	AWS::Cloud9::EnvironmentEC2 Use o recurso <code>AWS::Cloud9::EnvironmentEC2</code> para criar um ambiente de desenvolvimento do Amazon EC2 no AWS Cloud9.	15/05/2010
Recursos atualizados	29 de novembro de 2017	AWS::ECS::TaskDefinition Use a propriedade <code>Cpu</code> para especificar o número de unidades de cpu necessárias para a tarefa. Use a propriedade <code>ExecutionRoleArn</code> para especificar o ARN da função de execução. Use a propriedade <code>Memory</code> para especificar a quantidade (em MiB) de memória necessária para a tarefa. Use a propriedade <code>RequiresCompatibilities</code> para especificar o tipo de inicialização necessário para a tarefa. AWS::ECS::Service Use a propriedade <code>LaunchType</code> para especificar o tipo de inicialização no qual executar o serviço. Use a propriedade <code>NetworkConfiguration</code> para especificar a configuração de rede para o serviço. Use a propriedade <code>PlatformVersion</code> para especificar a versão da plataforma na qual executar o serviço.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	28 de novembro de 2017	AWS::GuardDuty::Detector Use o recurso <code>AWS::GuardDuty::Detector</code> para criar um único detector do Amazon GuardDuty. AWS::GuardDuty::IPSet Use o recurso <code>AWS::GuardDuty::IPSet</code> para criar um conjunto do Amazon GuardDutyIP. AWS::GuardDuty::ThreatIntelSet Use o recurso <code>AWS::GuardDuty::ThreatIntelSet</code> para criar um ThreatIntelSet.	15/05/2010
Recursos atualizados	28 de novembro de 2017	AWS::CodeDeploy::Application Use a propriedade <code>ComputePlatform</code> para especificar uma plataforma de computação do AWS Lambda para o CodeDeploy implantar uma aplicação. AWS::CodeDeploy::DeploymentGroup No tipo de propriedade DeploymentStyle, use a propriedade <code>DeploymentType</code> para especificar uma implantação azul/verde em uma plataforma de computação do Lambda. AWS::EC2::SpotFleet No tipo de propriedade SpotFleetRequestConfigData, a propriedade <code>SpotPrice</code> agora é opcional. AWS::Lambda::Alias Use a propriedade <code>RoutingConfig</code> para especificar duas versões diferentes de uma função do AWS Lambda permitindo determinar qual porcentagem de tráfego invocará cada versão.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Nova política de atualização de CodeDeployLambdaAliasUpdate	28 de novembro de 2017	Use a política de atualização CodeDeployLambdaAliasUpdate para executar uma implantação do CodeDeploy quando a versão for alterada em um recurso <code>AWS::Lambda::Alias</code> . Para obter mais informações, consulte Atributo UpdatePolicy .	15/05/2010
Novos tipos de parâmetro do SSM	21 de novembro de 2017	Use os tipos de parâmetro SSM para usar parâmetros existentes no Systems Manager Parameter Store. Observação: no momento, o CloudFormation não oferece suporte ao tipo <code>SecureString</code> . Para obter mais informações, consulte Tipos de parâmetros do SSM .	15/05/2010
Novo campo ResolvedValue para tipo de dados Parameter	21 de novembro de 2017	O campo <code>ResolvedValue</code> retorna o valor usado na definição da pilha para um parâmetro SSM. Para obter mais informações, consulte o tipo de dados Parameter na Referência de API do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	20 de novembro de 2017	AWS::ApiGateway::ApiKey Use a propriedade <code>CustomerId</code> para especificar um identificador de cliente do AWS Marketplace. Use a propriedade <code>GenerateDistinctId</code> para especificar se o identificador de chave é distinto do valor da chave da API criada. AWS::ApiGateway::Authorizer Use a propriedade <code>AuthType</code> para especificar um campo definido pelo cliente que é usado em importações e exportações do Swagger sem impacto funcional. AWS::ApiGateway::DomainName Use a propriedade <code>EndpointConfiguration</code> para especificar os tipos de endpoint de um nome de domínio do API Gateway. Use a propriedade <code>RegionalCertificateArn</code> para fazer referência a um certificado para uso pelo endpoint regional para um nome de domínio. AWS::ApiGateway::Method Nos tipos de propriedade Integration e IntegrationResponse, use a propriedade <code>ContentHandling</code> para especificar como lidar com conversões de tipo de conteúdo de carga útil de solicitação. AWS::ApiGateway::RestApi Use a propriedade <code>EndpointConfiguration</code> para especificar os tipos de endpoint de uma API REST do API Gateway.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::ApplicationAutoScaling::ScalableTarget Use a propriedade <code>ScheduledActions</code> para especificar ações programadas para um destino escalável do Aplicativo Auto Scaling. AWS::ECR::Repository Use a propriedade <code>LifecyclePolicy</code> para especificar uma política de ciclo de vida para um repositório do Amazon ECR. AWS::ECS::TaskDefinition No tipo de propriedade ContainerDefinition, use a propriedade <code>LinuxParameters</code> para especificar opções específicas ao Linux para um contêiner do Amazon ECS. AWS::ElastiCache::ReplicationGroup Use a propriedade <code>AtRestEncryptionEnabled</code> para habilitar a criptografia em repouso. Use a propriedade <code>AuthToken</code> para especificar uma senha que é usada para acessar um servidor protegido por senha. Use a propriedade <code>TransitEncryptionEnabled</code> para habilitar a criptografia em trânsito. AWS::ElasticLoadBalancingV2::TargetGroup Use o atributo <code>TargetGroupName</code> com a função <code>Fn::GetAtt</code> para obter o nome de um grupo de destino do Elastic Load Balancing.	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::Elasticsearch::Domain Use a propriedade <code>VPCOptions</code> para especificar a configuração de uma VPC para o domínio do OpenSearch Service. AWS::EMR::Cluster Use a propriedade <code>EbsRootVolumeSize</code> para especificar o tamanho do volume raiz do EBS para um cluster do Amazon EMR. AWS::RDS::DBInstance Use as propriedades <code>SourceRegion</code> e <code>KmsKeyId</code> para criar uma réplica de leitura criptografada a partir de uma instância de banco de dados de origem entre regiões. AWS::Route53::HostedZone Use a propriedade <code>QueryLoggingConfig</code> para especificar a configuração do log de consultas do DNS.	
Novo campo <code>NoEcho</code> para objetos de recurso personalizado <code>Response</code>	20 de novembro de 2017	Agora, você pode usar o campo <code>NoEcho</code> opcional para mascarar a saída de um recurso personalizado. Para obter mais informações, consulte Objetos de resposta de recurso personalizado. O parâmetro <code>noEcho</code> correspondente é compatível com o método <code>send</code>. Para obter mais informações, consulte Módulo <code>cfn-response</code>.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
A instância de pilha substitui os conjuntos de pilhas adicionados.	17 de novembro de 2017	O CloudFormation StackSets permite substituir valores de parâmetros em instâncias de pilha por conta e região. Você pode substituir valores de parâmetros ao criar instâncias de pilha ou ao atualizar instâncias de pilha existentes. Para obter mais informações, consulte Parâmetros de substituição nas instâncias de pilha .	15/05/2010
Recurso atualizado	15 de novembro de 2017	AWS::StepFunctions::StateMachine Você pode usar <code>AWS::StepFunctions::StateMachine</code> para especificar um <code>StateMachineName</code> ao criar uma máquina de estado, e <code>DefinitionString</code> e <code>RoleArn</code> podem ser atualizados sem substituir a máquina de estado.	15/05/2010
O StackSets agora é compatível com, no máximo, 500 instâncias de pilha por conjunto de pilhas.	6 de novembro de 2017	Agora você pode criar até 500 instâncias de pilha por conjunto de pilhas. Para obter mais informações sobre limites do CloudFormation, consulte Entender as cotas do CloudFormation .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	2 de novembro de 2017	AWS::CloudFront::CloudFrontOriginAccessIdentity Use o recurso <code>AWS::CloudFront::CloudFrontOriginAccessIdentity</code> para especificar a identidade de acesso de origem do Amazon CloudFront a ser associada à origem de uma distribuição do CloudFront. AWS::CloudFront::StreamingDistribution Use o recurso <code>AWS::CloudFront::StreamingDistribution</code> para especificar uma distribuição em streaming Adobe Real-Time Messaging Protocol (RTMP) para o CloudFront.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	2 de novembro de 2017	AWS::ApiGateway::Deployment A propriedade <code>StageName</code> se tornou obsoleta no tipo de propriedade StageDescription. AWS::ApiGateway::Method Use a propriedade <code>OperationName</code> para atribuir um nome amigável a um método do API Gateway. Use a propriedade <code>RequestValidatorId</code> para associar um validador de solicitação a um método. AWS::AutoScaling::AutoScalingGroup Use a propriedade <code>LifecycleHookSpecificationList</code> para especificar as ações a serem executadas quando o Auto Scaling inicia ou encerra instâncias. AWS::CloudFront::Distribution Use a propriedade <code>Tags</code> para especificar um conjunto arbitrário de tags (pares de chave-valor) a ser associado a uma distribuição do CloudFront. Nos tipos de propriedade CacheBehavior e DefaultCacheBehavior, use a propriedade <code>LambdaFunctionAssociations</code> para especificar associações de funções do Lambda para uma distribuição do CloudFront. No tipo de propriedade CustomOriginConfig, use a propriedade <code>OriginKeepaliveTimeout</code> para especificar um tempo limite de keep-alive personalizado e a propriedade <code>OriginReadTimeout</code> para especificar um tempo limite de origem personalizado.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		No tipo de propriedade DistributionConfig, use a propriedade <code>IPv6Enabled</code> para especificar se o CloudFront responde a solicitações DNS IPv6 com um endereço IPv6 para a sua distribuição. AWS::CodeDeploy::DeploymentGroup No tipo de propriedade LoadBalancerInfo, use a propriedade <code>TargetGroupInfoList</code> para especificar informações sobre um grupo de destino no Elastic Load Balancing para uso em uma implantação. AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupEgress, e AWS::EC2::SecurityGroupIngress Use a propriedade <code>Description</code> para especificar a descrição da regra de um grupo de segurança. AWS::EC2::Subnet Agora a propriedade <code>Ipv6CidrBlock</code> é compatível com as atualizações <code>No interruption</code>. AWS::EC2::VPNGateway Use a propriedade <code>AmazonSideAsn</code> para especificar um Número de sistema autônomo (ASN) privado da Amazon de uma sessão BGP. AWS::EC2::VPNConnection Use a propriedade <code>VpnTunnelOptionsSpecifications</code> para configurar as opções de túnel de uma conexão VPN.	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::ElasticBeanstalk::ConfigurationTemplate e AWS::ElasticBeanstalk::Environment Nos tipos de propriedade ConfigurationOptionSetting e OptionSetting, use a propriedade ResourceName para especificar um nome de recurso para uma opção de configuração de escalabilidade baseada no tempo. AWS::EMR::Cluster Use a propriedade CustomAmiId para especificar uma AMI personalizada do Amazon Linux para um cluster. AWS::KinesisFirehose::DeliveryStream Use o atributo Arn com a função Fn::GetAtt para obter o Nome de recurso da Amazon (ARN) do fluxo de entrega. AWS::KMS::Key Use a propriedade Tags para especificar um conjunto arbitrário de tags (pares de chave/valor) a ser associado a uma chave gerenciada pelo cliente. AWS::OpsWorks::Layer e AWS::OpsWorks::Stack Use a propriedade Tags para especificar um conjunto arbitrário de tags (pares de chave/valor) a ser associado a uma camada ou pilha do OpsWorks. AWS::RDS::OptionGroup No tipo de propriedade OptionConfiguration, use a propriedade OptionVersion para especificar uma versão para a opção.	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::S3::Bucket Use a propriedade <code>AnalyticsConfigurations</code> para configurar um filtro de análise para um bucket do Amazon S3.	

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	24 de outubro de 2017	AWS::Glue::Classifier Use o recurso <code>AWS::Glue::Classifier</code> para criar um classificador do AWS Glue. AWS::Glue::Connection Use o recurso <code>AWS::Glue::Connection</code> para especificar uma conexão do AWS Glue a uma fonte de dados. AWS::Glue::Crawler Use o recurso <code>AWS::Glue::Crawler</code> para especificar um crawler do AWS Glue. AWS::Glue::Database Use o recurso <code>AWS::Glue::Database</code> para criar um banco de dados do AWS Glue. AWS::Glue::DevEndpoint Use o recurso <code>AWS::Glue::DevEndpoint</code> para especificar um endpoint de desenvolvimento para depurar remotamente scripts da ETL. AWS::Glue::Job Use o recurso <code>AWS::Glue::Job</code> para especificar um trabalho do AWS Glue no catálogo de dados. AWS::Glue::Partition Use o recurso <code>AWS::Glue::Partition</code> para criar uma partição do AWS Glue, que representa uma parte dos dados da tabela.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::Glue::Table Use o recurso <code>AWS::Glue::Table</code> para criar uma tabela do AWS Glue. AWS::Glue::Trigger Use o recurso <code>AWS::Glue::Trigger</code> para especificar os triggers que executam trabalhos do AWS Glue.	
Novos recursos	11 de outubro de 2017	AWS::SSM::MaintenanceWindow Use o recurso <code>AWS::SSM::MaintenanceWindow</code> para criar uma janela de manutenção do AWS Systems Manager. AWS::SSM::MaintenanceWindowTarget Use o recurso <code>AWS::SSM::MaintenanceWindowTarget</code> para registrar um destino com uma janela de manutenção. AWS::SSM::MaintenanceWindowTask Use o recurso <code>AWS::SSM::MaintenanceWindowTask</code> para definir uma tarefa de janela de manutenção. AWS::SSM::PatchBaseline Use o recurso <code>AWS::SSM::PatchBaseline</code> para definir uma linha de base de patch do Systems Manager.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	10 de outubro de 2017	AWS::ElasticLoadBalancingV2::ListenerCertificate Use o recurso <code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code> para especificar certificados para um listener do Elastic Load Balancing.	15/05/2010
Novo recurso	27 de setembro de 2017	AWS::Athena::NamedQuery Use o recurso <code>AWS::Athena::NamedQuery</code> para criar uma consulta do Amazon Athena.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	27 de setembro de 2017	AWS::EC2::NatGateway Use a propriedade Tags para especificar as tags de recurso para um gateway do NAT. AWS::ElasticBeanstalk::Application Use a propriedade ResourceLifecycleConfig para definir configurações de ciclo de vida para os recursos que pertencem ao aplicativo e a função de serviço que o Elastic Beanstalk assume para aplicar configurações de ciclo de vida. AWS::ElasticBeanstalk::ConfigurationTemplate e AWS::ElasticBeanstalk::Environment Use a propriedade PlatformArn para especificar uma plataforma personalizada para o Elastic Beanstalk. AWS::ElasticLoadBalancingV2::TargetGroup No tipo de propriedade TargetDescription, use a propriedade AvailabilityZone para especificar a zona de disponibilidade em que o endereço IP deve ser registrado. AWS::Events::Rule No tipo de propriedade Target, use as seguintes propriedades de transformação de entrada de eventos e definição dos destinos de tarefa do Amazon ECS e fluxo do Kinesis. - EcsParameters - InputTransformer - KinesisParameters	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		RunCommandParameters AWS::KinesisFirehose::DeliveryStream Use a propriedade <code>DeliveryStreamType</code> para especificar o tipo de fluxo e a propriedade <code>KinesisStreamSourceConfiguration</code> para especificar os ARNs de fluxo e função para um fluxo do Kinesis usado como origem para um fluxo de entrega. AWS::RDS::DBInstance Para a propriedade <code>Engine</code>, se você tiver especificado <code>oracle-se</code> ou <code>oracle-se1</code>, poderá atualizar para <code>oracle-se2</code> sem substituir a instância de banco de dados. AWS::S3::Bucket Use a propriedade <code>AccelerateConfiguration</code> para configurar o estado de aceleração de transferência para um bucket do Amazon S3.	
Proteção contra encerramento adicionada para pilhas.	26 de setembro de 2017	Ativar a proteção contra encerramento em uma pilha impede que ela seja excluída acidentalmente. Um usuário não pode excluir uma pilha que tem a proteção contra encerramento habilitada. Para obter mais informações, consulte Proteção de uma pilha contra exclusão .	15/05/2010
Valor <code>umask</code> padrão alterado a partir da versão 1.4-22	14 de setembro de 2017	O valor padrão do parâmetro <code>umask</code> do arquivo de configuração <code>cfn-hup.conf</code> agora é <code>022</code> . Para obter mais informações, consulte cfn-hup .	

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	7 de setembro de 2017	AWS::ElasticLoadBalancingV2::LoadBalancer Use a propriedade <code>SubnetMappings</code> para especificar os IDs das sub-redes para anexar ao load balancer. Use a propriedade <code>Type</code> para especificar o tipo de load balancer para criar. AWS::ElasticLoadBalancingV2::TargetGroup Use a propriedade <code>TargetType</code> para especificar o tipo de registro dos destinos neste grupo de destinos.	15/05/2010
Acionadores de reversão adicionados à API do CloudFormation	31 de agosto de 2017	Os acionadores de reversão permitem utilizar o CloudFormation para monitorar o estado da sua aplicação durante a criação e atualização de pilhas, e que você reverta essa operação se o aplicativo violar os limites de qualquer um dos alarmes especificados. Para obter mais informações, consulte RollbackConfiguration na Referência de APIs do CloudFormation.	15/05/2010
Novo parâmetro <code>umask</code> para o arquivo <code>cfn-hup.conf</code>	31 de agosto de 2017	Use o parâmetro <code>umask</code> no arquivo de configuração <code>cfn-hup.conf</code> para controlar as permissões de arquivos usadas pelo daemon <code>cfn-hup</code> (versão 1.4-21). Para obter mais informações, consulte cfn-hup .	

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados para oferecer suporte ao dimensionamento da VPC	29 de agosto de 2017	AWS::EC2::VPCCidrBlock Use a propriedade <code>CidrBlock</code> para associar um bloco CIDR do IPv4 a uma VPC. AWS::EC2::VPC Use o atributo <code>CidrBlockAssociations</code> com a função <code>Fn::GetAtt</code> para obter uma lista dos IDs de associação de blocos CIDR do IPv4 associados à VPC.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	23 de agosto de 2017	AWS::S3::Bucket No tipo de propriedade Rule, use a propriedade <code>TagFilters</code> para especificar as tags a serem usadas na identificação de um subconjunto de objetos para um bucket do Amazon S3. Use a propriedade <code>MetricsConfiguration</code> para especificar uma configuração de métricas para as métricas de solicitação do CloudWatch de um bucket do Amazon S3. AWS::IoT::TopicRule No tipo de propriedade Action, use a propriedade <code>DynamoDBv2Action</code> para descrever uma ação do AWS IoT que grava dados em uma tabela do DynamoDB. No tipo de propriedade Action, a propriedade <code>DynamoDBAction</code> agora oferece suporte às propriedades <code>HashKeyType</code> e <code>RangeKeyType</code>. AWS::Lambda::Permission Use a propriedade <code>EventSourceToken</code> para especificar um único token que deve ser fornecido pela função principal de chamada.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos pseudoparâmetros	23 de agosto de 2017	Use o pseudoparâmetro <code>AWS::Partition</code> para retornar a partição onde se encontra um recurso. Use o pseudoparâmetro <code>AWS::URLSuffix</code> para retornar o sufixo para um domínio. Para obter mais informações, consulte Referência de pseudoparâmetros.	15/05/2010
Novos recursos para oferecer suporte ao DAX	22 de agosto de 2017	AWS::DAX::Cluster Use o recurso <code>AWS::DAX::Cluster</code> para criar um cluster do DAX para usar com o Amazon DynamoDB. AWS::DAX::ParameterGroup Use o recurso <code>AWS::DAX::ParameterGroup</code> para criar um grupo de parâmetros para usar com o Amazon DynamoDB. AWS::DAX::SubnetGroup Use o recurso <code>AWS::DAX::SubnetGroup</code> para criar um grupo de sub-redes para usar com o DAX (DynamoDB Accelerator).	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	18 de agosto de 2017	AWS::ApiGateway::DocumentationPart e AWS::ApiGateway::DocumentationVersion para criar documentação para sua API do API Gateway. AWS::ApiGateway::GatewayResponse Use o recurso <code>AWS::ApiGateway::GatewayResponse</code> para criar uma resposta personalizada para sua API do API Gateway. AWS::ApiGateway::RequestValidator Use o recurso <code>AWS::ApiGateway::RequestValidator</code> para configurar regras de validação para as solicitações de entrada para sua API do API Gateway. AWS::EC2::NetworkInterfacePermission Use o recurso <code>AWS::EC2::NetworkInterfacePermission</code> para conceder permissão a uma conta da AWS para acessar uma interface de rede.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	18 de agosto de 2017	AWS::ApiGateway::Stage Use a propriedade <code>DocumentationVersion</code> para especificar um snapshot da documentação da API com controle de versão. AWS::AutoScaling::ScalingPolicy Use a propriedade <code>TargetTrackingConfiguration</code> para especificar uma configuração de política de escalabilidade de rastreamento de destino do Auto Scaling. AWS::CloudTrail::Trail Use a propriedade <code>EventSelectors</code> para oferecer suporte a eventos de dados do Amazon S3. AWS::CodeDeploy::DeploymentGroup Use as propriedades <code>LoadBalancerInfo</code> e <code>DeploymentStyle</code> para especificar um load balancer do Elastic Load Balancing para uma implantação no local. Use a propriedade <code>AutoRollbackConfiguration</code> para configurar a reversão automática da implantação. AWS::EC2::SpotFleet No tipo de propriedade SpotFleetRequestConfigData, use a propriedade <code>ReplaceUnhealthyInstances</code> para indicar se a frota spot deve substituir as instâncias com problemas de integridade e a propriedade <code>Type</code> para especificar o tipo de solicitação.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::EC2::Subnet Use as propriedades <code>AssignIpv6AddressOnCreation</code> e <code>Ipv6CidrBlock</code> para criar uma sub-rede com um bloco CIDR do IPv6. AWS::KinesisFirehose::DeliveryStream Use a propriedade <code>ExtendedS3DestinationConfiguration</code> para configurar um destino no Amazon S3. Use a subpropriedade <code>ProcessingConfiguration</code>, dentro de cada configuração de destino, para chamar as funções Lambda que transformam dados de origem de entrada e entregar os dados transformados nos destinos. AWS::RDS::DBCluster e AWS::RDS::DBInstance A <code>DeletionPolicy</code> padrão agora é <code>Snapshot</code> para os recursos <code>AWS::RDS::DBCluster</code> e para os recursos <code>AWS::RDS::DBInstance</code> que não especificam a propriedade <code>DBClusterIdentifier</code>. Para obter mais informações, consulte Atributo DeletionPolicy. AWS::S3::Bucket No tipo de propriedade Rule, use a propriedade <code>AbortIncompleteMultipartUpload</code> para especificar uma regra de ciclo de vida que cancela multipart uploads incompletos para um bucket do Amazon S3.	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::SQS::Queue Use as propriedades <code>KmsMasterKeyId</code> e <code>KmsDataKeyReusePeriodSeconds</code> para configurar a criptografia no lado do servidor para o Amazon SQS. Foi adicionado o atributo <code>Arn</code> à função intrínseca <code>Fn::GetAtt</code> para os seguintes recursos: • AWS::CloudTrail::Trail. também foi adicionado <code>SnsTopicArn</code>. • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	
Suporte para tags de pilhas em artefatos do CodePipeline	18 de agosto de 2017	Agora você pode especificar tags para pilhas em arquivos de configuração de modelo para usar como artefatos para os pipelines do CodePipeline. As tags especificadas são aplicadas às pilhas criadas usando o arquivo de configuração de modelo. Para obter mais informações, consulte Artefatos do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Criar sistemas de arquivos criptografados	14 de agosto de 2017	AWS::EFS::FileSystem Use a propriedade <code>Encrypted</code> para criptografar um sistema de arquivos do Amazon EFS durante sua criação. Use a propriedade <code>KmsKeyId</code> para especificar, opcionalmente, uma chave personalizada gerenciada pelo cliente a ser usada para proteger o sistema de arquivos criptografado.	15/05/2010
Novos recursos para oferecer suporte ao AWS Batch	8 de agosto de 2017	AWS::Batch::ComputeEnvironment Use o recurso <code>AWS::Batch::ComputeEnvironment</code> para definir o ambiente de computação AWS Batch. AWS::Batch::JobDefinition Use o recurso <code>AWS::Batch::JobDefinition</code> para especificar os parâmetros para a definição de um trabalho AWS Batch. AWS::Batch::JobQueue Use o recurso <code>AWS::Batch::JobQueue</code> para definir a fila de trabalhos AWS Batch.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos para suporte ao Amazon Managed Service for Apache Flink	28 de julho de 2017	AWS::KinesisAnalytics::Application Use o recurso <code>AWS::KinesisAnalytics::Application</code> para criar uma aplicação do Amazon Managed Service for Apache Flink. AWS::KinesisAnalytics::ApplicationOutput Use o recurso <code>AWS::KinesisAnalytics::ApplicationOutput</code> para adicionar um destino externo para sua aplicação do Amazon Managed Service for Apache Flink. AWS::KinesisAnalytics::ApplicationReferenceDataSource Use o recurso <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code> para adicionar uma fonte de dados de referência a uma aplicação do Amazon Managed Service for Apache Flink existente.	15/05/2010
Use StackSets para gerenciar centralmente as pilhas em todas as contas e regiões	25 de julho de 2017	O StackSets permite criar, atualizar ou excluir pilhas em várias contas e regiões em uma única operação. Usando uma conta de administrador, você define e gerencia um modelo do CloudFormation e o usa como base para o provisionamento de pilhas para as contas de destino selecionadas nas várias regiões especificadas. Para obter mais informações, consulte Gerenciar pilhas em todas as contas e regiões com o StackSets .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Visualizar eventos de pilha por token de solicitação do cliente	14 de julho de 2017	No console, as operações de pilha exibem o token de solicitação do cliente na guia Eventos. Todos os eventos acionados por uma determinada operação de pilha são atribuídos ao mesmo token de solicitação do cliente, o qual você pode usar para rastrear as operações. Para obter mais informações, consulte Visualizar dados e recursos de pilha do CloudFormation no Console de gerenciamento da AWS e StackEvent na Referência da API do AWS CloudFormation.	15/05/2010
Usar links de quick-create para pilhas	14 de julho de 2017	Use links de quick-create para preparar rapidamente as pilhas para execução. Você pode especificar o modelo de URL, nome da pilha e parâmetros do modelo para preencher previamente uma única página do Assistant e de criação de pilhas. Para obter mais informações, consulte Criar links de quick-create para pilhas .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos para oferecer suporte ao AWS Database Migration Service	12 de julho de 2017	AWS::DMS::Certificate Use o recurso <code>AWS::DMS::Certificate</code> para criar um certificado SSL que criptografa conexões entre endpoints do AWS DMS e a instância de replicação. AWS::DMS::Endpoint Use o recurso <code>AWS::DMS::Endpoint</code> para criar um endpoint do AWS DMS. AWS::DMS::EventSubscription Use o recurso <code>AWS::DMS::EventSubscription</code> para receber notificações de eventos do AWS DMS por meio do Amazon Simple Notification Service. AWS::DMS::ReplicationInstance Use o recurso <code>AWS::DMS::ReplicationInstance</code> para criar uma instância de replicação do AWS DMS. AWS::DMS::ReplicationSubnetGroup Use o recurso <code>AWS::DMS::ReplicationSubnetGroup</code> para criar um grupo de sub-redes de replicação do AWS DMS. AWS::DMS::ReplicationTask Use o recurso <code>AWS::DMS::ReplicationTask</code> para criar uma tarefa de replicação do AWS DMS.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	5 de julho de 2017	AWS::CloudWatch::Dashboard Use o recurso <code>AWS::CloudWatch::Dashboard</code> para especificar um painel do CloudWatch personalizado para o console do CloudWatch. AWS::ApiGateway::DomainName Use o recurso <code>AWS::ApiGateway::DomainName</code> para especificar um URL amigável personalizado para sua API que é implantado no Amazon API Gateway. AWS::EC2::EgressOnlyInternetGateway Use o recurso <code>AWS::EC2::EgressOnlyInternetGateway</code> para criar um gateway da Internet somente de saída para sua VPC. InstanceFleetConfig Use o recurso <code>InstanceFleetConfig</code> para configurar uma frota de instâncias spot para um cluster do Amazon EMR.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	5 de julho de 2017	AWS::ApiGateway::RestApi Use a propriedade <code>BinaryMediaTypes</code> para especificar os tipos de mídia binários que são compatíveis. AWS::ApplicationAutoScaling::ScalingPolicy Use a propriedade <code>TargetTrackingScalingPolicyConfiguration</code> para especificar uma configuração de política de escalabilidade de rastreamento de destino. AWS::CloudTrail::Trail Use a propriedade <code>TrailName</code> para especificar um nome personalizado para um recurso do AWS CloudTrail. Use a propriedade <code>Tags</code> para especificar tags de recurso. AWS::CodeDeploy::DeploymentGroup Use a propriedade <code>AlarmConfiguration</code> para configurar alarmes para o grupo de implantação. Use a propriedade <code>TriggerConfigurations</code> para configurar triggers de notificação para o grupo de implantação. AWS::EMR::Cluster Use as propriedades <code>CoreInstanceFleet</code> e <code>MasterInstanceFleet</code> no tipo de propriedade JobFlowInstancesConfig para configurar a frota de instâncias spot para um cluster do Amazon EMR.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::DynamoDB::Table Use a propriedade <code>TimeToLiveSpecification</code> para especificar as configurações de tempo de vida (TTL) para uma tabela do Amazon DynamoDB. Use a propriedade <code>Tags</code> para especificar as tags de recurso para uma tabela do DynamoDB. AWS::EC2::Instance Agora a propriedade <code>IamInstanceProfile</code> é compatível com as atualizações <code>No interruption</code>. AWS::EC2::Route Use a propriedade <code>EgressOnlyInternetGatewayId</code> para especificar um gateway da Internet somente de saída para uma rota do EC2. AWS::Kinesis::Stream Use a propriedade <code>RetentionPeriodHours</code> para especificar o número de horas em que os registros de dados armazenados em estilhaços permanecerão acessíveis. AWS::RDS::DBCluster Use a propriedade <code>ReplicationSourceIdentifier</code> para criar um cluster de banco de dados como uma réplica de leitura de outro cluster de banco de dados ou de uma instância de banco de dados MySQL do Amazon RDS.	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::Redshift::Cluster Use a propriedade <code>LoggingProperties</code> para criar arquivos de log de auditoria e armazená-los no Amazon S3.	
Novos recursos	6 de junho de 2017	AWS::EMR::SecurityConfiguration Use o recurso <code>AWS::EMR::SecurityConfiguration</code> para criar uma configuração de segurança, que é armazenada no serviço e pode ser especificada quando um cluster é criado.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	6 de junho de 2017	AWS::AutoScaling::LifecycleHook As propriedades <code>NotificationTargetARN</code> e <code>RoleARN</code> são opcionais. AWS::CloudWatch::Alarm Agora você pode usar as propriedades <code>EvaluateLowSampleCountPercentile</code>, <code>ExtendedStatistic</code> e <code>TreatMissingData</code> ao criar os recursos <code>AWS::CloudWatch::Alarm</code>. AWS::EC2::SpotFleet O CloudFormation é compatível com alterações mutáveis nas propriedades de frotas Spot. As seguintes propriedades da propriedade <code>SpotFleetRequestConfigData</code> são compatíveis com as atualizações <code>Replacement</code>: • <code>AllocationStrategy</code> • <code>IamFleetRole</code> • <code>LaunchSpecifications</code> • <code>SpotPrice</code> • <code>TerminateInstancesWithExpiration</code> • <code>ValidFrom</code> • <code>ValidUntil</code> As seguintes propriedades da propriedade <code>SpotFleetRequestConfigData</code> são compatíveis com as atualizações <code>No interruption</code>: • <code>ExcessCapacityTerminationPolicy</code> • <code>TargetCapacity</code>	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::EMR::InstanceGroupConfig O CloudFormation agora oferece suporte ao ajuste de escala automático nos grupos de instâncias de tarefa do Amazon EMR. AWS::Events::Rule A propriedade RoleArn está obsoleta no recurso Rule. Use a propriedade RoleArn no tipo de propriedade Target para especificar a função do IAM a ser usada em um destino. AWS::Kinesis::Stream Agora a propriedade ShardCount é compatível com as atualizações No interruption . AWS::Lambda::Function Use a propriedade TracingConfig para definir configurações de rastreamento para as funções do Lambda. AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, e AWS::Redshift::ClusterSubnetGroup Use a propriedade Tags para especificar tags de recurso. AWS::RDS::DBCluster Adição do atributo ReadEndpoint.Address à função intrínseca Fn::GetAtt .	

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::S3::Bucket Adição do atributo <code>Arn</code> à função intrínseca <code>Fn::GetAtt</code> .	

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	11 de maio de 2017	Os novos recursos a seguir oferecem suporte ao uso do AWS WAF com os Application Load Balancers do Elastic Load Balancing (ELB). AWS::WAFRegional::ByteMatchSet Use o recurso <code>AWS::WAFRegional::ByteMatchSet</code> para identificar uma parte de uma solicitação da web que você deseja inspecionar. AWS::WAFRegional::IPSet Use o recurso <code>AWS::WAFRegional::IPSet</code> para especificar quais solicitações da web serão permitidas ou bloqueadas com base nos endereços IP em que as solicitações são originadas. AWS::WAFRegional::Rule Use o recurso <code>AWS::WAFRegional::Rule</code> para especificar uma combinação de objetos <code>IPSet</code>, <code>ByteMatchSet</code> e <code>SqlInjectionMatchSet</code> que identificam as solicitações da web que serão permitidas, bloqueadas ou contabilizadas. AWS::WAFRegional::SizeConstraintSet Use o recurso <code>AWS::WAFRegional::SizeConstraintSet</code> para especificar uma restrição de tamanho usada para verificar o tamanho de uma solicitação da web e quais partes da solicitação serão verificadas. AWS::WAFRegional::SqlInjectionMatchSet Use o recurso <code>AWS::WAFRegional::SqlInjectionMatchSet</code> para permitir, bloquear ou contabilizar solicitações que contêm código SQL mal-inten-	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		cionado em uma parte específica das solicitações da web. AWS::WAFRegional::WebACL Use o recurso <code>AWS::WAFRegional::WebACL</code> para identificar as solicitações da web que você deseja permitir, bloquear ou contabilizar. AWS::WAFRegional::WebACLAssociation Use o recurso <code>AWS::WAFRegional::WebACLAssociation</code> para associar uma lista de controle de acesso da web a um recurso. AWS::WAFRegional::XssMatchSet Use o recurso <code>AWS::WAFRegional::XssMatchSet</code> para especificar as partes das solicitações da web que o AWS WAF deve inspecionar em busca de ataques de scripting entre sites e o nome do cabeçalho a ser inspecionado.	

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	28 de abril de 2017	AWS::Cognito::IdentityPool Use o recurso <code>AWS::Cognito::IdentityPool</code> para criar um grupo de identidades do Amazon Cognito. AWS::Cognito::IdentityPoolRoleAttachment Use o recurso <code>AWS::Cognito::IdentityPoolRoleAttachment</code> para gerenciar a configuração de função de um grupo de identidades do Amazon Cognito. AWS::Cognito::UserPool Use o recurso <code>AWS::Cognito::UserPool</code> para criar um grupo de usuários do Amazon Cognito. AWS::Cognito::UserPoolClient Use o recurso <code>AWS::Cognito::UserPoolClient</code> para criar um cliente de grupo de usuários. AWS::Cognito::UserPoolGroup Use o recurso <code>AWS::Cognito::UserPoolGroup</code> para criar um grupo de usuários em um grupo de usuários do Amazon Cognito. AWS::Cognito::UserPoolUser Use o recurso <code>AWS::Cognito::UserPoolUser</code> para criar um usuário do grupo de usuários do Amazon Cognito.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		<u>AWS::Cognito::UserPoolUserToGroupAttachment</u> Use o recurso <code>AWS::Cognito::UserPoolUserToGroupAttachment</code> para anexar um usuário a um grupo de usuários do Amazon Cognito.	

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	28 de abril de 2017	SourceDetails Use a subpropriedade <code>MaximumExecutionFrequency</code> do recurso <code>AWS::Config::ConfigurationRule</code> para executar avaliações para uma regra personalizada usando um trigger periódico. AWS::EC2::Volume Agora oferecemos suporte aos Elastic Volumes para Amazon Elastic Block Store (Amazon EBS) no CloudFormation. Oferecemos suporte às atualizações <code>No interruption</code> em três propriedades: <code>VolumeType</code>, <code>Size</code> e <code>Iops</code>. AWS::EC2::SecurityGroup Use a propriedade <code>GroupName</code> para especificar um nome para o grupo de segurança do Amazon EC2. AWS::ECS::Service Há três novas propriedades para <code>AWS::ECS::Service</code>: <code>PlacementConstraints</code>, <code>PlacementStrategies</code> e <code>ServiceName</code>. AWS::ECS::TaskDefinition Use a propriedade <code>PlacementConstraints</code> para definir as restrições de posicionamento para tarefas no serviço. AWS::ElastiCache::ReplicationGroup Adição dos atributos <code>ConfigurationEndpoint.Address</code> e <code>ConfigurationEndpoint.Port</code> à função intrínseca <code>Fn::GetAtt</code>.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::ElasticLoadBalancingV2::LoadBalancer Use a propriedade <code>IpAddressType</code> para especificar o tipo de endereços IP usados pelas sub-redes do load balancer. AWS::EMR::Cluster O CloudFormation agora oferece suporte ao ajuste de escala automático para clusters do Amazon EMR. AWS::IAM::ManagedPolicy Use a propriedade <code>ManagedPolicyName</code> para especificar um nome personalizado para a política gerenciada do IAM. AWS::Lambda::Function Use a propriedade <code>Tags</code> para adicionar tags à função do Lambda. AWS::OpsWorks::Instance Adição dos seguintes atributos à função intrínseca <code>Fn::GetAtt</code> : <code>AvailabilityZone</code> , <code>PrivateDnsName</code> , <code>PrivateIp</code> e <code>PublicDnsName</code> . AWS::OpsWorks::UserProfile Use a propriedade <code>SshUsername</code> para especificar o nome SSH de um usuário. Adição do atributo <code>SshUsername</code> à função intrínseca <code>Fn::GetAtt</code> . AWS::Redshift::Cluster Use a propriedade <code>IamRoles</code> para fornecer uma lista de uma ou mais funções do AWS Identity and Access	

Alteração	Data de lançamento	Descrição	Versão da API
		Management que o cluster do Amazon Redshift pode usar para acessar outros serviços da AWS.	
Editar modelos em YAML e JSON usando o CloudFormation Designer	6 de abril de 2017	Ao criar modelos do CloudFormation usando o Designer, agora é possível editar o modelo em YAML e JSON no editor integrado. Você também pode converter modelos JSON em YAML e vice-versa, dependendo da linguagem de criação de modelo de sua preferência. Para obter mais informações, consulte O que é o CloudFormation Designer? .	15/05/2010
Novo recurso	6 de abril de 2017	AWS::SSM::Parameter Use o recurso <code>AWS::SSM::Parameter</code> para criar um parâmetro SSM no Parameter Store.	15/05/2010
<code>AWS::Include</code> Transformação	28 de março de 2017	Use a transformação <code>AWS::Include</code> para fazer referência a trechos reutilizáveis armazenados no bucket do Amazon S3. Para obter mais informações, consulte Transformação AWS::Include .	15/05/2010
Emparelhe uma Amazon VPC com outra conta	28 de março de 2017	Agora é possível usar o CloudFormation para emparelhar sua Amazon VPC com uma VPC em outra conta da AWS. Para obter mais informações, consulte Emparelhar com uma Amazon VPC em outra conta da AWS .	15/05/2010
Novo recurso	28 de março de 2017	AWS::ApiGateway::UsagePlanKey Use o recurso <code>AWS::ApiGateway::UsagePlanKey</code> para associar uma chave de plano de uso e determinar a quais usuários o plano de uso se aplica.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	28 de março de 2017	AWS::EC2::VPCPeeringConnection Use as propriedades <code>PeerOwnerId</code> e <code>PeerRoleArn</code> para realizar o emparelhamento com uma VPC em outra conta da AWS. Para obter mais informações, consulte Emparelhar com uma Amazon VPC em outra conta da AWS. AWS::IAM::InstanceProfile Use a propriedade <code>InstanceProfileName</code> para configurar um perfil de instância. AWS::Lambda::Function Use a propriedade <code>DeadLetterConfig</code> para configurar como o AWS Lambda manipula os eventos que não consegue processar. Não há mais suporte para Node.js v0.10 na propriedade de Runtime. AWS::Route53::HealthCheck Há sete tipos novos de subpropriedade de recurso para a propriedade HealthCheckConfig do <code>HealthCheckConfig</code>: <code>AlarmIdentifier</code>, <code>ChildHealthChecks</code>, <code>EnableSNI</code>, <code>HealthThreshold</code>, <code>InsufficientDataHealthStatus</code>, <code>Inverted</code> e <code>MeasureLatency</code>. AWS::SQS::Queue Use as propriedades <code>ContentBasedDeduplication</code> e <code>FifoQueue</code> para criar filas FIFO (primeiro a entrar, primeiro a sair) do Amazon Simple Queue Service.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::S3::Bucket Agora você pode especificar nomes de domínio IPv6 para os buckets do Amazon S3.	
Novos recursos	10 de fevereiro de 2017	AWS::StepFunctions::Activity Use o recurso <code>AWS::StepFunctions::Activity</code> para criar uma atividade do AWS Step Functions. AWS::StepFunctions::StateMachine Use o recurso <code>AWS::StepFunctions::StateMachine</code> para criar uma máquina de estado do Step Functions.	15/05/2010
Nova função intrínseca	17 de janeiro de 2017	Use a função <code>Fn::Split</code> para dividir uma string em uma lista de valores de string. Para obter mais informações, consulte Fn::Split .	15/05/2010
Suporte do console às importações de listagens	17 de janeiro de 2017	Use o console do CloudFormation para ver todas as pilhas que estão importando um valor de saída exportado. Para obter mais informações, consulte Como listar pilhas que importam um valor de saída exportado .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	17 de janeiro de 2017	AWS::AutoScaling::AutoScalingGroup A propriedade <code>LoadBalancerNames</code> pode ser atualizada sem substituir o grupo do Auto Scaling. AWS::ECS::TaskDefinition Adição das propriedades <code>NetworkMode</code> e <code>MemoryReservation</code> . AWS::RDS::DBCluster O CloudFormation oferece suporte a atualizações para a propriedade <code>Tags</code>. AWS::RDS::DBInstance Adição da propriedade <code>Timezone</code>. FirehoseAction Adição da propriedade <code>Separator</code> . AWS::OpsWorks::Instance Adição do atributo <code>PublicIp</code> da função intrínseca <code>Fn::GetAtt</code> .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	01 de dezembro de 2016	AWS::CodeBuild::Project Use o recurso <code>AWS::CodeBuild::Project</code> para criar um projeto do AWS CodeBuild que define como o CodeBuild desenvolverá o código fonte. AWS::SSM::Association Use o recurso <code>AWS::SSM::Association</code> para associar um documento do Amazon EC2 Systems Manager a instâncias do EC2. AWS::EC2::SubnetCidrBlock Use o recurso <code>AWS::EC2::SubnetCidrBlock</code> para associar um único bloco CIDR IPv6 a uma sub-rede da Amazon VPC. AWS::EC2::VPCCidrBlock Use o recurso <code>AWS::EC2::VPCCidrBlock</code> para associar um único bloco CIDR IPv6 fornecido pela Amazon a uma VPC da Amazon.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização de recursos para suporte a IPv6	01 de dezembro de 2016	AWS::EC2::Instance Adição das propriedades <code>Ipv6AddressCount</code> e <code>Ipv6Addresses</code> . AWS::EC2::NetworkAclEntry Adição da propriedade <code>Ipv6CidrBlock</code> . AWS::EC2::NetworkInterface Adição das propriedades <code>Ipv6AddressCount</code> e <code>Ipv6Addresses</code> . AWS::EC2::Route Adição da propriedade <code>DestinationIpv6CidrBlock</code> . AWS::EC2::SecurityGroupEgress Adição da propriedade <code>CidrIpv6</code>. AWS::EC2::SecurityGroupIngress Adição da propriedade <code>CidrIpv6</code>. AWS::EC2::SpotFleet Adição das propriedades <code>Ipv6AddressCount</code> e <code>Ipv6Addresses</code> para as interfaces de rede de especificação de inicialização. AWS::EC2::Subnet Adição do atributo <code>Ipv6CidrBlocks</code> da função intrínseca <code>Fn::GetAtt</code> . AWS::EC2::VPC Adição do atributo <code>Ipv6CidrBlocks</code> da função intrínseca <code>Fn::GetAtt</code> .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::SSM::Document Adição da propriedade <code>DocumentType</code> .	
Especificação de recurso	22 de novembro de 2016	Use a especificação de recurso do CloudFormation para desenvolver ferramentas que ajudarão você a criar modelos do CloudFormation. A especificação é um arquivo de texto em formato JSON e legível por máquina. Para obter mais informações, consulte Especificação de recursos do CloudFormation .	15/05/2010
Novos recursos	22 de novembro de 2016	AWS::OpsWorks::UserProfile Use o recurso <code>AWS::OpsWorks::UserProfile</code> para configurar o acesso SSH para os usuários que precisam de acesso a instâncias em uma pilha do OpsWorks. AWS::OpsWorks::Volume Use o recurso <code>AWS::OpsWorks::Volume</code> para registrar um volume do Amazon Elastic Block Store em uma pilha do OpsWorks.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	22 de novembro de 2016	AWS::OpsWorks::App Adição da propriedade <code>DataSources</code> . AWS::OpsWorks::Instance Adição das propriedades <code>BlockDeviceMappings</code> , <code>AgentVersion</code> , <code>ElasticIps</code> , <code>Hostname</code> , <code>Tenancy</code> e <code>Volumes</code>. AWS::OpsWorks::Layer Adição das propriedades <code>CustomJson</code> e <code>VolumeConfigurations</code> . AWS::OpsWorks::Stack Adição das propriedades <code>ElasticIps</code> , <code>EcsClusterArn</code> , <code>RdsDbInstances</code> , <code>CloneAppIds</code> , <code>ClonePermissions</code> e <code>SourceStackId</code> . AWS::RDS::DBInstance Adição da propriedade <code>CopyTagsToSnapshot</code> .	15/05/2010
Importações de listagens	22 de novembro de 2016	Importações de listagens de um valor de saída exportado para rastrear quais pilhas do CloudFormation estão importando o valor. Para obter mais informações, consulte Como listar pilhas que importam um valor de saída exportado.	15/05/2010
Transformações	17 de novembro de 2016	Especifique o AWS Serverless Application Model (AWS SAM) usado pelo CloudFormation para processar a sintaxe do AWS SAM para aplicações sem servidor. Para obter mais informações, consulte Transformação.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	17 de novembro de 2016	AWS::SNS::Subscription Use o recurso <code>AWS::SNS::Subscription</code> para assinar um endpoint em um tópico do Amazon Simple Notification Service.	15/05/2010
Recurso atualizado	17 de novembro de 2016	AWS::Lambda::Function Use a propriedade <code>Environment</code> para especificar pares de chaves-valores (variáveis de ambiente) que a função do AWS Lambda pode acessar. Use a propriedade <code>KmsKeyArn</code> para especificar uma chave do KMS usada pelo AWS Lambda para criptografar e descriptografar variáveis de ambiente.	15/05/2010
Novos comandos da ILC	17 de novembro de 2016	Upload de artefatos em um bucket do S Use o comando <code>package</code> para carregar artefatos locais que são referenciados em um modelo do CloudFormation para um bucket do S3. Como implantar modelos com transformações rapidamente Use o comando <code>deploy</code> para combinar as ações de criação e execução de conjunto de alterações em um único comando. Esse comando é útil para criar ou atualizar rapidamente pilhas que contêm transformações.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recurso atualizado	03 de novembro de 2016	AWS::CloudFront::Distribution Para a propriedade DistributionConfig, use a propriedade <code>HttpVersion</code> para especificar a versão HTTP mais recente que os visualizadores podem usar para se comunicar com o Amazon CloudFront. Para a propriedade ForwardedValues, use a propriedade <code>QueryStringCacheKeys</code> para especificar os parâmetros de string de consulta usados pelo CloudFront para determinar qual conteúdo será armazenado em cache.	15/05/2010
Listar exportações de pilha	03 de novembro de 2016	Use o console, a API ou a AWS CLI do CloudFormation para ver uma lista de todos os valores de saída exportados de uma região. Para obter mais informações, consulte Exportação dos valores de saída da pilha .	15/05/2010
Entrega contínua com pilhas	03 de novembro de 2016	Use o AWS CodePipeline para criar fluxos de trabalho de entrega contínua com pilhas do CloudFormation. Para obter mais informações, consulte Entrega contínua com o CodePipeline .	15/05/2010
Ignorar recursos durante a reversão	03 de novembro de 2016	Se você tiver uma pilha no estado <code>UPDATE_ROLLBACK_FAILED</code> , use o parâmetro <code>ResourcesToSkip</code> da ação <code>ContinueUpdateRollback</code> para ignorar recursos que o CloudFormation não pode reverter. Para obter mais informações, consulte a seção Solução de problemas em Falha na reversão de atualização .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Aprimoramento dos conjuntos de alterações	03 de novembro de 2016	Você pode criar uma nova pilha usando um conjunto de alterações .	15/05/2010
Recurso atualizado	12 de outubro de 2016	AWS::ElastiCache::CacheCluster Atualize a propriedade CacheNodeType sem substituir o cluster. AWS::ElastiCache::ReplicationGroup Você pode criar um grupo de replicação do Redis (modo de cluster habilitado) que pode conter vários grupos de nós (estilhaços), cada um com um cluster primário e réplicas de leitura. AWS::ElastiCache::SubnetGroup Use a propriedade CacheSubnetGroupName para especificar um nome para um grupo de sub-redes do Amazon ElastiCache.	15/05/2010
Novos recursos	06 de outubro de 2016	AWS::ApiGateway::UsagePlan Use o recurso AWS::ApiGateway::UsagePlan para especificar um plano de uso para APIs implantadas do Amazon API Gateway. AWS::CodeCommit::Repository Use o recurso AWS::CodeCommit::Repository para criar um repositório do CodeCommit hospedado pela Amazon Web Services.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	06 de outubro de 2016	AWS::ApiGateway::Authorizer Use a propriedade <code>ProviderARNs</code> para usar grupos de usuários do Amazon Cognito como autorizadores de API do Amazon API Gateway. AWS::ApiGateway::Deployment A propriedade <code>StageName</code> não é mais necessária. AWS::ElasticLoadBalancingV2::TargetGroup Para a função <code>GetAtt</code>, use o atributo <code>LoadBalancerArns</code> para recuperar o Nome de recurso da Amazon (ARN) dos load balancers que roteiam o tráfego para o grupo de destino. AWS::RDS::DBInstance Use as propriedades <code>Domain</code> e <code>DomainIAMRoleName</code> para usar a Autenticação do Windows quando os usuários se conectarem à instância de banco de dados do RDS. AWS::EC2::SecurityGroupEgress Use a propriedade <code>DestinationPrefixListId</code> para especificar o prefixo do serviço da AWS de um endpoint da Amazon VPC.	15/05/2010
Aprimoramento da referência entre pilhas	06 de outubro de 2016	Use funções intrínsecas para personalizar o valor <code>Name</code> de uma exportação ou fazer referência a um valor na função ImportValue .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Perfil de serviço do CloudFormation	26 de setembro de 2016	Use um perfil de serviço do AWS Identity and Access Management (IAM) em operações de pilha do CloudFormation. O CloudFormation usa as credenciais do perfil para fazer chamadas para os recursos de pilha em seu nome. Para obter mais informações, consulte Perfil de serviço do CloudFormation .	15/05/2010
Novo recurso	19 de setembro de 2016	Você pode usar o campo de saída <code>Export</code> e a função intrínseca <code>Fn::ImportValue</code> para que uma pilha consulte saídas de recursos em outra pilha. Para obter mais informações, consulte Saídas , Fn::ImportValue e Passo a passo: fazer referência a de saídas de recursos em outra pilha do CloudFormation .	15/05/2010
Suporte a YAML	19 de setembro de 2016	É possível usar o formato YAML para criar modelos do CloudFormation. O YAML também permite, por exemplo, que você adicione comentários aos modelos ou use a forma curta das funções intrínsecas. Para obter mais informações, consulte Formato de modelo do CloudFormation .	15/05/2010
Nova função intrínseca	19 de setembro de 2016	Use a função <code>Fn::Sub</code> para substituir variáveis em uma string de entrada por valores especificados por você. Para obter mais informações, consulte Fn::Sub .	15/05/2010
Novos recursos	19 de setembro de 2016	AWS::KMS::Alias Use o recurso <code>AWS::KMS::Alias</code> para criar um alias para uma AWS KMS key.	

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	19 de setembro de 2016	AWS::EC2::SpotFleet Para a propriedade <code>LaunchSpecifications</code> , use a propriedade <code>SpotPrice</code> a fim de especificar uma sugestão de preço para um tipo de instância específico. AWS::ECS::Cluster Use a propriedade <code>ClusterName</code> para especificar um nome para um cluster do Amazon Elastic Container Service. AWS::ECS::TaskDefinition Use a propriedade <code>TaskRoleArn</code> para especificar uma função do AWS Identity and Access Management que os contêineres do Amazon Elastic Container Service usam para fazer chamadas da AWS em seu nome. Use a propriedade <code>Family</code> para registrar uma definição de tarefa para uma família específica. AWS::Elasticsearch::Domain Use a propriedade <code>ElasticsearchVersion</code> para especificar qual versão do Elasticsearch será usada.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	11 de agosto de 2016	Use os seguintes recursos do Elastic Load Balancing Application Load Balancer para distribuir o tráfego de entrada de aplicativos para vários destinos, como instâncias do EC2, em várias Zonas de disponibilidade: • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	15/05/2010
Recurso atualizado	11 de agosto de 2016	AWS::AutoScaling::AutoScalingGroup Use a propriedade <code>TargetGroupARNs</code> para associar o grupo do Auto Scaling a um ou mais grupos de destino do Application Load Balancer. AWS::ECS::Service Para a propriedade de carga <code>LoadBalancers</code>, use a propriedade <code>TargetGroupArn</code> para associar um serviço do Amazon Elastic Container Service a um grupo de destino do Application Load Balancer.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	09 de agosto de 2016	O CloudFormation adicionou os seguintes recursos: AWS::ApplicationAutoScaling::ScalableTarget e AWS::ApplicationAutoScaling::ScalingPolicy Use uma política de escalabilidade do Aplicativo Auto Scaling para definir quando e como um recurso de destino será escalado. AWS::CertificateManager::Certificate Provisione um certificado do AWS Certificate Manager que você pode usar com outros serviços da AWS para permitir conexões seguras.	15/05/2010
Recursos atualizados	09 de agosto de 2016	O CloudFormation atualizou os seguintes recursos: AWS::CloudFront::Distribution Para a propriedade de configuração de distribuição <code>ViewerCertificate</code>, é possível especificar um certificado do AWS Certificate Manager. Para a propriedade de configuração de distribuição <code>Origin</code>, é possível especificar cabeçalhos personalizados e os protocolos SSL para origens personalizadas. AWS::EFS::FileSystem Você pode especificar o modo de desempenho para um sistema de arquivos do Amazon Elastic File System.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	20 de julho de 2016	AWS IoT Use o AWS IoT para declarar uma política do AWS IoT, um certificado X.509, uma associação entre uma política e um principal (um certificado X.509 ou outra credencial), um item do AWS IoT, uma associação entre um principal e um item, ou uma regra do AWS IoT. • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	15/05/2010
Recursos atualizados	20 de julho de 2016	O CloudFormation atualizou os seguintes recursos: AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User Use as propriedades de nome para especificar um nome personalizado para os recursos do AWS Identity and Access Management (IAM). AWS::ApiGateway::Method Para a propriedade <code>Integration</code>, você pode usar a propriedade <code>PassthroughBehavior</code> para especificar quando o Amazon API Gateway passará as solicitações para o backend direcionado. AWS::ApiGateway::Model e AWS::ApiGateway::RestApi Você pode especificar objetos JSON para as propriedades <code>Schema</code> e <code>Body</code>.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
UpdatePolicy do grupo do Auto Scaling	9 de junho de 2016	Para o atributo UpdatePolicy , use a propriedade AutoScalingReplacingUpdate para especificar se um grupo do Auto Scaling e as instâncias nele contidas serão substituídos quando você atualizar o grupo do Auto Scaling. Durante uma substituição, o CloudFormation retém o grupo do Auto Scaling antigo até criar o novo, para que o CloudFormation possa fazer a reversão para o grupo do Auto Scaling antigo se a atualização falhar. Para obter mais informações, consulte Atributo UpdatePolicy .	15/05/2010
Novo recurso	9 de junho de 2016	O CloudFormation adicionou os seguintes recursos: AWS::EC2::FlowLog Cria um log de fluxo do Amazon Elastic Compute Cloud que captura o tráfego de IP de uma interface de rede, sub-rede ou VPC especificada. AWS::KinesisFirehose::DeliveryStream Cria um fluxo de entrega que fornece dados de transmissão em tempo real a um destino, como o Amazon Simple Storage Service, o Amazon Redshift ou o Amazon OpenSearch Service.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	9 de junho de 2016	O CloudFormation atualizou os seguintes recursos: AWS::Kinesis::Stream Use a propriedade Name para especificar um nome para um fluxo do Amazon Kinesis. AWS::Lambda::Function Para a propriedade Code, você pode usar a propriedade ZipFile e o módulo de resposta do cfn nos ambientes de runtime nodejs4.3 . AWS::SNS::Topic O CloudFormation habilitou atualizações para o recurso de tópico do Amazon Simple Notification Service.	15/05/2010
Novo recurso	25 de abril de 2016	Use o recurso AWS::EC2::Host a fim de alocar um servidor físico totalmente dedicado para inicializar instâncias do EC2.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	25 de abril de 2016	AWS::EC2::Instance Use as propriedades <code>Affinity</code> e <code>HostId</code> para executar instâncias em um host dedicado do Amazon Elastic Compute Cloud. AWS::ECS::Service Use a propriedade <code>DeploymentConfiguration</code> para configurar quantas tarefas podem ser executados durante uma implantação. AWS::ECS::TaskDefinition O CloudFormation adicionou suporte a propriedades adicionais de definição de contêiner do Amazon Elastic Container Service. AWS::GameLift::Fleet Use as propriedades <code>MaxSize</code> e <code>MinSize</code> para especificar o número máximo e o número mínimo de instâncias do EC2 permitidas na frota do Amazon GameLift Servers. AWS::Lambda::Function Use a propriedade <code>FunctionName</code> para especificar um nome para a função do AWS Lambda. Você também pode usar o Python 2.7 para especificar uma função em linha.	15/05/2016

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	18 de abril de 2016	Amazon API Gateway Use os recursos do Amazon API Gateway para publicar, manter e monitorar APIs em qualquer escala. É possível criar APIs que os clientes podem chamar para acessar os serviços de backend, como aplicativos que executam instâncias EC2 ou código em execução no AWS Lambda. • AWS::ApiGateway::Account • AWS::ApiGateway::ApiKey • AWS::ApiGateway::Authorizer • AWS::ApiGateway::BasePathMapping • AWS::ApiGateway::ClientCertificate • AWS::ApiGateway::Deployment • AWS::ApiGateway::Method • AWS::ApiGateway::Model • AWS::ApiGateway::Resource • AWS::ApiGateway::RestApi • AWS::ApiGateway::Stage AWS::Events::Rule Crie uma regra do Amazon CloudWatch Events que monitore as alterações feitas nos recursos da AWS na sua conta (eventos). Se um evento de entrada corresponder às condições que você descreveu na regra, o Amazon CloudWatch Events enviará mensagens para os destinos especificados e os ativará, como funções do AWS Lambda ou tópicos do Amazon Simple Notification Service.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::WAF::SizeConstraintSet e AWS::WAF::XssMatchSet Use as duas regras do AWS WAF para verificar o tamanho de uma solicitação da web ou impedir os ataques de scripting entre sites.	
Novos recursos	31 de março de 2016	Use o recurso AWS::Lambda::Alias para criar aliases para as funções do AWS Lambda e o recurso AWS::Lambda::Version para criar versões das funções.	15/05/2010
Recursos atualizados	31 de março de 2016	O CloudFormation atualizou os seguintes recursos: AWS::EMR::Cluster e AWS::EMR::InstanceGroupConfig Use a propriedade <code>EbsConfiguration</code> para configurar volumes de armazenamento do Amazon Elastic Block Store para clusters do Amazon EMR ou grupos de instâncias. AWS::Lambda::Function Use a propriedade <code>VpcConfig</code> para habilitar funções do AWS Lambda para acessar recursos em uma VPC. AWS::S3::Bucket Para as regras de ciclo de vida do Amazon Simple Storage Service, você pode especificar várias regras de transição que determinam quando os objetos farão a transição para uma classe de armazenamento especificada.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Conjuntos de alterações	29 de março de 2016	Antes de atualizar pilhas, use conjuntos de alterações para ver como as alterações podem afetar seus recursos em execução. Para obter mais informações, consulte Atualizar pilhas usando conjuntos de alterações .	15/05/2010
Novos recursos	15 de março de 2016	Use os recursos AWS::GameLift::Alias , AWS::GameLift::Build e AWS::GameLift::Fleet para implantar servidores de jogos multijogador na AWS.	15/05/2010
Novos recursos	26 de fevereiro de 2016	O CloudFormation adicionou os seguintes recursos: AWS::ECR::Repository Crie repositórios do Amazon Elastic Container Registry nos quais os usuários podem enviar e extrair imagens de docker. AWS::EC2::NatGateway Use o gateway de conversão de endereços de rede (NAT) para permitir que as instâncias EC2 em uma sub-rede privada se conectem à Internet. AWS::Elasticsearch::Domain Crie domínios do Amazon OpenSearch Service que executam clusters legados do Elasticsearch OSS. AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step Use recursos do Amazon EMR para ajudar você a analisar e processar grandes quantidades de dados. Você pode criar clusters e, em seguida, executar trabalhos nesses clusters.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Recursos atualizados	26 de fevereiro de 2016	O CloudFormation atualizou os seguintes recursos: AWS::CloudTrail::Trail Use a propriedade <code>IsMultiRegionTrail</code> para especificar se uma trilha do AWS CloudTrail será criada na região em que você cria uma pilha ou em todas as regiões. AWS::Config::ConfigurationRecorder Para o grupo de gravação, use a propriedade <code>IncludeGlobalResourceTypes</code> global para registrar todos os tipos de recursos globais. AWS::RDS::DBCluster Use as propriedades <code>KmsKeyId</code> e <code>StorageEncrypted</code> para criptografar instâncias de banco de dados no cluster.	15/05/2010
Reter recursos	26 de fevereiro de 2016	Para pilhas no estado <code>DELETE_FAILED</code>, use o parâmetro <code>RetainResources</code> para reter recursos que o CloudFormation não pode excluir. Para obter mais informações, consulte Excluir falhas de pilha.	15/05/2010
Atualizar tags de pilha	26 de fevereiro de 2016	Você pode adicionar, modificar ou remover tags de pilha ao atualizar uma pilha. Para obter mais informações, consulte Atualizações de pilhas do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Continuar revertendo reversões de atualização com falha	25 de janeiro de 2016	Para uma pilha no estado UPDATE_ROLLBACK_FAILED , você pode continuar revertendo a atualização até que a pilha atinja um estado funcional. Desse modo, você pode retornar a pilha às suas configurações originais e tentar atualizá-la novamente. Para obter mais informações, consulte Continuar revertendo uma atualização .	15/05/2010
Novos modelos de amostra disponíveis para a região Ásia-Pacífico (Seul).	7 de janeiro de 2016	A seguinte coleção de modelos de amostra do CloudFormation é para a região ap-northeast-2: • Soluções de amostra • Estruturas de aplicativos • Services Para obter mais informações, consulte Trabalhar com modelos do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	28 de dezembro de 2015	O CloudFormation adicionou os seguintes recursos: AWS::DirectoryService::MicrosoftAD Use o recurso Microsoft Active Directory para criar um diretório do Microsoft Active Directory na AWS. AWS::Logs::Destination e AWS::Logs::LogStream Use os recursos do Amazon CloudWatch Logs para criar um destino para processamento em tempo real dos dados de log ou criar fluxos de log, respectivamente. AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet, e AWS::WAF::WebACL Use os recursos do AWS WAF para controlar e monitorar solicitações da web no conteúdo.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualizações de recursos	28 de dezembro de 2015	O CloudFormation atualizou os seguintes recursos: AWS::CloudFront::Distribution Para a configuração de distribuição, use a propriedade <code>WebACLId</code> para associar uma lista de controle de acesso (ACL) da Web do AWS WAF a uma distribuição do Amazon CloudFront. Para o comportamento de cache e o comportamento de cache padrão, você pode especificar um valor de Tempo de vida (TTL) padrão e máximo. AWS::DynamoDB::Table Você pode criar, atualizar ou excluir um índice secundário global sem substituir a tabela do Amazon DynamoDB. AWS::S3::Bucket Use a propriedade <code>ReplicationConfiguration</code> para especificar quais objetos serão replicados e onde eles serão armazenados. Use as propriedades da propriedade <code>NotificationConfiguration</code> para especificar filtros de modo que o Amazon Simple Storage Service envie notificações para objetos especificados.	15/05/2010
Agrupamento e classificação de parâmetros	3 de dezembro de 2015	Use a chave de metadados AWS::CloudFormation::Interface para agrupar e classificar parâmetros no console do CloudFormation quando os usuários criarem ou atualizar em uma pilha com seu modelo.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualizar atributo de política	3 de dezembro de 2015	Para um atributo de atualização de política do Auto Scaling, use a propriedade <code>MinSuccessfulInstancesPercent</code> para especificar a porcentagem de instâncias que devem sinalizar o sucesso de uma atualização bem-sucedida.	15/05/2010
Novos recursos	3 de dezembro de 2015	O CloudFormation adicionou os seguintes recursos: AWS::CodePipeline::Pipeline e AWS::CodePipeline::CustomActionType Use os recursos do CodePipeline para criar um pipeline que descreve como as alterações de software passam por um processo de liberação. AWS::Config::ConfigurationRecorder, AWS::Config::DeliveryChannel, e AWS::Config::ConfigRule Use o recursos do AWS Config para monitorar alterações de configuração nos recursos específicos da AWS. AWS::KMS::Key Use o recurso do AWS Key Management Service (AWS KMS) para criar chaves gerenciadas pelo cliente no AWS KMS que os usuários podem usar para criptografar pequenas quantidades de dados. AWS::SSM::Document Use o Amazon EC2 Systems Manager para criar um documento que especifica as configurações de instâncias ativadas.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização de recursos	3 de dezembro de 2015	O CloudFormation atualizou os seguintes recursos: AWS::AutoScaling::LaunchConfiguration Especifique se os volumes do EBS são criptografados. AWS::AutoScaling::ScalingPolicy Você pode usar dois tipos de política diferentes (escalabilidade simples e por etapas) para especificar como um grupo do Auto Scaling é escalado quando um alarme do Amazon CloudWatch (CloudWatch) é violado. AWS::CloudTrail::Trail Use as propriedades do CloudWatch para enviar logs a um grupo de logs do CloudWatch. Você pode adicionar tags a uma trilha e especificar uma AWS KMS key que deseja usar para criptografar logs. AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentConfig, e AWS::CodeDeploy::DeploymentGroup Use as propriedades <code>ApplicationName</code>, <code>DeploymentConfigName</code> e <code>DeploymentGroupName</code> para especificar nomes personalizados para recursos do CodeDeploy. AWS::DynamoDB::Table Use a propriedade <code>StreamSpecification</code> para especificar configurações para capturar alterações em itens armazenados em uma tabela do Amazon DynamoDB (DynamoDB).	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::EC2::Instance Use a propriedade <code>SsmAssociations</code> para associar um documento do Amazon EC2 Systems Manager a uma instância. AWS::EC2::SpotFleet Use a propriedade <code>AllocationStrategy</code> para especificar como alocar a capacidade pretendida entre grupos spot. Use a propriedade <code>ExcessCapacityTerminationPolicy</code> para especificar como as instâncias serão encerradas se a capacidade pretendida estiver abaixo do tamanho da frota spot. AWS::Redshift::Cluster Use a propriedade <code>KmsKeyId</code> para especificar uma AWS KMS key para criptografar dados em um cluster do Amazon Redshift. AWS::WorkSpaces::Workspace Use as propriedades de criptografia para criptografar dados armazenados em volumes.	
Atualização de recurso	4 de novembro de 2015	Para o recurso AWS::EC2::Volume , use a propriedade <code>AutoEnableIO</code> para retomar operações de E/S automaticamente se os dados de um volume se tornarem inconsistentes.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	1 de outubro de 2015	O CloudFormation adicionou os seguintes recursos: AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentGroup, e AWS::CodeDeploy::DeploymentConfig Use o recursos do CodeDeploy para criar e aplicar implantações a instâncias EC2 ou on-premises. AWS::DirectoryService::SimpleAD Use o recurso Simple Active Directory para criar um Simple AD do Directory Service, que é um diretório compatível com o Microsoft Active Directory. AWS::EC2::PlacementGroup Use um grupo de posicionamento para criar um cluster de instâncias em uma rede de baixa latência. AWS::EC2::SpotFleet Use uma frota spot para executar uma coleção de instâncias spot que executam tarefas que podem ser interrompidas. AWS::Lambda::EventSourceMapping Use o recurso de mapeamento de origem de evento para especificar um fluxo como origem de evento para uma função do AWS Lambda (Lambda). AWS::Lambda::Permission Use uma permissão do Lambda para adicionar uma instrução à política de uma função do Lambda.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::Logs::SubscriptionFilter Use o filtro de assinatura para definir quais eventos de log serão entregues ao fluxo do Kinesis. AWS::RDS::DBCluster e AWS::RDS::DBClusterParameterGroup Use o cluster e recursos de grupo de parâmetros de cluster para criar um cluster de banco de dados do Amazon Aurora. AWS::WorkSpaces::Workspace Use o WorkSpaces para criar experiências de desktop baseadas em nuvem.	

Alteração	Data de lançamento	Descrição	Versão da API
Atualizações de recursos	1 de outubro de 2015	O CloudFormation atualizou os seguintes recursos: AWS::ElastiCache::ReplicationGroup Use a função intrínseca <code>Fn::GetAtt</code> para obter uma lista de portas e endereços de réplica somente leitura. AWS::OpsWorks::Stack Use a propriedade <code>AgentVersion</code> para especificar um agente do OpsWorks. AWS::OpsWorks::App Use a propriedade <code>Environment</code> para especificar variáveis de ambiente para um aplicativo do OpsWorks. AWS::S3::Bucket Para a propriedade NotificationConfiguration, você pode definir configurações de notificação para funções do Lambda e filas do Amazon Simple Queue Service (Amazon SQS).	15/05/2010
Chaves de condição do IAM	1 de outubro de 2015	Para políticas do AWS Identity and Access Management (IAM), use chaves de condição específicas do CloudFormation para especificar quando uma política do IAM entrará em vigor. Para obter mais informações, consulte o tópico sobre como Controlar o acesso com o AWS Identity and Access Management.	15/05/2010
CloudFormation Designer	1 de outubro de 2015	Use o CloudFormation Designer para criar e modificar modelos usando uma interface de arrastar e soltar.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	24 de agosto de 2015	Use o recurso AWS::EC2::VPCEndpoint para estabelecer uma conexão privada entre sua VPC e outro serviço da AWS.	15/05/2010
Atualizações de recursos	24 de agosto de 2015	O CloudFormation atualizou os seguintes recursos: AWS::ElasticBeanstalk::Environment Use a propriedade <code>Tags</code> para especificar as tags (pares de chaves-valores) para um ambiente do AWS Elastic Beanstalk (Elastic Beanstalk). AWS::Lambda::Function Para a propriedade <code>Code</code>, use a propriedade <code>ZipFile</code> para gravar o código-fonte da função do Lambda diretamente em um modelo. Atualmente, você pode usar a propriedade <code>ZipFile</code> apenas em ambientes de runtime <code>node.js</code>. Você pode, ainda, indicar um arquivo em um bucket do S3 para todos os ambientes de runtime, como <code>java8</code> e <code>node.js</code>. AWS::OpsWorks::Instance Use a propriedade <code>EbsOptimized</code> para indicar se uma instância é otimizada para E/S do Amazon Elastic Block Store (Amazon EBS). AWS::RDS::DBInstance Para a propriedade <code>SourceDBInstanceId</code>, você pode especificar uma instância de banco de dados em outra região para criar uma réplica de leitura entre regiões.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
URL do modelo do Amazon S3	24 de agosto de 2015	Para buckets habilitados para versionamento, você pode especificar um ID de versão em um URL de modelo do Amazon S3 ao criar ou atualizar uma pilha, como <code>https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTDW</code> .	15/05/2010
Novo recurso	3 de agosto de 2015	Use o recurso AWS::EFS::FileSystem para criar um sistema de arquivos do Amazon Elastic File System (Amazon EFS) e o recurso AWS::EFS::MountTarget para criar um ponto de montagem para um sistema de arquivos.	15/05/2010
Alteração de requisito de permissão	11 de junho de 2015	Ao criar ou atualizar um recurso AWS::RDS::DBInstance , você agora precisa também ter permissão para chamar a ação <code>ec2:DescribeAccountAttributes</code> .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	11 de junho de 2015	O CloudFormation adicionou os seguintes recursos: AWS::DataPipeline::Pipeline Use pipelines de dados para automatizar a movimentação e a transformação de dados. Recursos do Amazon Elastic Container Service Use os recursos AWS::ECS::Service, AWS::ECS::Cluster e AWS::ECS::TaskDefinition para criar contêineres do Docker em um cluster de instâncias do EC2. AWS::ElastiCache::ReplicationGroup Use grupos de replicação para criar uma coleção de nós com um cluster de leitura/gravação primário e um máximo de cinco clusters somente leitura secundários. AWS::IAM::ManagedPolicy Use políticas gerenciadas para criar políticas na conta da AWS que você pode usar para aplicar permissões a usuários, grupos e perfis do IAM. AWS::Lambda::Function Use funções do Lambda para executar código em resposta a eventos. AWS::RDS::OptionGroup Use grupos de opções para facilitar a criação e o gerenciamento de bancos de dados do Amazon Relational Database Service (Amazon RDS).	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualizações de recursos	11 de junho de 2015	O CloudFormation atualizou os seguintes recursos: AWS::EC2::Subnet Use a propriedade <code>MapPublicIpOnLaunch</code> para atribuir automaticamente endereços IP públicos a instâncias em uma sub-rede. AWS::ElastiCache::CacheCluster Use a propriedade <code>SnapshotName</code> para restaurar dados de snapshot em um novo cluster de cache do Redis. AWS::IAM::User Para a propriedade <code>LoginProfile</code>, use a propriedade <code>PasswordResetRequired</code> para que os usuários sejam solicitados a definir uma nova senha ao efetuarem login no Console de gerenciamento da AWS. AWS::OpsWorks::Layer Use a propriedade <code>LifecycleEventConfiguration</code> para configurar eventos de ciclo de vida de uma camada do OpsWorks. AWS::S3::Bucket Para a propriedade <code>LifecycleConfiguration</code>, use as propriedades <code>NoncurrentVersionExpirationInDays</code> e <code>NoncurrentVersionTransition</code> para especificar regras de ciclo de vida para versões de objetos não atuais.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos tipos de parâmetro	19 de maio de 2015	Sempre que você usar o console do CloudFormation para criar ou atualizar uma pilha, poderá procurar valores de tipo de parâmetro específicos da AWS por ID, nome, nome ou valor da tag Name. O CloudFormation também adicionou suporte aos seguintes tipos de parâmetro específicos da AWS. Para obter mais informações, consulte Parâmetros. • <code>AWS::EC2::AvailabilityZone::Name</code> • <code>List<AWS::EC2::AvailabilityZone::Name></code> • <code>AWS::EC2::Instance::Id</code> • <code>List<AWS::EC2::Instance::Id></code> • <code>AWS::EC2::Image::Id</code> • <code>List<AWS::EC2::Image::Id></code> • <code>AWS::EC2::SecurityGroup::GroupName</code> • <code>List<AWS::EC2::SecurityGroup::GroupName></code> • <code>AWS::EC2::Volume::Id</code> • <code>List<AWS::EC2::Volume::Id></code> • <code>AWS::Route53::HostedZone::Id</code> • <code>List<AWS::Route53::HostedZone::Id></code>	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novos recursos	16 de abril de 2015	O CloudFormation adicionou os seguintes recursos: AWS::AutoScaling::LifecycleHook Use hooks de ciclo de vida do Auto Scaling para controlar o estado de uma instância depois que ela for executada ou encerrada. AWS::RDS::EventSubscription Use assinaturas de evento para obter notificações sobre eventos do Amazon RDS.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualizações de recursos	16 de abril de 2015	O CloudFormation atualizou os seguintes recursos: AWS::AutoScaling::AutoScalingGroup Use a propriedade <code>NotificationConfigurations</code> para especificar várias notificações. AWS::AutoScaling::LaunchConfiguration Use a propriedade <code>PlacementTenancy</code> para especificar a locação de instâncias. Use as propriedades <code>ClassicLinkVPCId</code> e <code>ClassicLinkVPCSecurityGroups</code> para vincular instâncias do EC2-Classical a uma VPC habilitada para ClassicLink. AWS::AutoScaling::ScalingPolicy Use a propriedade <code>MinAdjustmentStep</code> para especificar o número mínimo de instâncias adicionadas ou removidas durante um evento de escalabilidade. AWS::CloudFront::Distribution Para certificados de visualizador, use a propriedade <code>MinimumProtocolVersion</code> para especificar uma versão de protocolo mínima. Para comportamentos de cache, use a propriedade <code>CachedMethods</code> para especificar a quais métodos o cache do Amazon CloudFront (CloudFront) responde. Para origens, use o <code>OriginPath</code> para especificar um caminho usado pelo CloudFront para solicitar conteúdo.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		AWS::ElastiCache::CacheCluster Para clusters de cache do Memcached, use as propriedades <code>AZMode</code> e <code>PreferredAvailabilityZones</code> para especificar nós em várias Zonas de disponibilidade (AZs).	
		AWS::EC2::Volume Use a propriedade <code>KmsKeyId</code> para especificar uma chave gerenciada pelo cliente para volumes criptografados.	
		AWS::OpsWorks::Instance Use a propriedade <code>TimeBasedAutoScaling</code> para escalar automaticamente instâncias com base em uma programação especificada.	
		AWS::OpsWorks::Layer Use a propriedade <code>LoadBasedAutoScaling</code> para especificar políticas de escalabilidade com base em carga. Para configurações de volume, use as propriedades <code>VolumeType</code> e <code>Iops</code> para especificar um tipo de volume e o número de operações de E/S por segundo, respectivamente.	
		AWS::RDS::DBInstance Use a propriedade <code>CharacterSetName</code> para especificar um conjunto de caracteres para mecanismos de banco de dados com suporte. Use a propriedade <code>StorageEncrypted</code> para indicar se instâncias de banco de dados serão criptografadas e o <code>KmsKeyId</code> para especificar uma	

Alteração	Data de lançamento	Descrição	Versão da API
		chave gerenciada pelo cliente para instâncias de banco de dados criptografadas. AWS::Route53::HealthCheck Use a propriedade <code>HealthCheckTags</code> para associar tags a verificações de integridade. AWS::Route53::HostedZone Use a propriedade <code>VPCs</code> para criar hosted zones privadas. Use a propriedade <code>HostedZoneTags</code> para associar tags a hosted zones.	
Novo seção de modelo	16 de abril de 2015	Adicione a seção Metadata aos modelos para incluir objetos JSON arbitrários que descrevem seus modelos, como os detalhes do design e da implementação.	15/05/2010
Atualização de recurso	8 de abril de 2015	Para o recurso AWS::CloudFormation::CustomResource , é possível especificar nomes do recurso da Amazon (ARN) da função do Lambda na propriedade <code>ServiceToken</code> .	15/05/2010
Atualização do Amazon RDS	24 de dezembro de 2014	O CloudFormation adicionou duas novas propriedades a instâncias de banco de dados do RDS. Você pode associar um grupo de opções a uma instância de banco de dados e especifique o tipo de armazenamento de instância de banco de dados. Para obter mais informações, consulte AWS::RDS::DBInstance .	15/05/2010
Atualização do Elastic Load Balancing	24 de dezembro de 2014	Você pode usar a propriedade <code>ConnectionSettings</code> para especificar por quanto tempo as conexões podem permanecer ociosas. Para obter mais informações, consulte AWS::ElasticLoadBalancing::LoadBalancer .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização do Route 53	6 de novembro de 2014	Agora, você pode provisionar e gerenciar zonas hospedadas do Route 53, verificações de integridade , conjuntos de registros de failover e conjuntos de registros de localização geográfica .	15/05/2010
Aprimoramento de atualização contínua do Auto Scaling	6 de novembro de 2014	Durante uma atualização, você pode usar o sinalizador <code>WaitOnResourceSignals</code> para instruir o CloudFormation a aguardar instâncias para sinalizar o sucesso. Dessa forma, o CloudFormation só atualizará o próximo lote de instâncias quando o lote atual estiver pronto. Para obter mais informações, consulte Atributo UpdatePolicy .	15/05/2010
Novos atributos <code>Fn::GetAtt</code> da VPC	6 de novembro de 2014	Considerando um ID da VPC, você pode recuperar o grupo de segurança e a Network ACL padrão dessa VPC. Para obter mais informações, consulte Fn::GetAtt .	15/05/2010
Novos tipos de parâmetro específicos da AWS	6 de novembro de 2014	Você pode especificar tipos de parâmetro da AWS nos modelos do CloudFormation. No console do CloudFormation, esses tipos de parâmetro fornecem uma lista suspensa de valores válidos. Com a API ou a AWS CLI, o CloudFormation pode validar rapidamente valores para esses tipos de parâmetro antes de criar ou atualizar uma pilha. Para obter mais informações, consulte Parâmetros .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atributo CreationPolicy	6 de novembro de 2014	Com o atributo CreationPolicy, é possível instruir o CloudFormation a aguardar até que as aplicações estejam prontas nas instâncias do EC2 para dar prosseguimento à criação da pilha. Você pode usar uma política de criação, em vez de uma condição de espera e um identificador de condição de espera. Para obter mais informações, consulte Atributo CreationPolicy .	15/05/2010
Valores encaminhados do Amazon CloudFront	29 de setembro de 2014	Para comportamentos de cache, você pode encaminhar cabeçalhos para a origem. Consulte ForwardedValues .	15/05/2010
Atualização do AWS OpsWorks	29 de setembro de 2014	Para o Chef 11.10, você pode usar a propriedade ChefConfiguration para habilitar o Berkshelf. Você também pode usar os grupos de segurança integrados do AWS OpsWorks com as pilhas do AWS OpsWorks. Para obter mais informações, consulte AWS::OpsWorks::Stack .	15/05/2010
Suporte ao uso de tags do Elastic Load Balancing	29 de setembro de 2014	O AWS CloudFormation marca balanceadores de carga do Elastic Load Balancing com tags em nível de pilha. Você também pode adicionar suas próprias tags a um load balancer. Consulte AWS::ElasticLoadBalancing::LoadBalancer .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização da política de tópico do Amazon Simple Notification Service	29 de setembro de 2014	Agora você pode atualizar as políticas de tópico do Amazon SNS. Para obter mais informações, consulte AWS::SNS::TopicPolicy .	15/05/2010
Atualização da instância de banco de dados do RDS	5 de setembro de 2014	Você pode especificar se uma instância de banco de dados está voltada para a internet usando a propriedade <code>PubliclyAccessible</code> no recurso AWS::RDS::DBInstance .	15/05/2010
Atualização do atributo UpdatePolicy	05 de setembro de 2014	Você pode especificar uma política de atualização para um grupo do Auto Scaling que tenha uma ação programada associada. Para obter mais informações, consulte Atributo UpdatePolicy .	15/05/2010
Suporte a Amazon CloudWatch	10 de julho de 2014	O CloudFormation pode ser usado para provisionar e gerenciar grupos de logs e filtros de métricas do Amazon CloudWatch Logs (CloudWatch Logs). Para acessar mais informações, consulte AWS::Logs::LogGroup ou AWS::Logs::MetricFilter .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização da configuração de distribuição do Amazon CloudFront	17 de junho de 2014	Você pode especificar propriedades adicionais de configuração de distribuição do CloudFront: • As respostas de erro personalizadas definem mensagens de erro personalizadas para códigos de status HTTP 4xx e 5xx. • A classe de preço define o preço máximo que você está disposto a pagar pelo serviço do CloudFront. • As restrições definem quem pode visualizar seu conteúdo. • O certificado de visualizador especifica o certificado a ser usado quando os visualizadores usarem o HTTPS. • Para comportamentos de cache, você pode especificar métodos HTTP permitidos e indicar se os cookies serão encaminhados. Para obter mais informações, consulte AWS::CloudFront::Distribution.	15/05/2010
Atualização da instância EC2	17 de junho de 2014	É possível especificar se uma instância será interrompida ou encerrada quando você invocar o comando de desligamento de sistema operacional da instância. Para obter mais informações, consulte AWS::EC2::Instance .	15/05/2010
Atualização de volume do EBS	17 de junho de 2014	Você pode usar os volumes criptografados do EBS com tipos de instância compatíveis. Para obter mais informações, consulte AWS::EC2::Volume .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Nova conexão de emparelhamento da Amazon VPC	17 de junho de 2014	O CloudFormation pode ser usado para criar uma conexão de emparelhamento da Amazon Virtual Private Cloud (Amazon VPC), que estabelece uma conexão de rede entre duas VPCs. Para obter mais informações, consulte AWS::EC2::VPCPeeringConnection .	15/05/2010
Atualização do grupo do Amazon EC2 Auto Scaling	17 de junho de 2014	Você pode especificar um placement group de cluster existente no qual serão executadas instâncias de um grupo do Amazon EC2 Auto Scaling. Para obter mais informações, consulte AWS::AutoScaling::AutoScalingGroup .	15/05/2010
Suporte a AWS CloudTrail	17 de junho de 2014	O CloudFormation oferece suporte ao AWS CloudTrail, que pode capturar chamadas de API efetuadas na conta da AWS e publicar os logs em um local que você designar. Para obter mais informações, consulte AWS::CloudTrail::Trail .	15/05/2010
Atualizar aprimoramentos de pilha	12 de maio de 2014	O CloudFormation oferece suporte a recursos adicionais para atualização de pilhas: • É possível atualizar os parâmetros de pilha do CloudFormation sem enviar novamente o modelo da pilha. • É possível adicionar ou remover tópicos de notificação do Amazon SNS para uma pilha do CloudFormation. Para obter mais informações, consulte Atualizações de pilhas do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Suporte a Amazon Kinesis	6 de maio de 2014	É possível usar o CloudFormation para criar fluxos do Amazon Kinesis que capturam e transportam registros de dados a partir de fontes de dados. Para obter mais informações, consulte AWS::Kinesis::Stream .	15/05/2010
Novas propriedades do buckets do S3	5 de maio de 2014	O CloudFormation é compatível com propriedades adicionais do bucket do S3: • O compartilhamento de recursos entre origens (CORS) define o compartilhamento de recursos entre origens de objetos em um bucket. • O ciclo de vida define como o Amazon S3 gerencia objetos durante sua vida útil. • A política de registro de acesso captura informações sobre solicitações feitas no bucket. • As notificações definem quais eventos serão relatados e a qual tópico do Amazon SNS serão enviadas mensagens. • O versionamento permite diversas variantes de todos os objetos em um bucket. • As regras de redirecionamento e de roteamento regem o comportamento de redirecionamento das solicitações feitas no endpoint de site de um bucket. Para obter mais informações, consulte AWS::S3::Bucket.	15/05/2010
Suporte a Amazon EC2 Auto Scaling	5 de maio de 2014	O CloudFormation é compatível com a coleção de métricas de um grupo do Auto Scaling. Para obter mais informações, consulte AWS::AutoScaling::AutoScalingGroup .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atualização do Fn::If	5 de maio de 2014	Você pode usar a função intrínseca Fn::If na seção de saída de um modelo. Para obter mais informações, consulte Funções de condição .	15/05/2010
Registro de API com AWS CloudTrail	2 de abril de 2014	É possível utilizar o AWS CloudTrail (CloudTrail) para registrar em log solicitações do CloudFormation. Com o CloudTrail, é possível obter um histórico de chamadas de API do CloudFormation para sua conta. Para obter mais informações, consulte Registrar em log chamadas de API do CloudFormation com o AWS CloudTrail .	15/05/2010
Atualização do Elastic Load Balancing	20 de março de 2014	Você pode especificar uma política de registro de acesso para capturar informações sobre solicitações feitas no load balancer. Também é possível especificar uma política de drenagem de conexão que descreve como manipular solicitações em processamento quando o registros das instâncias são cancelados ou quando as instâncias não estão íntegras. Para obter mais informações, consulte AWS::ElasticLoadBalancing::LoadBalancer .	15/05/2010
Suporte a OpsWorks	3 de março de 2014	É possível usar o CloudFormation para provisionar e gerenciar pilhas do OpsWorks. Para obter mais informações, consulte AWS::OpsWorks::Stack ou Trechos de modelos do AWS OpsWorks .	15/05/2010
Aumento do limite de tamanho do modelo do Amazon S3	18 de fevereiro de 2014	Você pode especificar tamanhos de modelo até 460.800 bytes no Amazon S3.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Suporte a Amazon Redshift	10 de fevereiro de 2014	É possível usar o CloudFormation para provisionar e gerenciar clusters do Amazon Redshift. Para obter mais informações, consulte Trechos de modelos do Amazon Redshift ou AWS::Redshift::Cluster .	15/05/2010
Atualização de buckets do S3 e de políticas de bucket	10 de fevereiro de 2014	Você pode atualizar algumas propriedades do bucket do S3 e dos recursos de política de bucket. Para acessar mais informações, consulte AWS::S3::Bucket ou AWS::S3::BucketPolicy .	15/05/2010
Atualização de ambientes do Elastic Beanstalk e de versões de aplicativo	10 de fevereiro de 2014	Você pode atualizar configurações de ambiente do Elastic Beanstalk e versões de aplicativo. Para obter mais informações, consulte AWS::ElasticBeanstalk::Environment , AWS::ElasticBeanstalk::ConfigurationTemplate ou AWS::ElasticBeanstalk::ApplicationVersion .	15/05/2010
Atualização do Amazon SQS	29 de janeiro de 2014	Você pode especificar uma dead letter queue para uma fila do Amazon SQS. Para obter mais informações, consulte AWS::SQS::Queue .	15/05/2010
Ações programadas do Auto Scaling	27 de janeiro de 2014	Você pode escalar o número de instâncias EC2 em um grupo de Auto Scaling com base em uma programação. Ao usar uma programação, você pode escalar aplicativos em resposta a alterações de carga previsíveis. Para obter mais informações, consulte AWS::AutoScaling::ScheduledAction .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Índices secundários do DynamoDB	27 de janeiro de 2014	Você pode criar índices secundários locais e globais para bancos de dados do DynamoDB. Ao usar índices secundários, você pode acessar dados de forma eficiente com atributos que não a chave primária. Para obter mais informações, consulte AWS::DynamoDB::Table .	15/05/2010
Atualização do Auto Scaling	2 de janeiro de 2014	Você pode especificar um ID de instância para um grupo do Auto Scaling ou uma configuração de execução. Também é possível especificar propriedades adicionais de dispositivo de blocos do Auto Scaling. Para acessar mais informações, consulte AWS::AutoScaling::AutoScalingGroup ou AWS::AutoScaling::LaunchConfiguration .	15/05/2010
Atualização do Amazon SQS	2 de janeiro de 2014	Você pode atualizar filas do SQS e especificar propriedades adicionais. Para obter mais informações, consulte AWS::SQS::Queue .	15/05/2010
Aumentos de limite	2 de janeiro de 2014	É possível especificar até 60 parâmetros e 60 saídas nos modelos do CloudFormation.	15/05/2010
Novo console	19 de dezembro de 2013	O novo console do CloudFormation adiciona recursos como atualização automática dos eventos de pilha e organização em ordem alfabética dos parâmetros de pilha.	15/05/2010
Balanceamento de carga entre zonas	19 de dezembro de 2013	Com o balanceamento de carga entre zonas, você pode rotear o tráfego para instâncias de backend em todas as Zonas de disponibilidade (AZs). Para obter mais informações, consulte AWS::ElasticLoadBalancing::LoadBalancer .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Níveis de ambiente do AWS Elastic Beanstalk	19 de dezembro de 2013	Você pode especificar se o AWS Elastic Beanstalk provisionará recursos para oferecer suporte a um servidor web ou manipular tarefas de processamento em segundo plano. Para obter mais informações, consulte AWS::ElasticBeanstalk::Environment .	15/05/2010
Nome de recurso	19 de dezembro de 2013	Você pode atribuir nomes (IDs físicos) aos seguintes recursos: • Clusters do ElastiCache • Load balancers Elastic Load Balancing • Instâncias de banco de dados do RDS Para obter mais informações, consulte Tipo de nome.	15/05/2010
Suporte a VPN	22 de novembro de 2013	Você pode habilitar um gateway privado virtual (VGW) para propagar rotas para as tabelas de roteamento de uma VPC. Para obter mais informações, consulte AWS::EC2::VPNGatewayRoutePropagation .	15/05/2010
Criar recursos e atribuir propriedade condicionalmente	8 de novembro de 2013	Com o uso de parâmetros de entrada, você pode controlar a criação e as configurações de recursos de pilha designados, definindo as condições nos modelos do CloudFormation. Por exemplo, você pode usar as condições para criar recursos de pilha para um ambiente de produção. Usando o mesmo modelo, você pode criar recursos de pilha semelhantes com menor capacidade para um ambiente de teste. Para obter mais informações, consulte Funções de condição .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Impedir atualizações acidentais para recursos de pilha	8 de novembro de 2013	É possível impedir atualizações de pilha que podem resultar em alterações não intencionais nos recursos de pilha. Por exemplo, se você tiver uma pilha com uma camada de banco de dados que raramente será atualizada, defina uma política de pilha que impeça que a maioria dos usuários atualizem a camada de banco de dados. Para obter mais informações, consulte o tópico sobre como Prevenir atualizações para recursos de pilha .	15/05/2010
Recursos de nome	8 de novembro de 2013	Em vez de usar os IDs físicos gerados pelo CloudFormation, é possível atribuir nomes a determinados recursos. Os seguintes recursos do CloudFormation oferecem suporte a nomes • Alarmes do Amazon CloudWatch • Tabelas do DynamoDB • Aplicações e ambientes do AWS Elastic Beanstalk • Buckets do Amazon S3 • Tópicos do Amazon SNS • Filas do Amazon SQS Para obter mais informações, consulte Tipo de nome.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Atribuir tipos de recurso personalizados	8 de novembro de 2013	Nos modelos, é possível especificar seu próprio tipo de recurso para recursos personalizados do CloudFormation (<code>AWS::CloudFormation::CustomResource</code>). Ao usar seu nome de tipo de recurso personalizado, você poderá identificar rapidamente o tipo de recursos personalizados contido na sua pilha. Por exemplo, é possível especificar <code>"Type": "Custom::MyCustomResource"</code> . Para obter mais informações, consulte AWS::CloudFormation::CustomResource .	15/05/2010
Adicionar pseudoparâmetro	8 de novembro de 2013	Agora, você pode fazer referência ao AccountID da AWS nos modelos do CloudFormation fazendo referência ao pseudoparâmetro <code>AWS::AccountId</code> . Para obter mais informações, consulte Referência de pseudoparâmetros .	15/05/2010
Especificar pilhas nas políticas do IAM	8 de novembro de 2013	Você pode permitir ou negar usuários, grupos ou perfis do IAM para operar em pilhas específicas do CloudFormation. Por exemplo, você pode negar a ação de exclusão de pilha em um ID de pilha específico. Para obter mais informações, consulte o tópico sobre como Controlar o acesso com o AWS Identity and Access Management .	15/05/2010
Suporte à federação	14 de outubro de 2013	O CloudFormation oferece suporte a credenciais de segurança temporárias em perfis do IAM, que permitem cenários como federação e logon único no Console de gerenciamento da AWS. Você também pode fazer chamadas para o CloudFormation em instâncias do EC2 sem incorporar credenciais de segurança de longo prazo usando funções do IAM. Para obter mais informações sobre o IAM e o CloudFormation, consulte Controlar o acesso com o AWS Identity and Access Management .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Suporte a réplica de leitura do Amazon RDS	24 de setembro de 2013	Agora você pode criar réplicas de leitura do Amazon RDS em uma instância de banco de dados de origem. Para obter mais informações, consulte a propriedade <code>SourceDBInstanceIdentifier</code> no recurso AWS::RDS::DBInstance .	15/05/2010
Associar endereço IP público a instâncias em um grupo do Auto Scaling	19 de setembro de 2013	Agora você pode associar endereços IP públicos a instâncias em um grupo do Auto Scaling. Para obter mais informações, consulte AWS::AutoScaling::LaunchConfiguration .	15/05/2010
Suporte adicional a VPC	17 de setembro de 2013	O CloudFormation adiciona vários aprimoramentos para oferecer suporte à VPC e à funcionalidade da VPN - Você pode associar um endereço IP público e vários endereços IP privados a interfaces de rede do Amazon EC2. Para obter mais informações, consulte AWS::EC2::NetworkInterface. Você também pode associar um endereço IP privado primário a um endereço IP elástico (EIP). - Você pode habilitar o suporte a DNS e especificar nomes de host DNS. Para obter mais informações, consulte AWS::EC2::VPC. - Você pode especificar uma rota estática entre um gateway privado virtual e o gateway VPN. Para obter mais informações, consulte AWS::EC2::VPNConnectionRoute.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Suporte a Redis e grupos de segurança de VPC para Amazon ElastiCache	3 de setembro de 2013	Agora, você pode especificar o Redis como mecanismo de cache de um cluster do Amazon ElastiCache (ElastiCache). Você também pode atribuir grupos de segurança de VPC a clusters do ElastiCache. Para obter mais informações, consulte AWS::ElastiCache::CacheCluster .	15/05/2010
Criação, atualização e exclusão paralelas de pilha, e atualizações de pilha aninhada	12 de agosto de 2013	Agora o AWS CloudFormation cria, atualiza e exclui recursos paralelamente, melhorando o desempenho das operações. Se você atualizar um modelo de nível superior, o AWS CloudFormation atualizará automaticamente pilhas aninhadas que foram alteradas. Para obter mais informações, consulte Atualizações de pilhas do CloudFormation .	15/05/2010
Agora os grupos de segurança de VPC podem ser definidos em instâncias de banco de dados do RDS	28 de fevereiro de 2013	Agora é possível atribuir grupos de segurança de VPC a uma instância de banco de dados do RDS com o CloudFormation. Para obter mais informações, consulte a propriedade VPCSecurityGroups em AWS::RDS::DBInstance .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Implantações contínuas para grupos do Amazon EC2 Auto Scaling	20 de fevereiro de 2013	O CloudFormation agora oferece suporte a políticas de atualização em grupos do Amazon EC2 Auto Scaling, que descrevem como as instâncias do grupo do Amazon EC2 Auto Scaling são substituídas ou modificadas quando o grupo do Amazon EC2 Auto Scaling adiciona ou remove instâncias. Você pode modificar essas configurações durante a criação ou a atualização da pilha. Para obter mais informações e um exemplo, consulte Atributo UpdatePolicy.	15/05/2010
Cancelar e reverter a ação nas atualizações de pilha	20 de fevereiro de 2013	O CloudFormation oferece suporte ao cancelamento de uma atualização de pilha. O estado da pilha deve ser UPDATE_IN_PROGRESS quando a solicitação de atualização é realizada. Para obter mais informações, consulte os seguintes tópicos: • Como cancelar uma atualização de pilha • cancel-update-stack • CancelUpdateStack na Referência da API do CloudFormation	15/05/2010
Instâncias otimizadas para EBS nos grupos do Amazon EC2 Auto Scaling	20 de fevereiro de 2013	Agora, você pode provisionar instâncias otimizadas para EBS em grupos do Amazon EC2 Auto Scaling para o throughput dedicado ao Amazon Elastic Block Store (Amazon EBS) em instâncias autoscaled. A implementação é semelhante à do suporte lançado anteriormente para instâncias EC2 otimizadas do Amazon EBS. Para obter mais informações, consulte a nova propriedade de EbsOptimized em AWS::AutoScaling::LaunchConfiguration.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Nova documentação	21 de dezembro de 2012	O recurso AWS::EC2::Instance agora fornece uma propriedade <code>BlockDeviceMappings</code> para permitir que você defina mapeamentos de dispositivos de blocos para a instância do EC2. Com essa alteração, dois novos tipos foram adicionados: • BlockDeviceMapping • Dispositivo de blocos	15/05/2010
Nova documentação	21 de dezembro de 2012	Foram adicionadas novas seções para descrever os procedimentos de criação e visualização das pilhas usando o recém-reprojetado Console de gerenciamento da AWS. Você pode encontrá-los aqui: • Criar uma pilha via console do CloudFormation • Visualizar informações da pilha no console do CloudFormation	15/05/2010
Nova documentação	15 de novembro de 2012	Informações sobre recursos personalizados são fornecidas nos seguintes tópicos: • Recursos personalizados • AWS::CloudFormation::CustomResource • Referência de recursos personalizados	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Documentação atualizado	15 de novembro de 2012	O CloudFormation agora oferece suporte à especificação das operações de E/S provisionadas por segundo (IOPS) para instâncias de banco de dados do RDS. É possível definir este valor de 1000 a 10.000 em incrementos de 1000 IOPS usando a nova propriedade Iops em AWS::RDS::DBInstance. Para obter mais informações sobre a especificação de IOPS para instâncias de banco de dados do RDS, consulte Provisioned IOPS no Amazon Relational Database Service User Guide.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Documentação nova e atualizada	27 de agosto de 2012	Os tópicos foram reorganizados para fornecer informações específicas mais claras sobre como usar o Console de gerenciamento da AWS e a interface de linha de comandos (CLI) do CloudFormation. Informações sobre o uso de tags em pilhas do CloudFormation foram adicionadas, incluindo novas guias e tópicos de referência atualizados: • Novo tópico: Configurar opções de pilha. • Novas informações sobre tags na Referência da API do CloudFormation: CreateStack, Stack e Tag. Novas informações sobre como trabalhar com pilhas do Windows: • Imagens de máquina da Amazon (AMIs) do Microsoft Windows e modelos do CloudFormation • Fazer bootstrap de pilhas do Windows no CloudFormation Novo tópico: Usar expressões regulares em modelos do CloudFormation.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	25 de abril de 2012	O CloudFormation agora oferece suporte total a segurança da nuvem privada virtual (VPC) com o Amazon EC2. Você pode criar e preencher toda uma VPC com cada tipo de recurso de VPC (sub-redes, gateways, ACLs de rede, tabelas de rotas etc.) usando um único modelo do CloudFormation. A documentação dos seguintes tipos de recurso foi atualizada: • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer Novos tipos de recurso foram adicionados à documentação: • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry • AWS::EC2::Subnet	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
		• AWS::EC2::VPNGateway • AWS::EC2::CustomerGateway	
Novo recurso	13 de abril de 2012	O CloudFormation agora permite adicionar ou remover elementos de uma pilha ao atualizá-los. Atualizações de pilhas do CloudFormation foi atualizado e uma nova seção foi adicionada à demonstração: Alterar os recursos da pilha , que descreve como adicionar e remover recursos ao atualizar a pilha.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	2 de fevereiro de 2012	O CloudFormation agora oferece suporte a recursos em uma Amazon Virtual Private Cloud (Amazon VPC) existente. Com esta versão, você pode: • Executar uma instância EC2 dedicada em uma Amazon VPC existente. Para obter mais informações, consulte AWS::EC2::Instance. • Definir o atributo <code>SourceDestCheck</code> de uma instância do EC2 que reside em uma Amazon VPC existente. Para obter mais informações, consulte AWS::EC2::Instance. • Criar endereços IP elásticos em uma Amazon VPC existente. Para obter mais informações, consulte AWS::EC2::EIP. • Use o CloudFormation para criar grupos de segurança e regras de entrada/saída da Amazon VPC em uma VPC existente. Para obter mais informações, consulte AWS::EC2::SecurityGroup. • Associar um grupo do Auto Scaling a uma Amazon VPC existente, definindo a propriedade <code>VPCZoneId</code> do recurso <code>AWS::AutoScaling::AutoScalingGroup</code>. Para obter mais informações, consulte AWS::AutoScaling::AutoScalingGroup. • Anexar o load balancer do Elastic Load Balancing a uma sub-rede da Amazon VPC e criar grupos de segurança do load balancer. Para obter mais informações, consulte AWS::ElasticLoadBalancing::LoadBalancer. • Criar uma instância de banco de dados do RDS em uma Amazon VPC existente. Para obter mais informações, consulte AWS::RDS::DBInstance.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo recurso	2 de fevereiro de 2012	Agora você pode atualizar as propriedades dos seguintes recursos em uma pilha existente: • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::EIPAssociation • AWS::RDS::DBSubnetGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSecurityGroupIngress • AWS::Route53::RecordSetGroup Para obter uma lista completa dos recursos atualizáveis e detalhes sobre o que considerar ao atualizar uma pilha, consulte Atualizações de pilhas do CloudFormation.	15/05/2010
Guia reestruturado	2 de fevereiro de 2012	As seções existentes foram reorganizadas em novas seções: Trabalhar com modelos do CloudFormation e Gerenciar pilhas. Referência de modelos movida para o nível superior do índice. O tópico Estimar o custo da pilha do CloudFormation foi movido para a seção Conceitos básicos.	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Novo conteúdo	2 de fevereiro de 2012	Adicionadas novas seções: • Demonstração: atualização de uma pilha é um tutorial que aborda o processo de atualização de uma pilha LAMP. • Implantar aplicações no Amazon EC2 com o AWS CloudFormation descreve como usar scripts auxiliares do CloudFormation para implantar aplicações usando metadados armazenados no modelo. • Referência de scripts auxiliares do CloudFormation fornece material de referência sobre os scripts auxiliares do CloudFormation (cfn-init, cfn-get-metadata, cfn-signal e cfn-hup).	15/05/2010
Novo recurso	26 de maio de 2011	O CloudFormation agora fornece o comando <code>list-stacks</code> que permite listar pilhas filtradas por status da pilha. As pilhas excluídas podem ficar listadas por até 90 dias após terem sido excluídas. Para obter mais informações, consulte Descrição e listagem das pilhas .	15/05/2010
Novos atributos	26 de maio de 2011	Agora os comandos <code>describe-stack-resources</code> e <code>get-template</code> permitem que você obtenha informações das pilhas que foram excluídas por 90 dias após a exclusão. Para obter mais informações, consulte Listagem de recursos e Como recuperar um modelo .	15/05/2010
Novo link	1 de março de 2011	As informações do endpoint do CloudFormation agora estão localizadas no Referência geral da AWS. Para obter mais informações, consulte Regiões e endpoints em Referência geral da AWS .	15/05/2010

Alteração	Data de lançamento	Descrição	Versão da API
Lançamento inicial	25 de fevereiro de 2011	O lançamento público inicial do CloudFormation.	15/05/2010