Manual do usuário

AWS CodeDeploy



Versão da API 2014-10-06

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

AWS CodeDeploy: Manual do usuário

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

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

Table of Contents

O que é CodeDeploy? 1
Benefícios do AWS CodeDeploy 2
Visão geral das CodeDeploy plataformas de computação 3
Visão geral dos tipos de CodeDeploy implantação 11
Visão geral de uma implantação no local 13
Visão geral de uma blue/green implantação 14
Deixe seu comentário 19
Componentes primários 19
Aplicação 19
Plataforma de computação 20
Configuração de implantação 21
Grupo de implantação 21
Tipo de implantação 22
Tipo de implantação 23
Revisão 23
Perfil de serviço 24
Revisão de destino 24
Outros componentes 24
Implantações 25
Implantações em um AWS Plataforma de computação Lambda 25
Implantação azul/verde em uma plataforma de computação do Amazon ECS 29
Implantações em uma plataforma de EC2/On-Premises computação 41
Arquivos de especificação de aplicativo 48
AppSpec arquivos em uma plataforma de computação Amazon ECS 48
AppSpec arquivos em um AWS Lambda Plataforma de computação 49
AppSpec arquivos em uma plataforma de EC2/on-premises computação 49
Como o CodeDeploy agente usa o AppSpec arquivo 50
Introdução 51
Etapa 1: configuração 51
Inscreva-se para um Conta da AWS 51
Etapa 2: Criar um perfil de serviço 52
Criar um perfil de serviço (console) 54
Criar um perfil de serviço (CLI) 56
Obter o ARN do perfil de serviço (console) 60

Obter o ARN do perfil de serviço (CLI) ... 60
Etapa 3: limitar as permissões do CodeDeploy usuário ... 60
Etapa 4: criar um perfil de instância do IAM ... 64
Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (CLI) ... 65
Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (console) ... 70
Integrações de produtos e serviços ... 73
Integração com outros AWS serviços ... 73
Amazon EC2 Auto Scaling ... 82
Elastic Load Balancing ... 91
Integração com produtos e serviços de parceiros ... 96
GitHub ... 102
Exemplos de integração da comunidade ... 106
Publicações no blog ... 106
Tutoriais ... 108
Tutorial: WordPress implantar em uma instância que não seja do Windows ... 108
Etapa 1: iniciar uma instância do Amazon EC2 ... 109
Etapa 2: configurar o conteúdo de origem ... 112
Etapa 3: carregar seu aplicativo no Amazon S3 ... 118
Etapa 4: implantar seu aplicativo ... 122
Etapa 5: atualizar e reimplantar seu aplicativo ... 129
Etapa 6: limpar ... 133
Tutorial: implantar um aplicativo Hello World em uma instância do Windows Server ... 136
Etapa 1: iniciar uma instância do Amazon EC2 ... 137
Etapa 2: Configurar o conteúdo de origem ... 139
Etapa 3: carregar seu aplicativo no Amazon S3 ... 143
Etapa 4: implantar seu aplicativo ... 148
Etapa 5: atualizar e reimplantar seu aplicativo ... 153
Etapa 6: limpar ... 156
Tutorial: implantar um aplicativo em uma instância on-premises ... 160
Pré-requisitos ... 160
Etapa 1: configurar a instância on-premises ... 161
Etapa 2: criar uma revisão de aplicativo de amostra ... 161
Etapa 3: agrupar e carregar sua revisão de aplicativo no Amazon S3 ... 166
Etapa 4: implantar sua revisão de aplicativo ... 167
Etapa 5: verificar sua implantação ... 167
Etapa 6: limpar os recursos ... 167

Tutorial: implantar em um grupo do Auto Scaling 169
Pré-requisitos 170
Etapa 1: criar e configurar o grupo do Auto Scaling 171
Etapa 2: implantar o aplicativo no grupo do Auto Scaling 177
Etapa 3: verificar os resultados 187
Etapa 4: Aumentar o número de instâncias do Amazon EC2 no grupo do Auto Scaling. 189
Etapa 5: verificar novamente os resultados 190
Etapa 6: limpar 193
Tutorial: Implantar um aplicativo a partir de GitHub 195
Pré-requisitos 196
Etapa 1: configurar uma GitHub conta 196
Etapa 2: criar um GitHub repositório 196
Etapa 3: Carregar um aplicativo de amostra no seu GitHub repositório 199
Etapa 4: provisionar uma instância 204
Etapa 5: criar um aplicativo e um grupo de implantação 204
Etapa 6: implantar o aplicativo na instância 207
Etapa 7: monitorar e verificar a implantação 211
Etapa 8: Limpeza 212
Tutorial: Implantar um aplicativo no Amazon ECS 214
Pré-requisitos 216
Etapa 1: Atualizar o código do aplicativo do Amazon ECS 217
Etapa 2: criar o AppSpec arquivo 217
Etapa 3: usar o CodeDeploy console para implantar seu aplicativo 219
Etapa 4: limpar 223
Tutorial: Implantar um serviço do Amazon ECS com um teste de validação 223
Pré-requisitos 226
Etapa 1: Criar um receptor de teste 226
Etapa 2: Atualizar o código do aplicativo do Amazon ECS 226
Etapa 3: Criar uma função do Lambda de gancho do ciclo de vida 227
Etapa 4: atualize seu AppSpec arquivo 230
Etapa 5: Use o CodeDeploy console para implantar seu serviço Amazon ECS 231
Etapa 6: Visualize a saída da função de gancho do Lambda em registros CloudWatch 233
Etapa 7: limpar 234
Tutorial: Implantar uma função Lambda usando o SAM AWS 235
Pré-requisitos 236
Etapa 1: Configurar sua infraestrutura 236

Etapa 2: Atualizar a função do Lambda ... 252
Etapa 3: Implantar a função do Lambda atualizada ... 254
Etapa 4: Visualizar os resultados da implantação ... 257
Etapa 5: limpar ... 259
Trabalhando com o CodeDeploy agente ... 261
Sistemas operacionais suportados pelo CodeDeploy agente ... 262
Sistemas operacionais AMI do Amazon EC2 com suporte ... 262
Sistemas operacionais on-premises com suporte ... 262
Protocolo de comunicação e porta para o CodeDeploy agente ... 263
Histórico de versões do CodeDeploy agente ... 264
Gerenciando o CodeDeploy processo ... 283
Limpeza de revisões de aplicativo e arquivos de log ... 284
Arquivos instalados pelo CodeDeploy agente ... 285
Gerenciando as operações CodeDeploy do agente ... 288
Verifique se o CodeDeploy agente está em execução ... 289
Determine a versão do CodeDeploy agente ... 292
Instale o CodeDeploy agente ... 295
Atualize o CodeDeploy agente ... 308
Desinstale o CodeDeploy agente ... 311
Enviar registros CodeDeploy do agente para CloudWatch ... 312
Como trabalhar com instâncias do ... 318
Comparação de instâncias do Amazon EC2 com instâncias on-premises ... 318
Tarefas de instância para CodeDeploy ... 320
Marcação de instâncias para CodeDeploy implantações ... 321
Exemplo 1: Grupo de tag única, tag única ... 322
Exemplo 2: Grupo de tag única, várias tags ... 324
Exemplo 3: Grupos de várias tags, tags únicas ... 325
Exemplo 4: Grupos de várias tags, várias tags ... 328
Trabalhar com as instâncias do Amazon EC2 ... 332
Crie uma instância do Amazon EC2 para CodeDeploy ... 332
Crie uma instância do Amazon EC2 (modelo)CloudFormation ... 339
Configurar uma instância do Amazon EC2 ... 350
Trabalhar com instâncias no local ... 354
Pré-requisitos para configurar uma instância on-premises ... 355
Registrar uma instância local ... 357
Gerenciando operações de instâncias on-premises ... 394

Visualizar detalhes da instância ... 401
Exibir detalhes da instância (console) ... 402
Exibir detalhes da instância (CLI) ... 403
Integridade da instância ... 403
Status de integridade ... 404
Sobre o número mínimo de instâncias íntegras ... 405
Sobre o número mínimo de instâncias íntegras por zona de disponibilidade. ... 409
Como trabalhar com configurações de implantação ... 412
Configurações de implantação em uma plataforma de EC2/on-premises computação ... 412
Configurações de implantação predefinidas ... 412
Configurações de implantação em uma plataforma de computação do Amazon ECS ... 418
Configurações de implantação predefinidas para o Amazon ECS ... 418
Configurações de implantação para CloudFormation blue/green implantações (Amazon ECS) ... 419
Configurações de implantação em um AWS Lambda Plataforma de computação ... 420
Configurações de implantação predefinidas para o Lambda ... 420
Tópicos ... 421
Criar uma configuração de implantação ... 421
Criar uma configuração de implantação (console) ... 422
Criar uma configuração de implantação (AWS CLI) ... 425
Visualizar detalhes de configurações de implantação ... 426
Visualizar detalhes de configurações de implantação (console) ... 426
Visualizar configuração de implantação (CLI) ... 427
Excluir uma configuração de implantação ... 427
Como trabalhar com aplicações ... 428
Criar uma aplicação. ... 429
Criar um aplicativo para uma implantação no local (console) ... 431
Criar um aplicativo para uma blue/green implantação (console) ... 434
Criar um aplicativo para uma implantação do serviço do Amazon ECS (console) ... 439
Crie um aplicativo para um AWS Lambda implantação de funções (console) ... 441
Criar um aplicativo (CLI) ... 443
Visualizar detalhes de aplicativos ... 444
Visualizar detalhes de aplicativos (console) ... 444
Visualizar detalhes de aplicativos (CLI) ... 444
Criar uma regra de notificação ... 445
Renomear um aplicativo ... 448

Deleta a aplicação ... 448
Excluir um aplicativo (console) ... 449
Excluir um aplicativo (AWS CLI) ... 449
Como trabalhar com grupos de implantação ... 450
Grupos de implantação em implantações da plataforma de computação ... 450
Grupos de implantação em AWS Lambda implantações de plataformas de computação ... 450
Grupos de implantação em implantações da EC2/On-Premises Compute Platform ... 450
Tópicos ... 451
Criar um grupo de implantação ... 452
Criar um grupo de implantação para uma implantação no local (console) ... 452
Crie um grupo de implantação para uma EC2/On-Premises blue/green implantação (console) ... 456
Criar um grupo de implantação para uma implantação do Amazon ECS (console) ... 463
Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2 ... 465
Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS ... 466
Criar um grupo de implantação (CLI) ... 471
Visualizar detalhes do grupo de implantação. ... 473
Visualizar detalhes do grupo de implantação (console) ... 473
Visualizar detalhes do grupo de implantação (CLI) ... 473
Altere as configurações do grupo de implantação. ... 474
Alterar configurações do grupo de implantação (console) ... 474
Alterar configurações do grupo de implantação (CLI) ... 475
Configurar opções avançadas para um grupo de implantação ... 477
Exclua um grupo de implantação ... 480
Excluir um grupo de implantação (console) ... 481
Excluir um grupo de implantação (CLI) ... 481
Trabalhar com revisões de aplicativos ... 482
Planejar uma revisão ... 482
Adicionar um AppSpec arquivo ... 483
Adicionar um AppSpec arquivo para uma implantação do Amazon ECS ... 483
Adicionar um AppSpec arquivo para um AWS implantação do Lambda ... 486
Adicionar um AppSpec arquivo para uma EC2/On-Premises implantação ... 488
Escolher um tipo de repositório ... 493
Enviar uma revisão ... 495

Faça uma revisão usando o AWS CLI ... 497
Visualizar detalhes de revisões de aplicativo ... 499
Visualizar detalhes de revisões de aplicativo (console) ... 500
Visualizar detalhes de revisões de aplicativo (CLI) ... 500
Registrar uma revisão de aplicativo ... 502
Registre uma revisão no Amazon S3 com (CodeDeploy CLI) ... 502
Registre uma revisão GitHub com CodeDeploy (CLI) ... 503
Trabalhar com implantações ... 505
Crie um implantação do 506
Pré-requisitos de implantação ... 507
Criar uma implantação da plataforma de computação do Amazon ECS (console) ... 510
Crie um AWS Implantação da Lambda Compute Platform (console) ... 512
Crie uma implantação da EC2/On-Premises Compute Platform (console) ... 514
Crie uma implantação da Amazon ECS Compute Platform (CLI) ... 519
Crie um AWS Implantação da Lambda Compute Platform (CLI) ... 520
Crie uma implantação da EC2/On-Premises Compute Platform (CLI) ... 522
Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation ... 526
Visualizar detalhes de implantações ... 529
Visualizar detalhes de implantações (console) ... 530
Visualizar detalhes de implantações (CLI) ... 531
Visualizar dados de log de implantações ... 532
Visualize dados do arquivo de log no CloudWatch console da Amazon ... 532
Visualizar arquivos de log em uma instância ... 532
Interromper uma implantação ... 536
Interromper uma implantação (console) ... 536
Interromper uma implantação (CLI) ... 537
Reimplantar e reverter uma implantação ... 538
Reversões automáticas ... 538
Reversões manuais ... 539
Fluxo de trabalho de reversão e reimplantação ... 539
Comportamento de reversão com conteúdo existente ... 540
Implantar um aplicativo em uma AWS conta diferente ... 544
Etapa 1: criar um bucket do S3 em qualquer uma das contas ... 544
Etapa 2: conceder permissões de bucket do Amazon S3; ao perfil de instância do IAM da conta de produção ... 545
Etapa 3: criar recursos e um perfil entre contas na conta de produção ... 546

Etapa 4: carregar a revisão de aplicativo no bucket do Amazon S3 547
Etapa 5: assumir o perfil entre contas e implantar aplicativos 548
Validar um pacote de implantação em uma máquina local 548
Pré-requisitos 549
Criar uma implantação local 551
Exemplos 555
Monitorar implantações 557
Ferramentas automatizadas 557
Ferramentas manuais 559
Monitoramento de implantações com ferramentas da Amazon CloudWatch 560
Monitorando implantações com alarmes CloudWatch 560
Monitoramento de implantações com Amazon Events CloudWatch 562
Monitorando implantações com AWS CloudTrail 565
CodeDeploy informações em CloudTrail 565
Entendendo as entradas do arquivo de CodeDeploy log 566
Monitoramento de implantações com notificações de eventos do Amazon SNS 567
Conceder permissões do Amazon SNS a um perfil de serviço 569
Crie um gatilho para um CodeDeploy evento 570
Editar um gatilho em um grupo de implantação 577
Excluir um gatilho de um grupo de implantação 579
Formatos de dados JSON para gatilhos 581
Segurança 583
Proteção de dados 583
Privacidade do tráfego entre redes 585
Criptografia em repouso 585
Criptografia em trânsito 585
Gerenciamento de chave de criptografia 586
Gerenciamento de identidade e acesso 586
Público 586
Autenticação com identidades 587
Gerenciar o acesso usando políticas 588
Como AWS CodeDeploy funciona com o IAM 589
AWS políticas gerenciadas (predefinidas) para CodeDeploy 594
CodeDeploy atualizações para AWS políticas gerenciadas 603
Identity-based exemplos de políticas 607
Solução de problemas 614

CodeDeploy referência de permissões ... 616
Cross-service prevenção delegada confusa ... 625
Resposta a incidentes ... 627
Auditoria de todas as interações com CodeDeploy ... 627
Gerenciamento de incidentes e alertas ... 627
Validação de conformidade ... 628
Resiliência ... 629
Segurança da infraestrutura ... 629
Referência ... 630
AppSpec referência de arquivo ... 630
AppSpec arquivos em uma plataforma computacional Amazon ECS ... 631
AppSpec arquivos em um AWS Lambda Plataforma de computação ... 631
AppSpec arquivos em uma plataforma de EC2/on-premises computação ... 631
AppSpec Estrutura do arquivo ... 632
AppSpec Exemplo de arquivo ... 679
AppSpec Espaçamento de arquivos ... 685
Valide seu AppSpec arquivo e a localização do arquivo ... 687
Referência de configuração do agente ... 687
Tópicos relacionados ... 699
CloudFormation referência de modelo ... 699
Use CodeDeploy com a Amazon Virtual Private Cloud ... 703
Disponibilidade ... 704
Crie endpoints VPC para CodeDeploy ... 707
Configurar as permissões do CodeDeploy agente e do IAM ... 708
Referência ao kit de recursos ... 709
Nomes de bucket do kit de recursos por região ... 709
Conteúdo do kit de recursos ... 712
Exibir uma lista dos arquivos do kit de recursos ... 714
Fazer download dos arquivos do kit de recursos ... 716
Cotas ... 719
Solução de problemas ... 726
Tópicos gerais de solução de problemas ... 726
Lista de verificação geral para solução de problemas ... 727
CodeDeploy os recursos de implantação são suportados apenas em alguns AWS Regiões ... 728
Os procedimentos deste guia não coincidem com o CodeDeploy console ... 729

Perfis do IAM necessários não estão disponíveis 729
Usar alguns editores de texto para criar AppSpec arquivos e scripts de shell pode causar falhas nas implantações 730
O uso do Finder no macOS para agrupar uma revisão de aplicativo pode causar falhas na implantação 731
Solucionar problemas de EC2/On-Premises implantação 731
CodeDeploy erro de credenciais CommandPoller ausentes do plugin 732
A implantação falha com a mensagem “Falha na validação da mensagem assinada pelo PKCS7” 732
A implantação ou reimplantação dos mesmos arquivos nos mesmos locais de instância falha com um erro indicando que a implantação falhou porque um arquivo especificado já existe na localização 733
Caminhos de arquivo longos causam erros do tipo “Nenhum arquivo ou diretório” 736
Long-running processos podem fazer com que as implantações falhem 737
Solução de problemas em um evento de AllowTraffic ciclo de vida com falha sem nenhum erro relatado nos registros de implantação 738
Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação 739
Solução de problemas em um evento DownloadBundle de ciclo de vida de implantação com falha com UnknownError: não aberto para leitura 740
Solução de problemas de erros relacionados a todos os eventos de ciclo de vida ignorados 741
PowerShell Os scripts do Windows não usam a versão de 64 bits do Windows PowerShell por padrão 743
Solucionar problemas de implantação do Amazon ECS 744
Ocorre um tempo limite enquanto se aguarda o conjunto de tarefas de substituição 745
Ocorre um limite de tempo enquanto se aguarda a continuação de uma notificação 745
O perfil do IAM não tem permissões suficientes 746
A implantação atingiu o tempo limite enquanto aguardava um retorno de chamada de status 747
A implantação falhou porque uma ou mais funções de validação de evento do ciclo de vida falharam 747
Não foi possível atualizar o ELB devido ao seguinte erro: o grupo de destino do conjunto de tarefas primário deve estar atrás do receptor 748
Às vezes, minha implantação falha ao usar o ajuste de escala automático 749

Somente o ALB suporta roteamento de tráfego gradual; em vez disso, use o roteamento de AllAtOnce tráfego ao usar o grupo de implantação create/update 750
Embora minha implantação tenha sido bem-sucedida, o conjunto de tarefas de substituição falha nas verificações de integridade do Elastic Load Balancing, e meu aplicativo está inativo 750
Posso conectar vários balanceadores de carga a um grupo de implantação? 752
Posso realizar CodeDeploy blue/green implantações sem um balanceador de carga? 752
Como posso atualizar meu serviço Amazon ECS com novas informações durante uma implantação? 752
Solução de problemas AWS Problemas de implantação do Lambda 753
AWS Lambda as implantações falham após interromper manualmente uma implantação do Lambda que não tem reversões configuradas 753
Solução de problemas de grupo de implantação 754
Marcar uma instância como parte de um grupo de implantação não implanta automaticamente seu aplicativo na nova instância 754
Solução de problemas de instância 754
As tags devem ser definidas corretamente 754
AWS CodeDeploy o agente deve estar instalado e em execução nas instâncias 755
As implantações não falham por até uma hora quando uma instância é encerrada durante uma implantação 755
Analisando arquivos de log para investigar falhas de implantação em instâncias 755
Crie um novo arquivo de CodeDeploy log se ele tiver sido excluído acidentalmente 756
Solução de problemas de "InvalidSignatureException — A assinatura expirou: [hora] agora é anterior à [hora]” de erros de implantação 756
Solucionar problemas de GitHub token 757
Token OAuth GitHub inválido 757
Número máximo de tokens GitHub OAuth excedido 757
Solucionar problemas do Amazon EC2 Auto Scaling 758
Solução de problemas gerais do Amazon EC2 Auto Scaling 758
“CodeDeployRole não lhe dá permissão para realizar operações nas seguintes AWS serviço: AmazonAutoScaling "erro 759
As instâncias em um grupo de Amazon EC2 Auto Scaling são continuamente provisionadas e encerradas antes que uma revisão possa ser implantada 760
O encerramento ou a reinicialização de uma instância do Amazon EC2 Auto Scaling pode causar falhas nas implantações 761

Evite associar vários grupos de implantação a um único grupo de Amazon EC2 Auto Scaling 762
As instâncias do EC2 em um grupo Amazon EC2 Auto Scaling falham ao ser iniciadas e recebem o erro "Heartbeat Timeout" 762
Ganchos do ciclo de vida do Amazon EC2 Auto Scaling podem fazer com que as implantações automáticas em grupos do Amazon EC2 Auto Scaling parem ou falhem. 765
Erro “A implantação falhou porque nenhuma instância foi encontrada para seu grupo de implantação” 767
Códigos de erro 775
Tópicos relacionados 780
Recursos 781
Guias de referência e recursos de suporte 781
Amostras 781
Blogs 781
AWS kits e ferramentas de desenvolvimento de software 782
Histórico do documento 783
Atualizações anteriores 803
AWS Glossário 826
..... dcccxxvii

O que é CodeDeploy?

CodeDeploy é um serviço de implantação que automatiza implantações de aplicativos em instâncias do Amazon EC2, instâncias locais, funções Lambda sem servidor ou serviços do Amazon ECS.

Você pode implantar uma variedade quase ilimitada de conteúdos de aplicativos, incluindo:

- Código
- Funções sem servidor AWS Lambda
- Arquivos de configuração e da Web
- Executáveis
- Pacotes
- Scripts
- Arquivos de multimídia

CodeDeploy pode implantar conteúdo de aplicativo executado em um servidor e armazenado em buckets, GitHub repositórios ou repositórios Bitbucket do Amazon S3. CodeDeploy também pode implantar uma função Lambda sem servidor. Você não precisa fazer alterações no seu código existente antes de poder usar o CodeDeploy.

CodeDeploy torna mais fácil para você:

- Lançar novos recursos rapidamente.
- Atualize as versões da AWS Lambda função.
- Evite tempo de inatividade durante a implantação do aplicativo.
- Processar a complexidade da atualização de seus aplicativos sem muitos dos riscos associados a implantações manuais propensas a erro.

O serviço se adequa à sua infraestrutura para que você possa implantar uma instância ou milhares de maneira fácil.

CodeDeploy trabalha com vários sistemas para gerenciamento de configuração, controle de origem, integração contínua, entrega contínua e implantação contínua. Para obter mais informações, consulte Integrações de produtos.

O CodeDeploy console também fornece uma maneira de pesquisar rapidamente seus recursos, como repositórios, criar projetos, aplicativos de implantação e pipelines. Selecione Go to resource (Acessar recurso) ou pressione a tecla / e digite o nome do recurso. Qualquer correspondência aparecerá na lista. As pesquisas não diferenciam letras maiúsculas de minúsculas. Só é possível ver recursos para os quais você tem permissão de visualizar. Para obter mais informações, consulte Gerenciamento de identidade e acesso para AWS CodeDeploy.

Tópicos

- Benefícios do AWS CodeDeploy
- Visão geral das CodeDeploy plataformas de computação
- Visão geral dos tipos de CodeDeploy implantação
- Deixe seu comentário
- CodeDeploy componentes primários
- CodeDeploy implantações
- CodeDeploy arquivos de especificação do aplicativo (AppSpec)

Benefícios do AWS CodeDeploy

CodeDeploy oferece os seguintes benefícios:

- Aplicativos de servidor, sem servidor e de contêiner. CodeDeploypermite que você implante aplicativos tradicionais em servidores e aplicativos que implantam uma versão de AWS Lambda função sem servidor ou um aplicativo Amazon ECS.
- Implantações automatizadas. CodeDeploy automatiza totalmente suas implantações de aplicativos em seus ambientes de desenvolvimento, teste e produção. CodeDeployse expande com sua infraestrutura para que você possa implantar em uma instância ou em milhares.
- Minimize o tempo de inatividade. Se seu aplicativo usa a plataforma de EC2/On-Premises computação, CodeDeploy ajuda a maximizar a disponibilidade do aplicativo. Durante uma implantação local, CodeDeploy executa uma atualização contínua em todas as instâncias do Amazon EC2. Você pode especificar o número de instâncias a serem colocadas offline de cada vez para atualizações. Durante uma blue/green implantação, a revisão mais recente do aplicativo é instalada nas instâncias substitutas. O tráfego é roteado novamente para essas instâncias no momento que você escolher, seja imediatamente ou assim que o teste do novo ambiente terminar. Para ambos os tipos de implantação, o CodeDeploy controla a integridade do aplicativo de acordo com regras que você configura.

- Interromper e reverter. Em caso de erros, é possível interromper e reverter implantações automática ou manualmente.
- Controle centralizado. Você pode iniciar e acompanhar o status de suas implantações por meio do CodeDeploy console ou do AWS CLI. Você recebe um relatório que lista quando cada revisão de aplicativo foi implantada e em quais instâncias do Amazon EC2.
- Fácil de adotar. CodeDeploy é independente de plataforma e funciona com qualquer aplicativo. Você pode facilmente reutilizar seu código de configuração. CodeDeploy também pode se integrar ao seu processo de lançamento de software ou à cadeia de ferramentas de entrega contínua.
- Implantações simultâneas. Se você tiver mais de um aplicativo que usa a plataforma de EC2/On-Premises computação, CodeDeploy pode implantá-los simultaneamente no mesmo conjunto de instâncias.

Visão geral das CodeDeploy plataformas de computação

CodeDeploy é capaz de implantar aplicativos em três plataformas de computação:

- EC2/On-Premises: descreve instâncias de servidores físicos que podem ser instâncias de nuvem do Amazon EC2, servidores locais ou ambos. Os aplicativos criados usando a plataforma de EC2/On-Premises computação podem ser compostos por arquivos executáveis, arquivos de configuração, imagens e muito mais.

  As implantações que usam a plataforma de EC2/On-Premises computação gerenciam a forma como o tráfego é direcionado às instâncias usando um tipo local ou blue/green de implantação. Para obter mais informações, consulte Visão geral dos tipos de CodeDeploy implantação.
- AWS Lambda: usado para implantar aplicativos que consistem em uma versão atualizada de uma função Lambda. AWS Lambda gerencia a função Lambda em um ambiente computacional sem servidor composto por uma estrutura computacional de alta disponibilidade. Toda a administração dos recursos computacionais é realizada por AWS Lambda. Para obter mais informações, consulte Computação e Aplicativos com tecnologia sem servidor. Para obter mais informações sobre AWS Lambda as funções Lambda, consulte. AWS Lambda

  Você pode gerenciar a maneira como o tráfego é desviado para as versões da função do Lambda atualizadas durante uma implantação, escolhendo uma configuração de canário, linear ou todas de uma só vez.

- Amazon ECS: usado para implantar um aplicativo em contêineres do Amazon ECS como um conjunto de tarefas. CodeDeploy executa uma blue/green implantação instalando uma versão atualizada do aplicativo como um novo conjunto de tarefas de substituição. CodeDeploy redireciona o tráfego de produção do conjunto de tarefas original do aplicativo para o conjunto de tarefas de substituição. O conjunto de tarefas original é encerrado após uma implantação bem-sucedida. Para obter mais informações sobre o Amazon ECS, consulte Amazon Elastic Container Service.

  Você pode gerenciar a maneira como o tráfego é deslocado para o conjunto de tarefas atualizado durante uma implantação escolhendo uma configuração canário, linear ou todas de uma só vez.

  > **Note**
  >
  > As blue/green implantações do Amazon ECS são suportadas usando e. CodeDeploy CloudFormation Os detalhes dessas implantações são descritos nas seções subsequentes.

A tabela a seguir descreve como CodeDeploy os componentes são usados com cada plataforma de computação. Para obter mais informações, consulte:

- Trabalhando com grupos de implantação em CodeDeploy
- Trabalhando com implantações em CodeDeploy
- Trabalhando com configurações de implantação em CodeDeploy
- Trabalhando com revisões de aplicativos para CodeDeploy
- Trabalhando com aplicativos em CodeDeploy

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Grupo de implantação	Implanta uma revisão em um conjunto de instâncias.	Implanta uma nova versão de uma função do Lambda com tecnologia sem servidor em uma infraestrutura de	Especifica o serviço do Amazon ECS com o aplicativo em

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
		computação de alta disponibilidade.	contêineres para fazer a implantação como um conjunto de tarefas, um receptor de produção e de teste opcional usado para fornecer o tráfego para o aplicativo implantado, quando rotear novamente o tráfego e encerrar o conjunto de tarefas original do aplicativo implantado e configurações opcionais de trigger,

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			alarme e reversão.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Implantação	Implanta uma nova revisão que consiste em um aplicativo e um AppSpec arquivo. O AppSpec especifica como implantar o aplicativo nas instâncias em um grupo de implantação.	Passa o tráfego de produção de uma versão de uma função do Lambda para uma nova versão da mesma função. O AppSpec arquivo especifica qual versão da função Lambda deve ser implantada.	Implanta uma versão atualizada de um aplicativo em contêineres do Amazon ECS como um novo conjunto de tarefas substituto. CodeDeploy redireciona o tráfego de produção do conjunto de tarefas com a versão original para o novo conjunto de tarefas de substituição com a versão

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			atualizada. Quando a implantação for concluída, o conjunto de tarefas original é encerrado.
Configuração de implantação	Configurações que determinam a velocidade de implantação e o número mínimo de instâncias que devem permanecer íntegras em qualquer ponto de uma implantação.	Configurações que determinam como o tráfego é deslocado para as versões atualizadas da função do Lambda.	Configurações que determinam como o tráfego é deslocado para o conjunto de tarefas do Amazon ECS atualizado.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
Revisão	Uma combinação de um AppSpec arquivo e arquivos de aplicativo, como executáveis, arquivos de configuração e assim por diante.	Um AppSpec arquivo que especifica qual função Lambda implantar e funções Lambda que podem executar testes de validação durante ganchos de eventos do ciclo de vida da implantação.	Um AppSpec arquivo que especifica: • A definição de tarefa do Amazon ECS para o serviço do Amazon ECS com o aplicativo em contêineres que será implantado. • O contêiner onde seu aplicativo atualizado é implantado.

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			• Uma porta para o contêiner em que o tráfego de produção é redirecionado. • As configurações de rede opcionais e as funções do Lambda que podem executar testes de validação durante os hooks de eventos de ciclo de vida de

CodeDeploy componente	EC2/On-Premises	AWS Lambda	Amazon ECS
			implantação.
Aplicação	Uma coleção de grupos de implantação e revisões. Um EC2/On-Premises aplicativo usa a plataforma de EC2/On-Premises computação.	Uma coleção de grupos de implantação e revisões. Um aplicativo usado para uma AWS Lambda implantação usa a plataforma de computação AWS Lambda sem servidor.	Uma coleção de grupos de implantação e revisões. Um aplicativo usado para uma implantação do Amazon ECS usa a plataforma de computação do Amazon ECS.

Visão geral dos tipos de CodeDeploy implantação

CodeDeploy fornece duas opções de tipo de implantação:

- In-place implantação: o aplicativo em cada instância do grupo de implantação é interrompido, a revisão mais recente do aplicativo é instalada e a nova versão do aplicativo é iniciada e validada. Você pode usar um balanceador de carga de forma que cada registro de instância é cancelado durante sua implantação e, em seguida, restaurado para o serviço após a conclusão da implantação. Somente implantações que usam a plataforma de EC2/On-Premises computação

podem usar implantações no local. Para obter mais informações sobre implantações no local, consulte Visão geral de uma implantação no local.

> **Note**
>
> AWS As implantações do Lambda e do Amazon ECS não podem usar um tipo de implantação local.

- Blue/green implantação: o comportamento da sua implantação depende da plataforma de computação que você usa:
  - Blue/green em uma plataforma de EC2/On-Premises computação: as instâncias em um grupo de implantação (o ambiente original) são substituídas por um conjunto diferente de instâncias (o ambiente substituto) usando estas etapas:
    - As instâncias são provisionadas para o ambiente de substituição.
    - A revisão de aplicativo mais recente será instalada nas instâncias de substituição.
    - Um tempo de espera opcional ocorre para atividades como teste de aplicativos e verificação do sistema.
    - As instâncias no ambiente de substituição são registradas com um ou mais balanceadores de carga do Elastic Load Balancing, fazendo com que o tráfego seja redirecionado para elas. O registro das instâncias no ambiente original é cancelado e pode ser encerrado ou mantido em execução para outros usos.

    > **Note**
    >
    > Se você usa uma plataforma EC2/On-Premises computacional, saiba que as blue/green implantações funcionam somente com instâncias do Amazon EC2.

  - Blue/green em uma plataforma computacional AWS Lambda ou Amazon ECS: o tráfego é deslocado em incrementos de acordo com uma configuração de implantação canária, linear ou completa.
  - Blue/green implantações por meio de CloudFormation: o tráfego é transferido de seus recursos atuais para seus recursos atualizados como parte de uma atualização de CloudFormation pilha. Atualmente, somente blue/green implantações do ECS são suportadas.

Para obter mais informações sobre blue/green implantações, consulteVisão geral de uma blue/green implantação.

> **Note**
>
> Usando o CodeDeploy agente, você pode realizar uma implantação em uma instância na qual está conectado sem a necessidade de um aplicativo, grupo de implantação ou até mesmo uma AWS conta. Para mais informações, consulte Use o CodeDeploy agente para validar um pacote de implantação em uma máquina local.

Tópicos

- Visão geral de uma implantação no local
- Visão geral de uma blue/green implantação

Visão geral de uma implantação no local

> **Note**
>
> AWS As implantações do Lambda e do Amazon ECS não podem usar um tipo de implantação local.

Veja como funciona uma implantação no local:

1. Primeiro, você cria conteúdo implantável em sua máquina de desenvolvimento local ou ambiente similar e, em seguida, adiciona um arquivo de especificação do aplicativo (AppSpec arquivo). O AppSpec arquivo é exclusivo do CodeDeploy. Ele define as ações de implantação que você CodeDeploy deseja executar. Você agrupa o conteúdo implantável e o AppSpec arquivo em um arquivo de arquivamento e, em seguida, faz o upload para um bucket ou repositório do Amazon S3. GitHub Esse arquivo é chamado de revisão de aplicativo (ou simplesmente de revisão).
2. Em seguida, você CodeDeploy fornece informações sobre sua implantação, como de qual bucket ou GitHub repositório do Amazon S3 extrair a revisão e em qual conjunto de instâncias do Amazon EC2 implantar seu conteúdo. CodeDeploy chama um conjunto de instâncias do Amazon EC2 de um grupo de implantação. Um grupo de implantação contém instâncias do Amazon EC2 marcadas individualmente, instâncias do Amazon EC2 no Amazon EC2 Auto Scaling ou ambas.

   Cada vez que você carrega com êxito uma nova revisão de aplicativo que deseja implantar no grupo de implantação, esse pacote é definido como a revisão de destino para esse grupo de

implantação. Em outras palavras, a revisão de aplicativo atualmente direcionada para implantação é a revisão de destino. Ela também é a revisão que é extraída para implantações automáticas.

3. Em seguida, o CodeDeploy agente em cada instância faz uma pesquisa CodeDeploy para determinar o que e quando extrair do bucket GitHub ou repositório especificado do Amazon S3.
4. Por fim, o CodeDeploy agente em cada instância extrai a revisão de destino do bucket GitHub ou repositório do Amazon S3 e, usando as instruções no arquivo, implanta AppSpec o conteúdo na instância.

CodeDeploy mantém um registro de suas implantações para que você possa obter o status da implantação, os parâmetros de configuração da implantação, a integridade da instância e assim por diante.

Visão geral de uma blue/green implantação

Uma blue/green implantação é usada para atualizar seus aplicativos e, ao mesmo tempo, minimizar as interrupções causadas pelas alterações de uma nova versão do aplicativo. CodeDeploy provisiona a nova versão do aplicativo junto com a versão antiga antes de redirecionar o tráfego de produção.

- AWS Lambda: o tráfego é transferido de uma versão de uma função Lambda para uma nova versão da mesma função Lambda.
- Amazon ECS: o tráfego é passado de um conjunto de tarefas em seu serviço do Amazon ECS para um conjunto de tarefas de substituição atualizado no mesmo serviço do Amazon ECS.
- EC2/On-Premises: O tráfego é passado de um conjunto de instâncias no ambiente original para um conjunto de instâncias de substituição.

Todas as AWS implantações do Lambda e do Amazon ECS são. blue/green Uma EC2/On-Premises implantação pode ser no local ou blue/green. Uma blue/green implantação oferece várias vantagens em relação a uma implantação local:

- Você pode instalar e testar um aplicativo no novo ambiente de substituição e implantá-lo na produção simplesmente redirecionando o tráfego.
- Se você estiver usando a plataforma de EC2/On-Premises computação, voltar para a versão mais recente de um aplicativo é mais rápido e confiável. Isso ocorre porque o tráfego pode ser direcionado de volta para as instâncias originais, desde que elas não tenham sido encerradas.

Com uma implantação no local, as versões devem ser revertidas com a reimplantação da versão anterior do aplicativo.

- Se você estiver usando a plataforma de EC2/On-Premises computação, novas instâncias são provisionadas para uma blue/green implantação e refletem as configurações de servidor mais atualizadas. Isso ajuda a evitar os tipos de problemas que ocorrem algumas vezes em instâncias de execução prolongada.
- Se você estiver usando a plataforma de computação AWS Lambda, você controla como o tráfego é transferido da versão original da função Lambda para a nova versão AWS da função Lambda. AWS
- Caso esteja usando a plataforma de computação do Amazon ECS, você controla como o tráfego é deslocado do conjunto de tarefas original para o novo conjunto de tarefas.

Uma blue/green implantação com CloudFormation pode usar um dos seguintes métodos:

- CloudFormation modelos para implantações: quando você configura implantações com CloudFormation modelos, suas implantações são acionadas por atualizações. CloudFormation Quando você altera um recurso e carrega uma alteração de modelo, uma atualização de pilha CloudFormation inicia a nova implantação. Para obter uma lista dos recursos que você pode usar em CloudFormation modelos, consulteCloudFormation modelos para CodeDeploy referência.
- Blue/green implantações por meio de CloudFormation: você pode usar CloudFormation para gerenciar suas blue/green implantações por meio de atualizações de pilha. Defina os recursos azuis e verdes, além de especificar as configurações de roteamento e estabilização de tráfego, dentro do modelo de pilha. Em seguida, se você atualizar os recursos selecionados durante uma atualização da pilha, CloudFormation gera todos os recursos verdes necessários, desloca o tráfego com base nos parâmetros de roteamento de tráfego especificados e exclui os recursos azuis. Para obter mais informações, consulte Automatizar as blue/green implantações do Amazon ECS CodeDeploy usando o Guia do CloudFormationAWS CloudFormation usuário.

> **Note**
>
> Compatível somente com blue/green implantações do Amazon ECS.

A forma como você configura uma blue/green implantação depende da plataforma computacional que sua implantação está usando.

Blue/Green implantação em um AWS Lambda ou plataforma de computação Amazon ECS

Se você estiver usando a plataforma de computação AWS Lambda ou o Amazon ECS, deverá indicar como o tráfego é transferido da função original AWS Lambda ou do conjunto de tarefas do Amazon ECS para a nova função ou conjunto de tarefas. Para indicar como o tráfego é deslocado, você deve especificar uma das seguintes configurações de implantação:

- canário
- linear
- tudo de uma vez

Para obter informações sobre como o tráfego é deslocado em uma configuração de implantação canária, linear ou tudo de uma vez, consulte Configuração de implantação.

Para obter detalhes sobre a configuração de implantação do Lambda, consulte Configurações de implantação em um AWS Lambda Plataforma de computação.

Para obter detalhes sobre a configuração de implantação do Amazon ECS, consulte Configurações de implantação em uma plataforma de computação do Amazon ECS.

Blue/Green implantação em uma plataforma EC2/on-premises computacional

> **Note**
>
> Você deve usar instâncias do Amazon EC2 para blue/green implantações na plataforma computacional. EC2/On-Premises On-premises as instâncias não são compatíveis com o tipo de blue/green implantação.

Se você estiver usando a plataforma de EC2/On-Premises computação, o seguinte se aplica:

É necessário ter uma ou mais instâncias do Amazon EC2 com etiquetas do Amazon EC2 ou um grupo do Amazon EC2 Auto Scaling. As instâncias devem atender a estes requisitos adicionais:

- Cada instância do Amazon EC2 deve ter o perfil de instância do IAM correto anexado.
- O CodeDeploy agente deve estar instalado e executado em cada instância.

> **Note**
>
> Normalmente, você também tem uma revisão do aplicativo em execução nas instâncias do seu ambiente original, mas isso não é um requisito para uma blue/green implantação.

Ao criar um grupo de implantação que é usado em blue/green implantações, você pode escolher como seu ambiente substituto é especificado:

Copie um grupo existente do Amazon EC2 Auto Scaling: blue/green durante a CodeDeploy implantação, cria as instâncias para seu ambiente substituto durante a implantação. Com essa opção, CodeDeploy usa o grupo Amazon EC2 Auto Scaling que você especifica como modelo para o ambiente de substituição, incluindo o mesmo número de instâncias em execução e muitas outras opções de configuração.

Escolher instâncias manualmente: você pode especificar as instâncias a serem contadas como substituição usando tags de instância do Amazon EC2, nome de grupos do Amazon EC2 Auto Scaling ou ambos. Se você escolher essa opção, não precisará especificar as instâncias para o ambiente de substituição até criar uma implantação.

Como isso funciona:

1. Você já tem instâncias ou um grupo do Amazon EC2 Auto Scaling que serve como seu ambiente original. Na primeira vez que você executa uma blue/green implantação, normalmente usa instâncias que já foram usadas em uma implantação local.
2. Em um CodeDeploy aplicativo existente, você cria um grupo de blue/green implantação em que, além das opções necessárias para uma implantação local, você especifica o seguinte:
   - O balanceador de carga ou balanceadores de carga que roteiam o tráfego do ambiente original para o ambiente substituto durante o processo de blue/green implantação.
   - Se deseja redirecionar o tráfego para o ambiente de substituição imediatamente ou esperar e redirecioná-lo manualmente.
   - A taxa em que o tráfego é direcionado para as instâncias de substituição.
   - Se as instâncias substituídas devem ser encerradas ou continuar em execução.
3. Você cria uma implantação para esse grupo de implementação e, durante o processo, acontece o seguinte:
   a. Se você optou por copiar um grupo de Amazon EC2 Auto Scaling , as instâncias serão provisionadas para o seu ambiente de substituição.

b. A revisão de aplicativo que você especificou para a implantação será instalada nas instâncias de substituição.

c. Se você especificou um tempo de espera nas configurações do grupo de implantação, a implantação será pausada. Este é o momento em que é possível executar testes e verificações no ambiente de substituição. Se você não redirecionar manualmente o tráfego antes do final do período de espera, a implantação será interrompida.

d. As instâncias no ambiente de substituição serão registradas em um balanceador de carga do Elastic Load Balancing, e o tráfego começará a ser redirecionado para elas.

e. As instâncias no ambiente original terão o registro cancelado e serão manipuladas de acordo com as suas especificações no grupo de implantação, ou seja, serão encerradas ou continuarão em execução.

Blue/Green implantação por meio de CloudFormation

Você pode gerenciar CodeDeploy blue/green implantações modelando seus recursos com um CloudFormation modelo.

Ao modelar seus blue/green recursos usando um CloudFormation modelo, você cria uma atualização de pilha CloudFormation que atualiza seu conjunto de tarefas. O tráfego de produção é deslocado do conjunto de tarefas original do serviço para um conjunto de tarefas de substituição de uma só vez, com implantações lineares e tempos de incorporação ou com implantações canário. A atualização de pilha inicia uma implantação no CodeDeploy. Você pode visualizar o status e o histórico de implantação no modelo CodeDeploy, mas não cria nem gerencia CodeDeploy recursos fora do CloudFormation modelo.

> **Note**
>
> Para blue/green implantações concluídas CloudFormation, você não cria um CodeDeploy aplicativo ou grupo de implantação.

Esse método é compatível somente com blue/green implantações do Amazon ECS. Para obter mais informações sobre blue/green implantações por meio de CloudFormation, consulteCrie uma blue/green implantação do Amazon ECS por meio de CloudFormation.

Deixe seu comentário

Os seus comentários são bem-vindos. Para entrar em contato conosco, visite o CodeDeploy fórum.

Tópicos

- Primary Components
- Deployments
- Application Specification Files

CodeDeploy componentes primários

Antes de começar a trabalhar com o serviço, você deve se familiarizar com os principais componentes do processo de CodeDeploy implantação.

Tópicos

- Aplicação
- Plataforma de computação
- Configuração de implantação
- Grupo de implantação
- Tipo de implantação
- Perfil de instância do IAM
- Revisão
- Perfil de serviço
- Revisão de destino
- Outros componentes

Aplicação

Um aplicativo é um nome que identifica de forma exclusiva o aplicativo que você deseja implantar. CodeDeploy usa esse nome, que funciona como um contêiner, para garantir que a combinação correta de revisão, configuração de implantação e grupo de implantação seja referenciada durante uma implantação.

Plataforma de computação

Uma plataforma de computação é uma plataforma na qual CodeDeploy implanta um aplicativo. Existem três plataformas de computação:

- EC2/On-Premises: descreve instâncias de servidores físicos que podem ser instâncias de nuvem do Amazon EC2, servidores locais ou ambos. Os aplicativos criados usando a plataforma de EC2/On-Premises computação podem ser compostos por arquivos executáveis, arquivos de configuração, imagens e muito mais.

  As implantações que usam a plataforma de EC2/On-Premises computação gerenciam a forma como o tráfego é direcionado às instâncias usando um tipo local ou blue/green de implantação. Para obter mais informações, consulte Visão geral dos tipos de CodeDeploy implantação.
- AWS Lambda: usado para implantar aplicativos que consistem em uma versão atualizada de uma função Lambda. AWS Lambda gerencia a função Lambda em um ambiente computacional sem servidor composto por uma estrutura computacional de alta disponibilidade. Toda a administração dos recursos computacionais é realizada por AWS Lambda. Para obter mais informações, consulte Computação e Aplicativos com tecnologia sem servidor. Para obter mais informações sobre AWS Lambda as funções Lambda, consulte. AWS Lambda

  Você pode gerenciar a maneira como o tráfego é desviado para as versões da função do Lambda atualizadas durante uma implantação, escolhendo uma configuração de canário, linear ou todas de uma só vez.
- Amazon ECS: usado para implantar um aplicativo em contêineres do Amazon ECS como um conjunto de tarefas. CodeDeploy executa uma blue/green implantação instalando uma versão atualizada do aplicativo como um novo conjunto de tarefas de substituição. CodeDeploy redireciona o tráfego de produção do conjunto de tarefas original do aplicativo para o conjunto de tarefas de substituição. O conjunto de tarefas original é encerrado após uma implantação bem-sucedida. Para obter mais informações sobre o Amazon ECS, consulte Amazon Elastic Container Service.

  Você pode gerenciar a maneira como o tráfego é deslocado para o conjunto de tarefas atualizado durante uma implantação escolhendo uma configuração canário, linear ou todas de uma só vez.

> **Note**
>
> As blue/green implantações do Amazon ECS são suportadas por meio de e. CodeDeploy CloudFormation Os detalhes dessas implantações são descritos nas seções subsequentes.

Configuração de implantação

Uma configuração de implantação é um conjunto de regras de implantação e condições de sucesso e falha de implantação usadas CodeDeploy durante uma implantação. Se sua implantação usa a plataforma de EC2/On-Premises computação, você pode especificar o número mínimo de instâncias íntegras para a implantação. Se sua implantação usa a plataforma de computação Amazon ECS AWS Lambda ou a plataforma de computação Amazon ECS, você pode especificar como o tráfego é roteado para sua função Lambda atualizada ou conjunto de tarefas do ECS.

Para obter mais informações sobre como especificar o número mínimo de hosts íntegros para uma implantação que usa a plataforma de EC2/On-Premises computação, consulte. Sobre o número mínimo de instâncias íntegras

As configurações de implantação a seguir especificam como o tráfego é roteado durante uma implantação que usa a plataforma de computação Lambda ou ECS:

- Canário: o tráfego é deslocado em dois incrementos. Você pode escolher entre as opções canário predefinidas que especificam a porcentagem de tráfego deslocado para a sua função do Lambda atualizada ou o conjunto de tarefas do ECS no primeiro incremento e o intervalo, em minutos, antes que o tráfego restante seja deslocado no segundo incremento.
- Linear: o tráfego é deslocado em incrementos iguais com um número igual de minutos entre cada incremento. Você pode escolher entre opções lineares predefinidas que especificam a porcentagem de tráfego deslocado em cada incremento e o número de minutos entre cada incremento.
- All-at-once: todo o tráfego é transferido da função Lambda original ou do conjunto de tarefas do ECS para a função ou conjunto de tarefas atualizado de uma só vez.

Grupo de implantação

Um grupo de implantação é um conjunto de instâncias individuais. Um grupo de implantação contém instâncias do Amazon EC2; individualmente marcadas e/ou instâncias do Amazon EC2; em grupos

do Amazon EC2 Auto Scaling. Para obter informações sobre tags de instâncias do Amazon EC2, consulte Trabalhando com tags usando o console. Para obter informações sobre instâncias on-premises, consulte Working with On-Premises Instances. Para obter informações sobre o Amazon EC2 Auto Scaling, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

Tipo de implantação

Um tipo de implantação é um método usado para disponibilizar a revisão mais recente do aplicativo nas instâncias em um grupo de implantação. Existem dois tipos de implantação:

- In-place implantação: o aplicativo em cada instância do grupo de implantação é interrompido, a revisão mais recente do aplicativo é instalada e a nova versão do aplicativo é iniciada e validada. Você pode usar um balanceador de carga de forma que cada registro de instância é cancelado durante sua implantação e, em seguida, restaurado para o serviço após a conclusão da implantação. Somente implantações que usam a plataforma de EC2/On-Premises computação podem usar implantações no local. Para obter mais informações sobre implantações no local, consulte Visão geral de uma implantação no local.
- Blue/green implantação: o comportamento da sua implantação depende da plataforma de computação que você usa:
  - Blue/green em uma plataforma de EC2/On-Premises computação: as instâncias em um grupo de implantação (o ambiente original) são substituídas por um conjunto diferente de instâncias (o ambiente substituto) usando estas etapas:
    - As instâncias são provisionadas para o ambiente de substituição.
    - A revisão de aplicativo mais recente será instalada nas instâncias de substituição.
    - Um tempo de espera opcional ocorre para atividades como teste de aplicativos e verificação do sistema.
    - As instâncias no ambiente de substituição são registradas com um ou mais balanceadores de carga do Elastic Load Balancing, fazendo com que o tráfego seja redirecionado para elas. O registro das instâncias no ambiente original é cancelado e pode ser encerrado ou mantido em execução para outros usos.

> **Note**
>
> Se você usa uma plataforma EC2/On-Premises computacional, saiba que as blue/green implantações funcionam somente com instâncias do Amazon EC2.

- Blue/green em uma plataforma computacional AWS Lambda ou Amazon ECS: o tráfego é deslocado em incrementos de acordo com uma configuração de implantação canária, linear ou completa.
- Blue/green implantações por meio de CloudFormation: o tráfego é transferido de seus recursos atuais para seus recursos atualizados como parte de uma atualização de CloudFormation pilha. Atualmente, somente blue/green implantações do ECS são suportadas.

Para obter mais informações sobre blue/green implantações, consulteVisão geral de uma blue/green implantação.

> **Note**
>
> As blue/green implantações do Amazon ECS são suportadas usando e. CodeDeploy CloudFormation Os detalhes dessas implantações são descritos nas seções subsequentes.

Perfil de instância do IAM

Um perfil de instância do IAM é um perfil do IAM que você anexa às suas instâncias do Amazon EC2. Esse perfil inclui as permissões necessárias para acessar os buckets ou GitHub repositórios do Amazon S3 onde os aplicativos são armazenados. Para obter mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.

Revisão

Uma revisão é uma versão do seu aplicativo. Uma revisão de implantação do AWS Lambda é um JSON-formatted arquivo ou YAML que especifica informações sobre a função Lambda a ser implantada. Uma revisão de EC2/On-Premises implantação é um arquivo que contém conteúdo de origem (código-fonte, páginas da Web, arquivos executáveis e scripts de implantação) e um arquivo de especificação do aplicativo (AppSpec arquivo). AWS As revisões do Lambda podem ser armazenadas em buckets do Amazon S3. EC2/On-Premises as revisões são armazenadas em GitHub buckets ou repositórios do Amazon S3. Para o Amazon S3, uma revisão é identificada de maneira exclusiva por sua uma chave de objeto do Amazon S3 e sua ETag, versão ou ambos. Pois GitHub, uma revisão é identificada exclusivamente por seu ID de confirmação.

Perfil de serviço

Uma função de serviço é uma função do IAM que concede permissões a um AWS serviço para que ele possa acessar AWS recursos. As políticas que você anexa à função de serviço determinam quais AWS recursos o serviço pode acessar e as ações que ele pode realizar com esses recursos. Para CodeDeploy, uma função de serviço é usada para o seguinte:

- Para ler as tags aplicadas às instâncias ou aos nomes de grupo do Amazon EC2 Auto Scaling associados a essas instâncias. Isso permite CodeDeploy identificar instâncias nas quais ele pode implantar aplicativos.
- Para realizar operações em instâncias, grupos do Amazon EC2 Auto Scaling e balanceadores de carga do Elastic Load Balancing.
- Para publicar informações nos tópicos do Amazon SNS para que as notificações possam ser enviadas quando ocorrerem eventos específicos de implantação ou instância.
- Para recuperar informações sobre CloudWatch alarmes para configurar o monitoramento de alarmes para implantações.

Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Revisão de destino

Uma revisão de destino é a versão mais recente da revisão do aplicativo que você carregou no seu repositório e deseja implantar nas instâncias em um grupo de implantação. Em outras palavras, a revisão de aplicação que é o destino atual da implantação. Ela também é a revisão que é extraída para implantações automáticas.

Outros componentes

Para obter informações sobre outros componentes no CodeDeploy fluxo de trabalho, consulte os tópicos a seguir:

- Escolha um tipo de CodeDeploy repositório
- Deployments
- Application Specification Files
- Instance Health
- Trabalhando com o CodeDeploy agente

- Working with On-Premises Instances

CodeDeploy implantações

Este tópico fornece informações sobre os componentes e o fluxo de trabalho de implantações no CodeDeploy. O processo de implantação varia, dependendo da plataforma computacional ou do método de implantação (Lambda, Amazon ECS EC2/On-Premises ou AWS CloudFormation por meio) que você usa para suas implantações.

Tópicos

- Implantações em um AWS Plataforma de computação Lambda
- Implantação azul/verde em uma plataforma de computação do Amazon ECS
- Implantações em uma plataforma de EC2/On-Premises computação

Implantações em um AWS Plataforma de computação Lambda

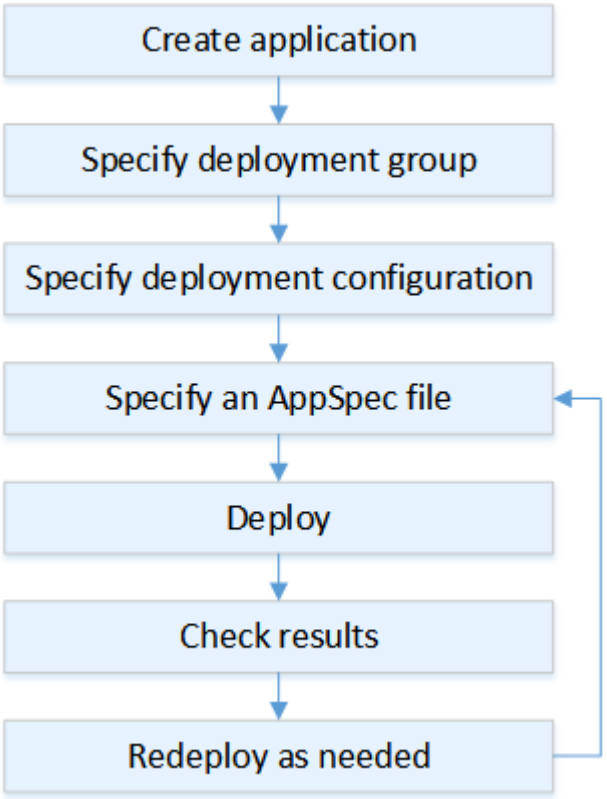
Este tópico fornece informações sobre os componentes e o fluxo de trabalho das CodeDeploy implantações que usam a plataforma de computação AWS Lambda.

Tópicos

- Fluxo de trabalho de implantação em um AWS Lambda Plataforma de computação
- Carregando sua revisão de aplicativo
- Criando seus grupos de implantação e aplicativos
- Implantando sua revisão de aplicativo
- Atualizar o aplicativo do
- Implantações interrompidas e com falha
- Reimplantações e reversões de implantação

Fluxo de trabalho de implantação em um AWS Lambda Plataforma de computação

O diagrama a seguir mostra as principais etapas na implantação de funções do AWS Lambda novas e atualizadas.



Essas etapas incluem:

1. Criar um aplicativo e atribuir a ele um nome que identifique exclusivamente as revisões de aplicativo que você deseja implantar. Para implantar funções Lambda, escolha a plataforma de computação AWS Lambda ao criar seu aplicativo. CodeDeploy usa esse nome durante uma implantação para garantir que esteja referenciando os componentes de implantação corretos, como o grupo de implantação, a configuração de implantação e a revisão do aplicativo. Para obter mais informações, consulte Crie um aplicativo com CodeDeploy.
2. Configurar um grupo de implantação especificando o nome de seu grupo de implantação.
3. Escolha uma configuração de implantação para especificar como o tráfego é transferido da versão original da AWS Lambda função para a nova versão da função Lambda. Para obter mais informações, consulte View Deployment Configuration Details.
4. Fazer upload de um arquivo de especificação do aplicativo (AppSpec arquivo) para o Amazon S3. O AppSpec arquivo especifica uma versão da função Lambda e as funções Lambda usadas para validar sua implantação. Se não quiser criar um AppSpec arquivo, você pode especificar uma versão da função Lambda e as funções de validação de implantação do Lambda diretamente

no console usando YAML ou JSON. Para obter mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

5. Implante sua revisão do aplicativo no grupo de implantação. AWS CodeDeploy implanta a revisão da função Lambda que você especificou. O tráfego é transferido para a revisão da função Lambda usando o arquivo de AppSpec implantação que você escolheu ao criar seu aplicativo. Para obter mais informações, consulte Crie uma implantação com CodeDeploy.
6. Verificar os resultados da implantação. Para obter mais informações, consulte Monitorando implantações em CodeDeploy.

Carregando sua revisão de aplicativo

Coloque um AppSpec arquivo no Amazon S3 ou insira-o diretamente no console ou. AWS CLI Para obter mais informações, consulte Application Specification Files.

Criando seus grupos de implantação e aplicativos

Um grupo CodeDeploy de implantação em uma plataforma de computação AWS Lambda identifica uma coleção de um ou mais arquivos. AppSpec Cada AppSpec arquivo pode implantar uma versão da função Lambda. Um grupo de implantação também define um conjunto de opções de configuração para futuras implantações, como configurações de alarmes e reversões.

Implantando sua revisão de aplicativo

Agora você está pronto para implantar a revisão da função especificada no AppSpec arquivo no grupo de implantação. Você pode usar o CodeDeploy console ou o comando create-deployment. Existem parâmetros que você pode especificar para controlar sua implantação, incluindo a revisão, o grupo de implantação e a configuração de implantação.

Atualizar o aplicativo do

Você pode fazer atualizações em seu aplicativo e, em seguida, usar o CodeDeploy console ou chamar o comando create-deployment para enviar uma revisão.

Implantações interrompidas e com falha

Você pode usar o CodeDeploy console ou o comando stop-deployment para interromper uma implantação. Quando você tenta interromper a implantação, uma de três coisas acontece:

- A implantação é interrompida, e a operação retorna a um status de sucesso. Nesse caso, mais nenhum evento de ciclo de vida de implantação é executado no grupo de implantação para a implantação interrompida.
- A implantação não é interrompida imediatamente, e a operação retorna um status pendente. Nesse caso, alguns eventos de ciclo de vida de implantação ainda podem estar em execução no grupo de implantação. Após a conclusão da operação pendente, as chamadas subsequentes para interromper a implantação retornam um status de sucesso.
- A implantação não pode ser interrompida, e a operação retorna um erro. Para obter mais informações, consulte ErrorInformationErros comuns na Referência AWS CodeDeploy da API.

Como as implantações interrompidas, as implantações com falha podem resultar em alguns eventos de ciclo de vida da implantação que já foram executados. Para descobrir por que uma implantação falhou, você pode usar o console do CodeDeploy ou analisar os dados do arquivo de log da implantação com falha. Para obter mais informações, consulte Limpeza de revisões de aplicativo e arquivos de log e Exibir dados de registro para CodeDeploy EC2/On-Premises implantações.

Reimplantações e reversões de implantação

CodeDeploy implementa reversões reimplantando, como uma nova implantação, uma revisão implantada anteriormente.

Você pode configurar um grupo de implantação para reverter automaticamente as implantações quando determinadas condições forem atendidas, inclusive quando uma implantação falhar ou um limite de monitoramento de alarme for atingido. Você também pode substituir configurações de reversão especificadas para um grupo de implantação em uma implantação individual.

Além disso, pode optar por reverter uma implantação com falha, reimplantando manualmente uma revisão anteriormente implantada.

Em todos os casos, a implantação nova ou revertida tem um ID de implantação próprio. A lista de implantações que você pode ver no CodeDeploy console mostra quais são o resultado de uma implantação automática.

Para obter mais informações, consulte Reimplante e reverta uma implantação com CodeDeploy.

Implantação azul/verde em uma plataforma de computação do Amazon ECS

Este tópico fornece informações sobre os componentes e o fluxo de trabalho das CodeDeploy implantações que usam a plataforma computacional Amazon ECS.

Tópicos

- Antes de começar uma implantação do
- Fluxo de trabalho da implantação (alto nível) em uma plataforma de computação do Amazon ECS
- O que acontece durante uma implantação do
- Carregando sua revisão de aplicativo
- Criando seus grupos de implantação e aplicativos
- Implantando sua revisão de aplicativo
- Atualizar o aplicativo
- Implantações interrompidas e com falha
- Reimplantações e reversões de implantação
- blue/green Implantações do Amazon ECS por meio de AWS CloudFormation

Antes de começar uma implantação do

Antes de iniciar uma implantação de aplicativo do Amazon ECS, você deve ter o seguinte preparado. Alguns requisitos são especificados quando você cria seu grupo de implantação e alguns são especificados no AppSpec arquivo.

Requisito	Onde especificado
Cluster do Amazon ECS	Grupo de implantação
Serviço do Amazon ECS	Grupo de implantação
Application Load Balancer e Network Load Balancer	Grupo de implantação
Listener de produção	Grupo de implantação

Requisito	Onde especificado
Listener de teste (opcional)	Grupo de implantação
Dois grupos de destino	Grupo de implantação
Definição de tarefa do Amazon ECS	AppSpec arquivo
Nome do contêiner	AppSpec arquivo
Porta do contêiner	AppSpec arquivo

Cluster do Amazon ECS

Um cluster do Amazon ECS é um agrupamento lógico de tarefas ou serviços. Você especifica o cluster do Amazon ECS que contém seu serviço Amazon ECS ao criar o grupo de implantação do seu CodeDeploy aplicativo. Para obter mais informações, consulte clusters do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.

Serviço do Amazon ECS

Um serviço do Amazon ECS ajuda você a executar e manter simultaneamente um número especificado de instâncias de uma definição de tarefa em um cluster do Amazon ECS. Seu serviço Amazon ECS deve estar habilitado para CodeDeploy. Por padrão, um serviço do Amazon ECS é habilitado para implantações do Amazon ECS . Ao criar o grupo de implantação, você pode optar por implantar um serviço do Amazon ECS que está no seu cluster do Amazon ECS. Para obter mais informações, consulte no Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.

Application Load Balancer e Network Load Balancer

Você deve usar o Elastic Load Balancing com o serviço Amazon ECS que você deseja atualizar com uma implantação do Amazon ECS. É possível utilizar um Application Load Balancer ou Network Load Balancer. Recomendamos um Application Load Balancer para que você possa aproveitar atributos, como o mapeamento de porta dinâmico e o roteamento baseado em caminhos e regras de prioridade. Você especifica o balanceador de carga ao criar o grupo de implantação do seu CodeDeploy aplicativo. Para obter mais informações, consulte Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS e Criar um cluster no balanceador de carga no Guia do desenvolvedor do Amazon Elastic Container Service.

Um ou dois receptores

Um listener é usado pelo seu load balancer para direcionar o tráfego para seus grupos de destino. Um listener de produção é necessário. Você pode especificar um segundo listener de teste opcional que direciona o tráfego para o seu conjunto de tarefas de substituição enquanto você executa testes de validação. Especifique um ou dois listeners ao criar seu grupo de implantação. Se você usar o console do Amazon ECS para criar o serviço do Amazon ECS, seus receptores são criados para você. Para obter mais informações, consulte Receptores para seus Application Load Balancers no Guia do usuário do Elastic Load Balancing e Criar um serviço no Guia do usuário do Amazon Elastic Container Service.

Dois grupos-alvo do Amazon ECS

Um grupo de destino é usado para rotear o tráfego para um destino registrado. Uma implantação do Amazon ECS requer dois grupos de destino: um para o conjunto de tarefas original do aplicativo do Amazon ECS e um para seu conjunto de tarefas de substituição. Durante a implantação, CodeDeploy cria um conjunto de tarefas de substituição e redireciona o tráfego do conjunto de tarefas original para o novo. Especifique os grupos de destino ao criar o grupo de implantação do aplicativo do CodeDeploy .

Durante uma implantação, CodeDeploy determina qual grupo-alvo está associado ao conjunto de tarefas em seu serviço Amazon ECS que tem o status `PRIMARY` (esse é o conjunto de tarefas original) e associa um grupo-alvo a ele e, em seguida, associa o outro grupo-alvo ao conjunto de tarefas de substituição. Se você executar outra implantação, o grupo de destino associado ao conjunto de tarefas original da implantação atual será associado ao conjunto de tarefas de substituição da próxima implantação. Para obter mais informações, consulte Grupos de destino para seus Application Load Balancer no Guia do usuário do Elastic Load Balancing.

Definição de tarefa do Amazon ECS

Uma definição de tarefa é necessária para executar o contêiner do Docker que contém o aplicativo do Amazon ECS. Você especifica o ARN da definição da tarefa no arquivo do CodeDeploy AppSpec aplicativo. Para obter mais informações, consulte Definições de tarefas do Amazon ECS no Guia do desenvolvedor do Serviço Amazon Elastic Container e AppSpec seção 'recursos' para implantações do Amazon ECS.

Contêiner para seu aplicativo do Amazon ECS

Um contêiner do Docker é uma unidade de software que reúne o código e suas dependências para que o aplicativo possa ser executado. Um contêiner isola o aplicativo para que ele seja executado em diferentes ambientes de computação. Seu balanceador de carga direciona o

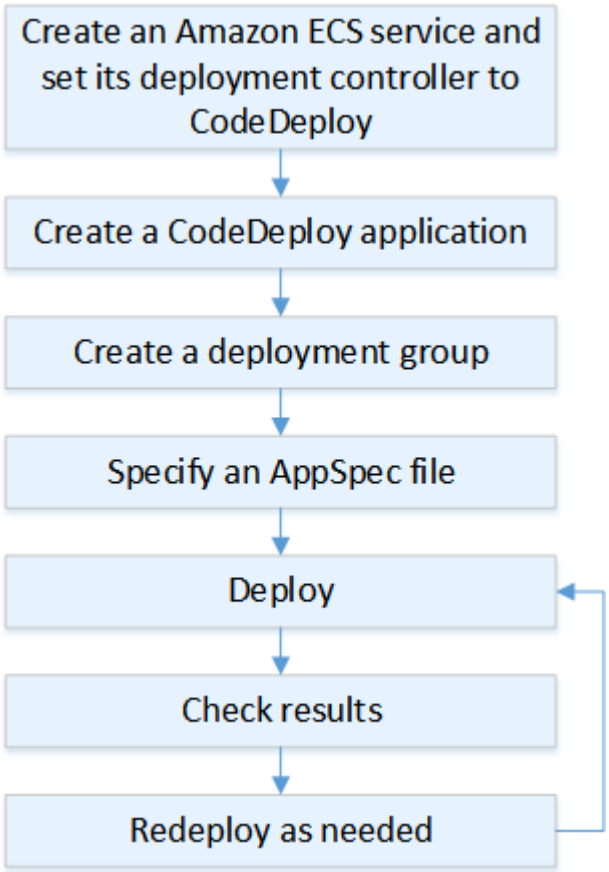
tráfego para um contêiner no conjunto de tarefas do aplicativo do Amazon ECS. Você especifica o nome do seu contêiner no AppSpec arquivo do seu CodeDeploy aplicativo. O contêiner especificado em seu AppSpec arquivo deve ser um dos contêineres especificados na definição de tarefas do Amazon ECS. Para obter mais informações, consulte O que é o Amazon Elastic Container Service? no Guia do usuário do Amazon Elastic Container Service e AppSpec seção 'recursos' para implantações do Amazon ECS.

Uma porta para o conjunto de tarefas de substituição

Durante a implantação do Amazon ECS, seu balanceador de carga direciona o tráfego para essa porta no contêiner especificado no arquivo do seu CodeDeploy aplicativo. AppSpec Você especifica a porta no AppSpec arquivo do seu CodeDeploy aplicativo. Para obter mais informações, consulte AppSpec seção 'recursos' para implantações do Amazon ECS.

Fluxo de trabalho da implantação (alto nível) em uma plataforma de computação do Amazon ECS

O diagrama a seguir mostra as principais etapas na implantação de serviços do Amazon ECS atualizados.



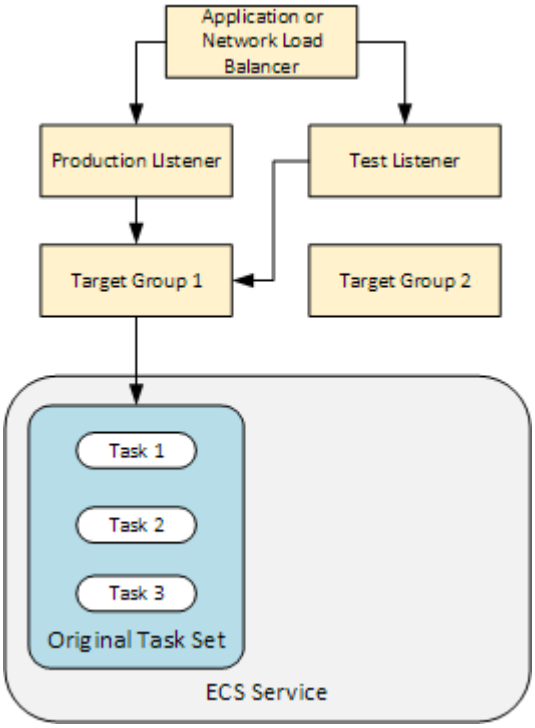
Essas etapas incluem:

1. Crie um AWS CodeDeploy aplicativo especificando um nome que represente de forma exclusiva o que você deseja implantar. Para implantar um aplicativo Amazon ECS, em seu AWS CodeDeploy aplicativo, escolha a plataforma computacional Amazon ECS. CodeDeploy usa um aplicativo durante uma implantação para referenciar os componentes de implantação corretos, como o grupo de implantação, grupos-alvo, ouvintes, comportamento de redirecionamento de tráfego e revisão do aplicativo. Para obter mais informações, consulte Crie um aplicativo com CodeDeploy.
2. Configurar um grupo de implantação especificando:
   - O nome do grupo de implantação.
   - Cluster e serviço do Amazon ECS O controlador de implantação do serviço Amazon ECS deve estar configurado como. CodeDeploy
   - O listener de produção, um listener de teste opcional e grupos de destino usados durante uma implantação.
   - Configurações de implantação, como quando redirecionar o tráfego de produção para o conjunto de tarefas de substituição do Amazon ECS em seu serviço do Amazon ECS e quando encerrar o conjunto de tarefas original do Amazon ECS definido em seu serviço do Amazon ECS.
   - Configurações opcionais, como triggers, alarmes e o comportamento de reversão.
3. Especifique um arquivo de especificação do aplicativo (AppSpec arquivo). Você pode enviá-lo para o Amazon S3, inseri-lo no console no formato YAML ou JSON ou especificá-lo com o ou SDK. AWS CLI O AppSpec arquivo especifica uma definição de tarefa do Amazon ECS para a implantação, um nome de contêiner e mapeamento de portas usados para rotear o tráfego, além de funções Lambda executadas após os ganchos do ciclo de vida da implantação. O nome do contêiner deve ser um contêiner em sua definição de tarefa do . Para obter mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.
4. Implante a revisão do seu aplicativo. AWS CodeDeploy redireciona o tráfego da versão original de um conjunto de tarefas em seu serviço Amazon ECS para um novo conjunto de tarefas substituto. Grupos de destino especificados no grupo de implantação são usados para fornecer o tráfego para conjuntos de tarefa originais e de substituição. Depois de concluir a implantação, o conjunto de tarefas original é encerrado. Você pode especificar um listener de teste opcional para fornecer o tráfego de teste para a sua versão de substituição antes que o tráfego seja redirecionado a ele. Para obter mais informações, consulte Crie uma implantação com CodeDeploy.
5. Verificar os resultados da implantação. Para obter mais informações, consulte Monitorando implantações em CodeDeploy.

O que acontece durante uma implantação do

Antes de uma implantação do Amazon ECS com um receptor de teste ser iniciada, você deve configurar seus componentes. Para obter mais informações, consulte Antes de começar uma implantação do.

O diagrama a seguir mostra a relação entre esses componentes quando uma implantação do Amazon ECS está pronta para ser iniciada.



Quando a implantação é iniciada, os eventos de ciclo de vida de implantação começam a ser executados um de cada vez. Alguns eventos do ciclo de vida são ganchos que executam somente funções Lambda especificadas no arquivo. AppSpec Os eventos de ciclo de vida de implantação na tabela a seguir são listados na ordem em que são executados. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.

Evento de ciclo de vida	Ação do evento de ciclo de vida
`BeforeInstall` (um hook para funções do Lambda)	Executar função do Lambda
Instalar	Configurar o conjunto de tarefas de substituição.

Evento de ciclo de vida	Ação do evento de ciclo de vida
`AfterInstall` (um hook para funções do Lambda)	Executar função do Lambda
AllowTestTraffic	Rotear o tráfego do listener de teste para o grupo de destino 2.
`AfterAllowTestTraffic` (um hook para funções do Lambda)	Executar função do Lambda
`BeforeAllowTraffic` (um hook para funções do Lambda)	Executar função do Lambda
AllowTraffic	Rotear o tráfego do listener de produção para o grupo de destino 2.
`AfterAllowTraffic`	Executar função do Lambda

> **Note**
>
> As funções do Lambda em um hook são opcionais.

1. Execute todas as funções do Lambda especificadas no `BeforeInstall` gancho do AppSpec arquivo.
2. Durante o evento de ciclo de vida `Install`:

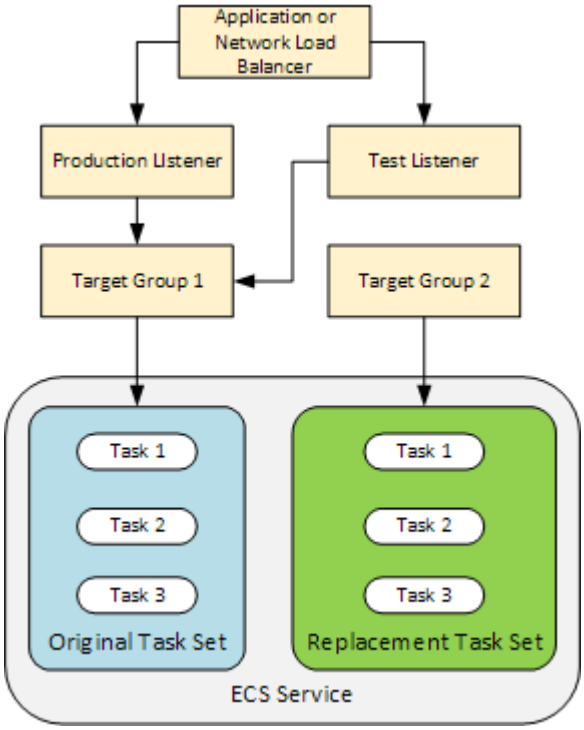
   a. Um conjunto de tarefas de substituição é criado no seu serviço .

   b. O aplicativo em contêineres atualizado é instalado no conjunto de tarefas de substituição.

   c. O segundo grupo de destino é associado ao conjunto de tarefas de substituição.

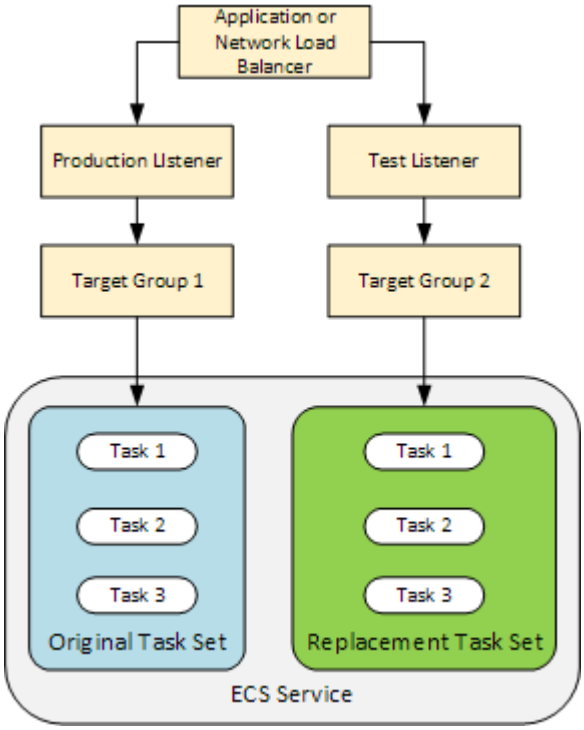
   Este diagrama mostra os componentes de implantação com o novo conjunto de tarefas de substituição. O aplicativo em contêineres está dentro desse conjunto de tarefas. O conjunto

de tarefas é composto de três tarefas. (Um aplicativo pode ter qualquer número de tarefas.) O segundo grupo de destino agora está associado ao conjunto de tarefas de substituição.



3. Execute todas as funções do Lambda especificadas no `AfterInstall` gancho do AppSpec arquivo.

4. O evento `AllowTestTraffic` é invocado. Durante esse evento de ciclo de vida, o listener de teste roteia o tráfego para o aplicativo em contêineres atualizado.



5. Execute todas as funções do Lambda especificadas no `AfterAllowTestTraffic` gancho do AppSpec arquivo. As funções do Lambda podem validar a implantação usando o tráfego de teste. Por exemplo, uma função Lambda pode fornecer o tráfego ao listener de teste e rastrear as métricas do conjunto de tarefas de substituição. Se as reversões estiverem configuradas, você poderá configurar um CloudWatch alarme que acionará uma reversão quando o teste de validação em sua função Lambda falhar.

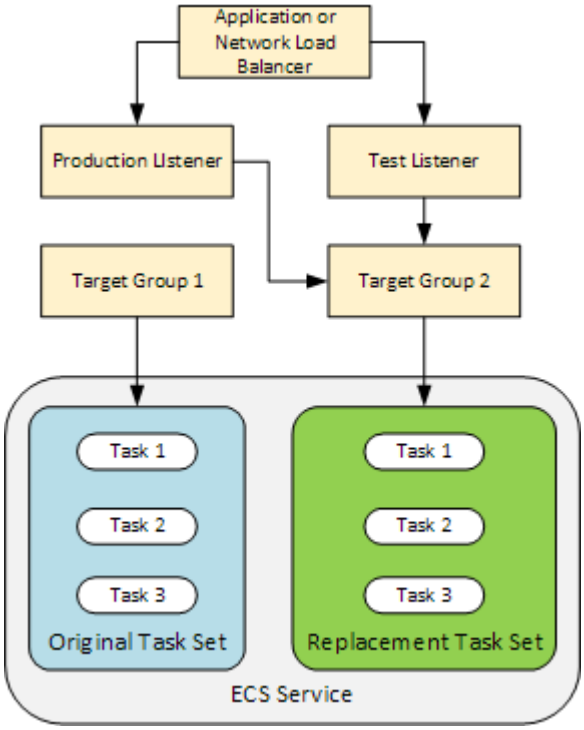
   Após a conclusão dos testes de validação, ocorre uma das seguintes situações:

   - Se a validação falhar e as reversões estiverem configuradas, o status da implantação será marcado como `Failed` e os componentes retornarão ao estado em que estavam quando a implantação começou.
   - Se a validação falhar e as reversões não estiverem configuradas, o status da implantação será marcado como `Failed` e os componentes permanecerão no estado atual.
   - Se a validação for bem-sucedida, a implantação continuará no gancho `BeforeAllowTraffic`.

   Para obter mais informações, consulte Monitorando implantações com CloudWatch alarmes em CodeDeploy, Reversões automáticas e Configurar opções avançadas para um grupo de implantação.

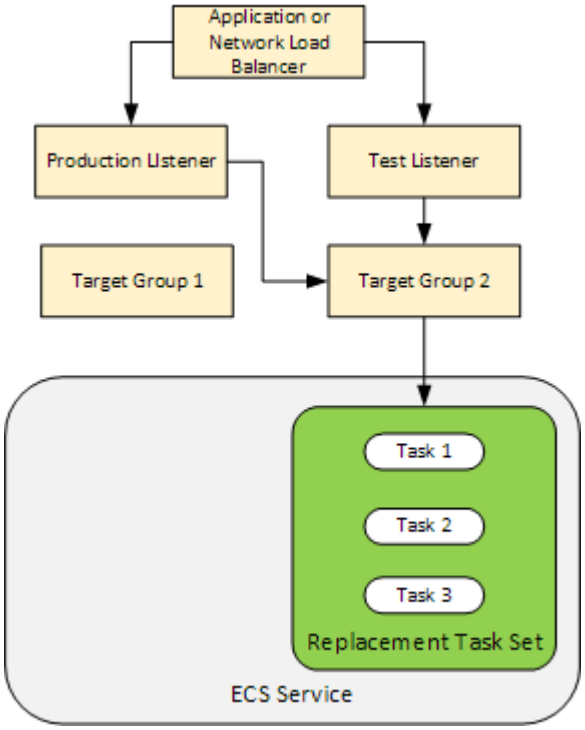
6. Execute todas as funções do Lambda especificadas no `BeforeAllowTraffic` gancho do AppSpec arquivo.

7. O evento `AllowTraffic` é invocado. O tráfego de produção é redirecionado do conjunto de tarefas original para o conjunto de tarefas de substituição. O diagrama a seguir mostra o conjunto de tarefas de substituição recebendo o tráfego de produção.



8. Execute todas as funções do Lambda especificadas no `AfterAllowTraffic` gancho do AppSpec arquivo.

9. Após a execução bem-sucedida de todos os eventos, o status da implantação é definido como `Succeeded` e o conjunto de tarefas original é removido.



Carregando sua revisão de aplicativo

Coloque um AppSpec arquivo no Amazon S3 ou insira-o diretamente no console ou. AWS CLI Para obter mais informações, consulte Application Specification Files.

Criando seus grupos de implantação e aplicativos

Um grupo de CodeDeploy implantação em uma plataforma computacional do Amazon ECS identifica ouvintes para fornecer tráfego para seu aplicativo Amazon ECS atualizado e dois grupos-alvo usados durante sua implantação. Um grupo de implantação também define um conjunto de opções de configuração, como configurações de reversões e alarmes.

Implantando sua revisão de aplicativo

Agora, você está pronto para implantar o serviço do Amazon ECS atualizado e especificado no seu grupo de implantação. Você pode usar o CodeDeploy console ou o comando create-deployment. Existem parâmetros que você pode especificar para controlar sua implantação, incluindo o grupo de implantação e a revisão.

Atualizar o aplicativo

Você pode fazer atualizações em seu aplicativo e, em seguida, usar o CodeDeploy console ou chamar o comando create-deployment para enviar uma revisão.

Implantações interrompidas e com falha

Você pode usar o CodeDeploy console ou o comando stop-deployment para interromper uma implantação. Quando você tenta interromper a implantação, uma de três coisas acontece:

- A implantação é interrompida, e a operação retorna a um status de sucesso. Nesse caso, mais nenhum evento de ciclo de vida de implantação é executado no grupo de implantação para a implantação interrompida.
- A implantação não é interrompida imediatamente, e a operação retorna um status pendente. Nesse caso, alguns eventos de ciclo de vida de implantação ainda podem estar em execução no grupo de implantação. Após a conclusão da operação pendente, as chamadas subsequentes para interromper a implantação retornam um status de sucesso.
- A implantação não pode ser interrompida, e a operação retorna um erro. Para obter mais informações, consulte Informações sobre erros e Erros comuns na Referência AWS CodeDeploy da API.

Reimplantações e reversões de implantação

CodeDeploy implementa reversões redirecionando o tráfego do conjunto de tarefas de substituição para o conjunto de tarefas original.

Você pode configurar um grupo de implantação para reverter automaticamente as implantações quando determinadas condições forem atendidas, inclusive quando uma implantação falhar ou um limite de monitoramento de alarme for atingido. Você também pode substituir configurações de reversão especificadas para um grupo de implantação em uma implantação individual.

Além disso, pode optar por reverter uma implantação com falha, reimplantando manualmente uma revisão anteriormente implantada.

Em todos os casos, a implantação nova ou revertida tem um ID de implantação próprio. O console do CodeDeploy exibe uma lista de implantações que são o resultado de uma implantação automática.

Se você implantar novamente, o grupo de destino associado ao conjunto de tarefas original da implantação atual será associado ao conjunto de tarefas de substituição da nova implantação.

Para obter mais informações, consulte Reimplante e reverta uma implantação com CodeDeploy.

blue/green Implantações do Amazon ECS por meio de AWS CloudFormation

Você pode usar AWS CloudFormation para gerenciar blue/green implantações do Amazon ECS por meio de. CodeDeploy Para obter mais informações, consulte Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation.

> **Note**
>
> O gerenciamento de blue/green implantações do Amazon ECS com não CloudFormation está disponível na região Ásia-Pacífico (Osaka).

Implantações em uma plataforma de EC2/On-Premises computação

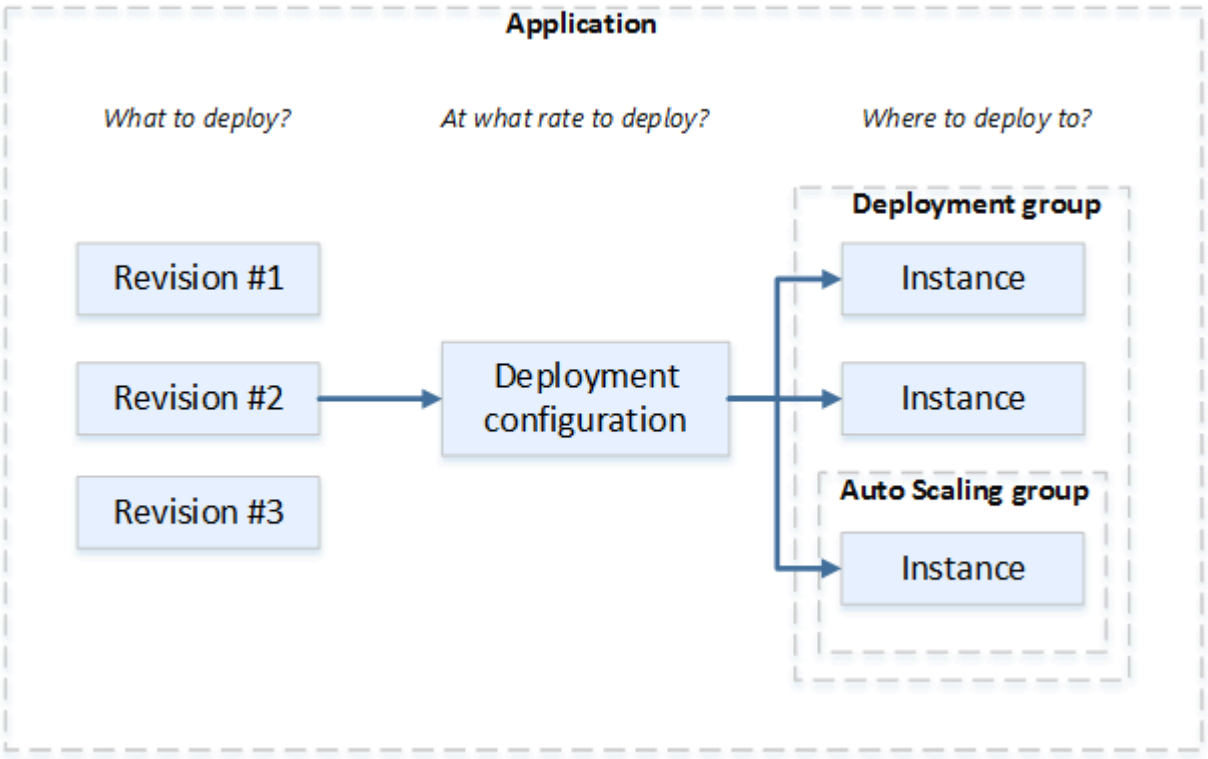
Este tópico fornece informações sobre os componentes e o fluxo de trabalho das CodeDeploy implantações que usam a plataforma de EC2/On-Premises computação. Para obter informações sobre blue/green implantações, consulteVisão geral de uma blue/green implantação.

Tópicos

- Componentes de implantação em uma EC2/on-premises plataforma computacional
- Fluxo de trabalho de implantação em uma EC2/on-premises plataforma de computação
- Configurando Instâncias
- Carregando sua revisão de aplicativo
- Criando seus grupos de implantação e aplicativos
- Implantando sua revisão de aplicativo
- Atualizar o aplicativo do
- Implantações interrompidas e com falha
- Reimplantações e reversões de implantação

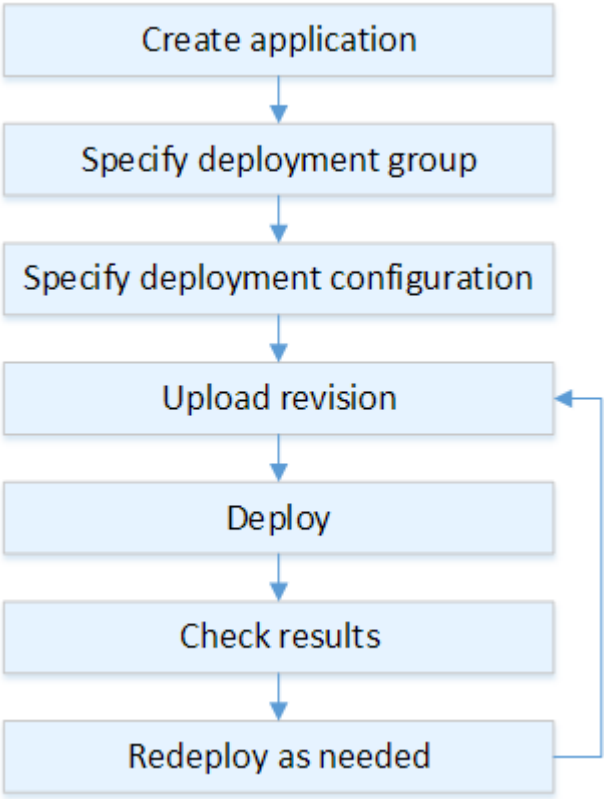
Componentes de implantação em uma EC2/on-premises plataforma computacional

O diagrama a seguir mostra os componentes em uma CodeDeploy implantação em uma plataforma EC2/On-Premises computacional.



Fluxo de trabalho de implantação em uma EC2/on-premises plataforma de computação

O diagrama a seguir mostra as principais etapas na implantação de revisões de aplicativo:



Essas etapas incluem:

1. Crie um aplicativo e dê a ele um nome que identifique de forma exclusiva as revisões do aplicativo que você deseja implantar e a plataforma de computação do seu aplicativo. CodeDeploy usa esse nome durante uma implantação para garantir que esteja referenciando os componentes de implantação corretos, como o grupo de implantação, a configuração de implantação e a revisão do aplicativo. Para obter mais informações, consulte Crie um aplicativo com CodeDeploy.
2. Configurar um grupo de implantação especificando um tipo de implantação e as instâncias nas quais você deseja implantar suas revisões de aplicativo. Uma implantação no local atualiza instâncias com a revisão de aplicativo mais recente. Uma blue/green implantação registra um conjunto substituto de instâncias para o grupo de implantação com um balanceador de carga e cancela o registro das instâncias originais.

   Você pode especificar as tags aplicadas às instâncias, os nomes dos grupos do Amazon EC2 Auto Scaling ou ambos.

   Se você especificar um grupo de tags em um grupo de CodeDeploy implantação, implanta em instâncias que tenham pelo menos uma das tags especificadas aplicada. Se você especificar dois ou mais grupos de tags, CodeDeploy implanta somente nas instâncias que atendem aos critérios

de cada um dos grupos de tags. Para obter mais informações, consulte Tagging Instances for Deployments.

Em todos os casos, as instâncias devem ser configuradas para serem usadas em uma implantação (ou seja, devem ser marcadas ou pertencer a um grupo do Amazon EC2 Auto Scaling) e CodeDeploy ter o agente instalado e em execução.

Você pode usar um CloudFormation modelo para configurar rapidamente uma instância do Amazon EC2 com base no Amazon Linux ou no Windows Server. Também fornecemos o CodeDeploy agente autônomo para que você possa instalá-lo em instâncias do Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) ou Windows Server. Para obter mais informações, consulte Crie um grupo de implantação com CodeDeploy.

Também é possível especificar as seguintes opções:

- Notificações do Amazon SNS. Crie gatilhos que enviem notificações aos assinantes de um tópico do Amazon SNS quando eventos específicos, como eventos de êxito ou falha, ocorrerem em implantações e instâncias. Para obter mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications.
- Alarm-based gerenciamento de implantação. Implemente o monitoramento de CloudWatch alarmes da Amazon para interromper as implantações quando suas métricas excederem ou ficarem abaixo dos limites definidos. CloudWatch
- Reversões de implantação automáticas. Configure uma implantação para que ela seja revertida automaticamente para a última revisão conhecida em bom estado quando uma implantação falhar ou um limite de alarme for atingido.

3. Especificar uma configuração de implantação para indicar em quantas instâncias suas revisões de aplicativo devem ser implantadas simultaneamente e descrever as condições de êxito e falha para a implantação. Para obter mais informações, consulte View Deployment Configuration Details.
4. Faça o upload de uma revisão do aplicativo para o Amazon S3 ou. GitHub Além dos arquivos que você deseja implantar e dos scripts que deseja executar durante a implantação, você deve incluir um arquivo de especificação do aplicativo (AppSpec arquivo). Esse arquivo contém instruções de implantação, como para onde copiar os arquivos em cada instância e quando executar scripts de implantação. Para obter mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.
5. Implantar a revisão de aplicativo no grupo de implantação. O CodeDeploy agente em cada instância no grupo de implantação copia a revisão do seu aplicativo do Amazon S3 ou GitHub para a instância. Em seguida, o CodeDeploy agente separa a revisão e, usando o AppSpec

arquivo, copia os arquivos nos locais especificados e executa todos os scripts de implantação. Para obter mais informações, consulte Crie uma implantação com CodeDeploy.

6. Verificar os resultados da implantação. Para obter mais informações, consulte Monitorando implantações em CodeDeploy.
7. Reimplantar uma revisão. Você pode querer fazer isso se precisar corrigir um bug no conteúdo de origem, executar os scripts de implantação em uma ordem diferente ou corrigir uma implantação com falha. Para fazer isso, reagrupe seu conteúdo de origem revisado, quaisquer scripts de implantação e o AppSpec arquivo em uma nova revisão e, em seguida, faça o upload da revisão no bucket ou repositório do Amazon S3. GitHub Depois execute uma nova implantação no mesmo grupo de implantação com a nova revisão. Para obter mais informações, consulte Crie uma implantação com CodeDeploy.

Configurando Instâncias

Você precisa configurar instâncias para poder implantar revisões de aplicativo pela primeira vez. Se uma revisão de aplicativo exigir três servidores de produção e dois servidores de backup, você executará ou usará cinco instâncias.

Para provisionar instâncias manualmente:

1. Instale o CodeDeploy agente nas instâncias. O CodeDeploy agente pode ser instalado nas instâncias Amazon Linux, Ubuntu Server, RHEL e Windows Server.
2. Ative a marcação, se você estiver usando tags para identificar instâncias em um grupo de implantação. CodeDeploy depende de tags para identificar e agrupar instâncias em grupos CodeDeploy de implantação. Embora os tutoriais de introdução usam ambas, você pode simplesmente usar uma chave ou um valor para definir uma tag para um grupo de implantação.
3. Inicie instâncias do EC2; com um perfil de instância do IAM anexado. O perfil da instância do IAM deve ser anexado a uma instância do Amazon EC2 à medida que é executado para que o CodeDeploy agente verifique a identidade da instância.
4. Crie uma função de serviço. Forneça acesso ao serviço para que CodeDeploy você possa expandir as tags em sua AWS conta.

Para uma implantação inicial, o CloudFormation modelo faz tudo isso por você. Ele cria e configura novas instâncias únicas do Amazon EC2 com base no Amazon Linux ou no Windows Server com CodeDeploy o agente já instalado. Para obter mais informações, consulte Trabalhando com instâncias para CodeDeploy.

> **Note**
>
> Para uma blue/green implantação, você pode escolher entre usar instâncias que você já tem para o ambiente de substituição ou deixar CodeDeploy provisionar novas instâncias para você como parte do processo de implantação.

Carregando sua revisão de aplicativo

Coloque um AppSpec arquivo na pasta raiz na estrutura de pastas de conteúdo de origem do seu aplicativo. Para obter mais informações, consulte Application Specification Files.

Agrupe a estrutura de pastas de conteúdo de origem do aplicativo em um formato de arquivamento, como zip, tar compactado. Faça o upload do arquivo (a revisão) em um bucket ou GitHub repositório do Amazon S3.

> **Note**
>
> Os formatos de arquivo tar e tar compactado (.tar e .tar.gz) não são compatíveis com instâncias do Windows Server.

Criando seus grupos de implantação e aplicativos

Um grupo de CodeDeploy implantação identifica uma coleção de instâncias com base em suas tags, nomes de grupos do Amazon EC2 Auto Scaling ou ambos. Várias revisões de aplicativos podem ser implantadas nas mesmas instâncias. Uma revisão de aplicativo pode ser implantada em várias instâncias.

Por exemplo, você pode adicionar uma tag "Prod" aos três servidores de produção e "Backup" aos dois servidores de backup. Essas duas tags podem ser usadas para criar dois grupos de implantação diferentes no CodeDeploy aplicativo, permitindo que você escolha qual conjunto de servidores (ou ambos) deve participar de uma implantação.

Você pode usar grupos de várias tags em um grupo de implantação para restringir as implantações em um pequeno conjunto de instâncias. Para mais informações, consulte Tagging Instances for Deployments.

Implantando sua revisão de aplicativo

Agora você está pronto para implantar a revisão do seu aplicativo no Amazon S3 ou no GitHub grupo de implantação. Você pode usar o CodeDeploy console ou o comando create-deployment. Existem parâmetros que você pode especificar para controlar sua implantação, incluindo a revisão, o grupo de implantação e a configuração de implantação.

Atualizar o aplicativo do

Você pode fazer atualizações em seu aplicativo e, em seguida, usar o CodeDeploy console ou chamar o comando create-deployment para enviar uma revisão.

Implantações interrompidas e com falha

Você pode usar o CodeDeploy console ou o comando stop-deployment para interromper uma implantação. Quando você tenta interromper a implantação, uma de três coisas acontece:

- A implantação é interrompida, e a operação retorna a um status de sucesso. Nesse caso, mais nenhum evento de ciclo de vida de implantação é executado no grupo de implantação para a implantação interrompida. Alguns arquivos podem já ter sido copiados para uma ou mais instâncias no grupo de implantação e alguns scripts talvez já tenham sido executados nessas instâncias.
- A implantação não é interrompida imediatamente, e a operação retorna um status pendente. Nesse caso, alguns eventos de ciclo de vida de implantação ainda podem estar em execução no grupo de implantação. Alguns arquivos podem já ter sido copiados para uma ou mais instâncias no grupo de implantação e alguns scripts talvez já tenham sido executados nessas instâncias. Após a conclusão da operação pendente, as chamadas subsequentes para interromper a implantação retornam um status de sucesso.
- A implantação não pode ser interrompida, e a operação retorna um erro. Para obter mais informações, consulte ErrorInformationErros comuns na Referência AWS CodeDeploy da API.

Como implantações interrompidas, implantações com falhas podem resultar em alguns eventos de ciclo de vida de implantação que já foram executados em uma ou mais instâncias no grupo de implantação. Para descobrir por que uma implantação falhou, você pode usar o CodeDeploy console, chamar o comando get-deployment-instance ou analisar os dados do arquivo de log da implantação com falha. Para obter mais informações, consulte Limpeza de revisões de aplicativo e arquivos de log e Exibir dados de registro para CodeDeploy EC2/On-Premises implantações.

Reimplantações e reversões de implantação

CodeDeploy implementa reversões reimplantando, como uma nova implantação, uma revisão implantada anteriormente.

Você pode configurar um grupo de implantação para reverter automaticamente as implantações quando determinadas condições forem atendidas, inclusive quando uma implantação falhar ou um limite de monitoramento de alarme for atingido. Você também pode substituir configurações de reversão especificadas para um grupo de implantação em uma implantação individual.

Além disso, pode optar por reverter uma implantação com falha, reimplantando manualmente uma revisão anteriormente implantada.

Em todos os casos, a implantação nova ou revertida tem um ID de implantação próprio. A lista de implantações que você pode ver no CodeDeploy console mostra quais são o resultado de uma implantação automática.

Para obter mais informações, consulte Reimplante e reverta uma implantação com CodeDeploy.

CodeDeploy arquivos de especificação do aplicativo (AppSpec)

Um arquivo de especificação do aplicativo (AppSpec arquivo), exclusivo do CodeDeploy, é um arquivo formatado em YAML ou em formato JSON. O AppSpec arquivo é usado para gerenciar cada implantação como uma série de ganchos de eventos de ciclo de vida, que são definidos no arquivo.

Para obter informações sobre como criar um AppSpec arquivo bem formado, consulteCodeDeploy AppSpec referência de arquivo.

Tópicos

- AppSpec arquivos em uma plataforma de computação Amazon ECS
- AppSpec arquivos em um AWS Lambda Plataforma de computação
- AppSpec arquivos em uma plataforma de EC2/on-premises computação
- Como o CodeDeploy agente usa o AppSpec arquivo

AppSpec arquivos em uma plataforma de computação Amazon ECS

Se seu aplicativo usa a plataforma de computação Amazon ECS, o AppSpec arquivo pode ser formatado com YAML ou JSON. Ele também poderá ser digitado diretamente em um editor no console. O AppSpec arquivo é usado para especificar:

- O nome do serviço do Amazon ECS, o nome do contêiner e a porta usada para direcionar o tráfego para o novo conjunto de tarefas.
- As funções a serem usadas como testes de validação.

É possível executar funções do Lambda de validação após os eventos de ciclo de vida da implantação. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS, AppSpec estrutura de arquivos para implantações do Amazon ECS e AppSpec Exemplo de arquivo para uma implantação do Amazon ECS.

AppSpec arquivos em um AWS Lambda Plataforma de computação

Se seu aplicativo usa a plataforma de computação AWS Lambda, o AppSpec arquivo pode ser formatado com YAML ou JSON. Ele também poderá ser digitado diretamente em um editor no console. O AppSpec arquivo é usado para especificar:

- A versão da AWS Lambda função a ser implantada.
- As funções a serem usadas como testes de validação.

É possível executar funções do Lambda de validação após os eventos de ciclo de vida da implantação. Para obter mais informações, consulte AppSpec seção 'ganchos' para um AWS implantação do Lambda.

AppSpec arquivos em uma plataforma de EC2/on-premises computação

Se seu aplicativo usa a plataforma de EC2/On-Premises computação, o AppSpec arquivo sempre YAML-formatted estará. O AppSpec arquivo é usado para:

- Mapear os arquivos de origem na sua revisão de aplicativo para os respectivos destinos na instância.
- Especificar permissões personalizadas para arquivos implantados.
- Especificar scripts a serem executados em cada instância, em vários estágios do processo de implantação.

Você pode executar scripts em uma instância após muitos dos eventos individuais do ciclo de vida da implantação. CodeDeploy executa somente os scripts especificados no arquivo, mas esses scripts podem chamar outros scripts na instância. Você pode executar qualquer tipo de script, desde que ele

seja compatível no sistema operacional em execução nas instâncias. Para obter mais informações, consulte AppSpec seção 'ganchos' para uma implantação EC2/On-Premises.

Como o CodeDeploy agente usa o AppSpec arquivo

Durante a implantação, o CodeDeploy agente procura o nome do evento atual na seção de ganchos do AppSpec arquivo. Se o evento não for encontrado, o agente do CodeDeploy passará para a próxima etapa. Se o evento for encontrado, o CodeDeploy agente recuperará a lista de scripts a serem executados. Os scripts são executados sequencialmente, na ordem em que aparecem no arquivo. O status de cada script é registrado no arquivo de log do CodeDeploy agente na instância.

Se um script for executado com sucesso, ele retornará um código de saída de 0 (zero).

> **Note**
>
> O CodeDeploy agente não é usado em uma implantação do AWS Lambda ou do Amazon ECS.

Durante o evento de instalação, o CodeDeploy agente usa os mapeamentos definidos na seção de arquivos do AppSpec arquivo para determinar quais pastas ou arquivos copiar da revisão para a instância.

Se o CodeDeploy agente instalado no sistema operacional não corresponder ao que está listado no AppSpec arquivo, a implantação falhará.

Para obter informações sobre os arquivos de log do CodeDeploy agente, consulteTrabalhando com o CodeDeploy agente.

Começando com CodeDeploy

Tópicos

- Etapa 1: configuração
- Etapa 2: criar uma função de serviço para CodeDeploy
- Etapa 3: limitar as permissões do CodeDeploy usuário
- Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2

Etapa 1: configuração

Antes de usar AWS CodeDeploy pela primeira vez, você deve concluir as etapas de configuração. As etapas envolvem a criação de uma AWS conta (se você ainda não tiver uma) e um usuário administrativo.

Neste guia, o usuário administrativo é chamado de usuário CodeDeploy administrativo.

Inscreva-se para um Conta da AWS

Para começar AWS, você precisa de um Conta da AWS. Há duas maneiras de se inscrever em AWS. Se você quiser ter controle total sobre a configuração da conta e onde seu conteúdo é armazenado, ou se tiver cargas de trabalho regulamentadas, como HIPAA ou FedRAMP, use Inscrever-se em (avançado). AWS Se você for um novo usuário e quiser usar um login que já possui para entrar AWS e criar um AWS ambiente com padrões pré-configurados em select Regiões da AWS, use Inscrever-se em. AWS

> **Important**
>
> É altamente recomendável configurar o usuário CodeDeploy administrativo como uma identidade da força de trabalho (um usuário gerenciado no IAM Identity Center) com o. AWS CLI Muitos dos procedimentos deste guia pressupõem que você esteja usando o AWS CLI para realizar configurações.

> **Important**
>
> Se você configurar o AWS CLI, poderá ser solicitado que você especifique uma AWS região. Escolha uma das regiões com suporte listada em Região e endpoints em Referência geral da AWS.

Etapa 2: criar uma função de serviço para CodeDeploy

Em AWS, as funções de serviço são usadas para conceder permissões a um AWS serviço para que ele possa acessar AWS recursos. As políticas que você anexa ao perfil de serviço determinam quais recursos o serviço pode acessar e o que ele pode fazer com esses recursos.

A função de serviço para a qual você cria CodeDeploy deve receber as permissões necessárias para sua plataforma de computação. Se você implantar para mais de uma plataforma de computação, crie um perfil de serviço para cada uma delas. Para adicionar permissões, anexe uma ou mais das seguintes políticas AWS fornecidas:

Para EC2/On-Premises implantações, anexe a **`AWSCodeDeployRole`** política. Ela fornece as permissões ao perfil de serviço para:

- Leia as tags nas instâncias ou identifique as instâncias do Amazon EC2 pelos nomes de grupo do Amazon EC2 Auto Scaling.
- Ler, criar, atualizar e excluir grupos do Amazon EC2 Auto Scaling, ganchos do ciclo de vida e políticas de escalabilidade.
- Publique informações nos tópicos do Amazon SNS.
- Recupere informações sobre CloudWatch alarmes.
- Leia e atualize o Elastic Load Balancing.

> **Note**
>
> Se criou grupo do Auto Scaling com um modelo de execução, você deverá adicionar as seguintes permissões:
>
> - `ec2:RunInstances`
> - `ec2:CreateTags`
> - `iam:PassRole`

> Para obter mais informações sobre como criar modelos de execução, consulte Etapa 2: Criar um perfil de serviço, Criar um modelo de execução para um grupo do Auto Scaling e Executar suporte de modelo no Guia do usuário do Amazon EC2 Auto Scaling.

Para implantações Amazon ECS, se você quiser aceso total aos serviços de suporte, anexe a política **`AWSCodeDeployRoleForECS`**. Ela fornece as permissões ao perfil de serviço para:

- Ler, atualizar e excluir conjuntos de tarefas do Amazon ECS.
- Atualizar grupos de destino, receptores e regras do Elastic Load Balancing.
- Invoque AWS Lambda funções.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Para implantações Amazon ECS, se você quiser acesso limitado a serviços de suporte, anexe a política **`AWSCodeDeployRoleForECSLimited`**. Ela fornece as permissões ao perfil de serviço para:

- Ler, atualizar e excluir conjuntos de tarefas do Amazon ECS.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Para implantações do AWS Lambda, se você quiser permitir a publicação no Amazon SNS, anexe a política. **`AWSCodeDeployRoleForLambda`** Ela fornece as permissões ao perfil de serviço para:

- Leia, atualize e invoque AWS Lambda funções e aliases.
- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.
- Publique informações nos tópicos do Amazon SNS.

Para implantações do AWS Lambda, se você quiser limitar o acesso ao Amazon SNS, anexe a política. **`AWSCodeDeployRoleForLambdaLimited`** Ela fornece as permissões ao perfil de serviço para:

- Leia, atualize e invoque AWS Lambda funções e aliases.

- Acessar arquivos de revisão em buckets do Amazon S3.
- Recupere informações sobre CloudWatch alarmes.

Como parte da configuração do perfil de serviço, você também atualiza seu relacionamento de confiança para especificar os endpoints aos quais deseja conceder acesso.

Você pode criar uma função de serviço com o console do IAM AWS CLI, o ou as APIs do IAM.

Tópicos

- Criar um perfil de serviço (console)
- Criar um perfil de serviço (CLI)
- Obter o ARN do perfil de serviço (console)
- Obter o ARN do perfil de serviço (CLI)

Criar 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, escolha Perfis e Criar perfil.
3. Escolha AWS serviço e, em Caso de uso, na lista suspensa, escolha. CodeDeploy
4. Escolha o caso de uso:
   - Para EC2/On-Premises implantações, escolha CodeDeploy.
   - Para implantações do AWS Lambda, escolha Lambda. CodeDeploy
   - Para implantações do Amazon ECS, escolha CodeDeploy - ECS.
5. Escolha Próximo.
6. Na página Adicionar permissões, a política de permissões correta para o caso de uso é exibida. Escolha Próximo.
7. Na página Nomear, revisar e criar, em Nome do perfil, insira um nome para o perfil de serviço (por exemplo, **`CodeDeployServiceRole`**) e escolha Criar perfil.

   Você também pode digitar uma descrição para este perfil de serviço na caixa Descrição do perfil.
8. Se quiser que esse perfil de serviço tenha permissão para acessar todos os endpoints atualmente com suporte, você concluiu este procedimento.

Para restringir o acesso desse perfil de serviço a alguns endpoints, continue com as etapas restantes desse procedimento.

9. Procure na lista de perfis e selecione o perfil que você acabou de criar (`CodeDeployServiceRole`).
10. Selecione a guia Trust relationships (Relações de confiança).
11. Selecione Edit trust policy (Editar política de confiança).

Você verá a seguinte política, que fornece permissão ao perfil de serviço para acessar todos os endpoints com suporte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "codedeploy.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Para conceder ao perfil de serviço acesso a somente alguns endpoints com suporte, substitua o conteúdo da caixa Documento de Política pela política a seguir. Remova as linhas dos endpoints aos quais você deseja impedir o acesso e selecione Atualizar política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [

                    "codedeploy.us-east-1.amazonaws.com",
```

```
                    "codedeploy.us-east-2.amazonaws.com",
                    "codedeploy.us-west-1.amazonaws.com",
                    "codedeploy.us-west-2.amazonaws.com",
                    "codedeploy.ca-central-1.amazonaws.com",
                    "codedeploy.ap-east-1.amazonaws.com",
                    "codedeploy.ap-northeast-1.amazonaws.com",
                    "codedeploy.ap-northeast-2.amazonaws.com",
                    "codedeploy.ap-northeast-3.amazonaws.com",
                    "codedeploy.ap-southeast-1.amazonaws.com",
                    "codedeploy.ap-southeast-2.amazonaws.com",
                    "codedeploy.ap-southeast-3.amazonaws.com",
                    "codedeploy.ap-southeast-4.amazonaws.com",
                    "codedeploy.ap-south-1.amazonaws.com",
                    "codedeploy.ap-south-2.amazonaws.com",
                    "codedeploy.ca-central-1.amazonaws.com",
                    "codedeploy.eu-west-1.amazonaws.com",
                    "codedeploy.eu-west-2.amazonaws.com",
                    "codedeploy.eu-west-3.amazonaws.com",
                    "codedeploy.eu-central-1.amazonaws.com",
                    "codedeploy.eu-central-2.amazonaws.com",
                    "codedeploy.eu-north-1.amazonaws.com",
                    "codedeploy.eu-south-1.amazonaws.com",
                    "codedeploy.eu-south-2.amazonaws.com",
                    "codedeploy.il-central-1.amazonaws.com",
                    "codedeploy.me-central-1.amazonaws.com",
                    "codedeploy.me-south-1.amazonaws.com",
                    "codedeploy.sa-east-1.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Para obter mais informações sobre a criação de funções de serviço, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Criar um perfil de serviço (CLI)

1. Na sua máquina de desenvolvimento, crie um arquivo de texto denominado, por exemplo, `CodeDeployDemo-Trust.json`. Esse arquivo é usado para permitir que o CodeDeploy trabalhe em seu nome.

Execute um destes procedimentos:

- Para conceder acesso a todas as AWS regiões suportadas, salve o seguinte conteúdo no arquivo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "codedeploy.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

- Para conceder acesso somente a algumas regiões suportadas, digite o conteúdo a seguir no arquivo e remova as linhas das regiões às quais você deseja excluir o acesso:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [

                    "codedeploy.us-east-1.amazonaws.com",
                    "codedeploy.us-east-2.amazonaws.com",
                    "codedeploy.us-west-1.amazonaws.com",
                    "codedeploy.us-west-2.amazonaws.com",
                    "codedeploy.ca-central-1.amazonaws.com",
                    "codedeploy.ap-east-1.amazonaws.com",
                    "codedeploy.ap-northeast-1.amazonaws.com",
                    "codedeploy.ap-northeast-2.amazonaws.com",
                    "codedeploy.ap-northeast-3.amazonaws.com",
```

```
                    "codedeploy.ap-southeast-1.amazonaws.com",
                    "codedeploy.ap-southeast-2.amazonaws.com",
                    "codedeploy.ap-southeast-3.amazonaws.com",
                    "codedeploy.ap-southeast-4.amazonaws.com",
                    "codedeploy.ap-south-1.amazonaws.com",
                    "codedeploy.ap-south-2.amazonaws.com",
                    "codedeploy.ca-central-1.amazonaws.com",
                    "codedeploy.eu-west-1.amazonaws.com",
                    "codedeploy.eu-west-2.amazonaws.com",
                    "codedeploy.eu-west-3.amazonaws.com",
                    "codedeploy.eu-central-1.amazonaws.com",
                    "codedeploy.eu-central-2.amazonaws.com",
                    "codedeploy.eu-north-1.amazonaws.com",
                    "codedeploy.eu-south-1.amazonaws.com",
                    "codedeploy.eu-south-2.amazonaws.com",
                    "codedeploy.il-central-1.amazonaws.com",
                    "codedeploy.me-central-1.amazonaws.com",
                    "codedeploy.me-south-1.amazonaws.com",
                    "codedeploy.sa-east-1.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

> **Note**
>
> Não use uma vírgula após o último endpoint na lista.

2. No mesmo diretório, chame o comando create-role para criar um perfil de serviço denominado **`CodeDeployServiceRole`**, com base nas informações do arquivo de texto que você acabou de criar:

```
aws iam create-role --role-name CodeDeployServiceRole --assume-role-policy-document
 file://CodeDeployDemo-Trust.json
```

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

Na saída do comando, anote o valor da entrada `Arn` no objeto `Role`. Você precisará disso mais tarde ao criar grupos de implantação. Se você se esquecer do valor, siga as instruções em Obter o ARN do perfil de serviço (CLI).

3. A política gerenciada que você usa depende da plataforma de computação.

   - Se sua implantação for em uma plataforma de EC2/On-Premises computação:

     Chame o comando attach-role-policy para dar ao perfil de serviço chamado **`CodeDeployServiceRole`** as permissões com base na política gerenciada pelo IAM chamada **`AWSCodeDeployRole`**. Por exemplo:

     ```
     aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
      arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole
     ```

   - Se sua implantação for em uma plataforma de computação AWS Lambda:

     Chame o comando attach-role-policy para conceder ao perfil de serviço chamado **`CodeDeployServiceRole`** as permissões com base na política gerenciada pelo IAM chamada **`AWSCodeDeployRoleForLambda`** ou **`AWSCodeDeployRoleForLambdaLimited`**. Por exemplo:

     ```
     aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
      arn:aws:iam::aws:policy/service-role/AWSCodeDeployRoleForLambda
     ```

   - Se sua implantação é para uma plataforma de computação do Amazon ECS:

     Chame o comando attach-role-policy para conceder ao perfil de serviço chamado **`CodeDeployServiceRole`** as permissões com base na política gerenciada pelo IAM chamada **`AWSCodeDeployRoleForECS`** ou **`AWSCodeDeployRoleForECSLimited`**. Por exemplo:

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
 arn:aws:iam::aws:policy/AWSCodeDeployRoleForECS
```

Para obter mais informações sobre a criação de funções de serviço, consulte Como criar uma função para um AWS serviço no Guia do usuário do IAM.

Obter o ARN do perfil de serviço (console)

Para usar o console do IAM para obter o ARN do perfil de serviço:

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, escolha Perfis.
3. Na caixa de texto Filter (Filtro), digite **`CodeDeployServiceRole`** e pressione Enter.
4. Selecione CodeDeployServiceRole.
5. Anote o valor do campo ARN do perfil.

Obter o ARN do perfil de serviço (CLI)

Para usar o AWS CLI para obter o ARN da função de serviço, chame o get-role comando na função de serviço chamada: **`CodeDeployServiceRole`**

```
aws iam get-role --role-name CodeDeployServiceRole --query "Role.Arn" --output text
```

O valor retornado é o ARN do perfil de serviço.

Etapa 3: limitar as permissões do CodeDeploy usuário

Por motivos de segurança, recomendamos que você limite as permissões do usuário administrativo que você criou apenas Etapa 1: configuração às necessárias para criar e gerenciar implantações no CodeDeploy.

Use a série de procedimentos a seguir para limitar as permissões do usuário CodeDeploy administrativo.

Antes de começar

- Certifique-se de ter criado um usuário CodeDeploy administrativo no IAM Identity Center seguindo as instruções emEtapa 1: configuração.

Para criar um conjunto de permissões

Você atribuirá esse conjunto de permissões ao usuário CodeDeploy administrativo posteriormente.

1. Faça login no Console de gerenciamento da AWS e abra o Centro de Identidade do AWS IAM console em https://console.aws.amazon.com/singlesignon/.
2. No painel de navegação, escolha Conjuntos de permissões e, em seguida, escolha Criar conjunto de permissões.
3. Escolha Conjunto de permissões personalizado.
4. Escolha Próximo.
5. Selecione Políticas em linha.
6. Remova o código de exemplo.
7. Adicione a política a seguir:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeDeployAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "autoscaling:*",
                "codedeploy:*",
                "ec2:*",
                "lambda:*",
                "ecs:*",
                "elasticloadbalancing:*",
                "iam:AddRoleToInstanceProfile",
                "iam:AttachRolePolicy",
                "iam:CreateInstanceProfile",
                "iam:CreateRole",
                "iam:DeleteInstanceProfile",
```

```
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListInstanceProfilesForRole",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:PutRolePolicy",
                "iam:RemoveRoleFromInstanceProfile",
                "s3:*",
                "ssm:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeDeployRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/
CodeDeployServiceRole"
        }
    ]
}
```

Nessa política, *arn:aws:iam::account-ID:role/CodeDeployServiceRole* substitua pelo valor ARN da função de CodeDeploy serviço que você criou em. Etapa 2: criar uma função de serviço para CodeDeploy É possível encontrar o valor do ARN na página de detalhes do perfil de serviço no console do IAM.

A política anterior permite que você implante um aplicativo em uma plataforma de computação AWS Lambda, EC2/On-Premises uma plataforma de computação e uma plataforma de computação Amazon ECS.

Você pode usar os CloudFormation modelos fornecidos nesta documentação para iniciar instâncias do Amazon EC2 que sejam compatíveis com o. CodeDeploy Para usar CloudFormation modelos para criar aplicativos, grupos de implantação ou configurações de implantação, você deve fornecer acesso a CloudFormation— e AWS serviços e ações que

CloudFormation dependam — adicionando a `cloudformation:*` permissão à política de permissão do usuário CodeDeploy administrativo, da seguinte forma:

JSON

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

8. Escolha Próximo.
9. Em Nome do conjunto de permissões, digite:

```
CodeDeployUserPermissionSet
```

10. Escolha Próximo.
11. Na página Revisar e criar, revise as informações e selecione Criar.

Para atribuir o conjunto de permissões ao usuário CodeDeploy administrativo

1. No painel de navegação Contas da AWS, escolha e marque a caixa de seleção ao lado da caixa de seleção na Conta da AWS qual você está conectado no momento.
2. Escolha o botão Atribuir usuários ou grupos.
3. Escolha a guia Users.
4. Marque a caixa de seleção ao lado do usuário CodeDeploy administrativo.
5. Escolha Próximo.
6. Marque a caixa de seleção ao lado dos logs do `CodeDeployUserPermissionSet`.
7. Escolha Próximo.
8. Revise suas informações e selecione Enviar.

Agora você atribuiu o usuário CodeDeploy administrativo e `CodeDeployUserPermissionSet` ao seu Conta da AWS, vinculando-os.

Para sair e entrar novamente como usuário CodeDeploy administrativo

1. Antes de sair, verifique se você tem a URL do portal de AWS acesso, o nome de usuário e a senha de uso único do usuário CodeDeploy administrativo.

   > **Note**
   >
   > Se você não tiver essas informações, acesse a página de detalhes do usuário CodeDeploy administrativo no IAM Identity Center, escolha Redefinir senha, Gerar uma senha de uso único [...] e redefina a senha novamente para exibir as informações na tela.

2. Sair de AWS.
3. Cole o URL do portal de AWS acesso na barra de endereço do seu navegador.
4. Faça login como usuário CodeDeploy administrativo.

   Uma caixa de Conta da AWS aparece na tela.
5. Escolha e Conta da AWS, em seguida, escolha o nome do Conta da AWS ao qual você atribuiu o usuário CodeDeploy administrativo e o conjunto de permissões.
6. Ao lado de `CodeDeployUserPermissionSet`, selecione Console de gerenciamento.

   O Console de gerenciamento da AWS aparece. Agora você está conectado como usuário CodeDeploy administrativo com as permissões limitadas. Agora você pode realizar CodeDeploy-related operações, e somente CodeDeploy-related operações, como esse usuário.

Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2

> **Note**
>
> Se você estiver usando a plataforma de computação Amazon ECS ou AWS Lambda, pule esta etapa.

Suas instâncias do Amazon EC2 precisam de permissão para acessar os buckets GitHub ou repositórios do Amazon S3 onde os aplicativos estão armazenados. Para iniciar instâncias do Amazon EC2 que sejam compatíveis com CodeDeploy, você deve criar uma função adicional do IAM, um perfil de instância. Estas instruções mostram como criar um perfil de instância do IAM; para anexar às suas instâncias do Amazon EC2. Essa função dá ao CodeDeploy agente permissão para acessar os buckets ou GitHub repositórios do Amazon S3 onde seus aplicativos estão armazenados.

Você pode criar um perfil de instância do IAM com o AWS CLI console do IAM ou as APIs do IAM.

> **Note**
>
> Você pode anexar um perfil da instância do IAM a uma instância do Amazon EC2 ao executá-la ou a uma instância executada anteriormente. Para obter mais informações, consulte Perfis de instância.

Tópicos

- Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (CLI)
- Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (console)

Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (CLI)

Nestas etapas, partimos do princípio de que você já seguiu as instruções das três primeiras etapas em Começando com CodeDeploy.

1. Na máquina de desenvolvimento, crie um arquivo de texto chamado `CodeDeployDemo-EC2-Trust.json`. Cole o seguinte conteúdo, que permite ao Amazon EC2; trabalhar em seu nome:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ec2.cn-north-1.amazonaws.com",
                    "ec2.cn-northwest-1.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. No mesmo diretório, crie um arquivo de texto chamado `CodeDeployDemo-EC2-Permissions.json`. Cole o seguinte conteúdo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

```
}
```

> **Note**
>
> Recomendamos que você restrinja essa política somente aos buckets do Amazon S3 que suas instâncias do Amazon EC2 precisam acessar. Certifique-se de dar acesso aos buckets do Amazon S3 que contêm o agente. CodeDeploy Caso contrário, poderá ocorrer um erro quando o CodeDeploy agente for instalado ou atualizado nas instâncias. Para conceder ao perfil da instância do IAM acesso somente a alguns buckets do kit de CodeDeploy recursos no Amazon S3, use a política a seguir, mas remova as linhas dos buckets aos quais você deseja impedir o acesso:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
```

```
        "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
        "arn:aws:s3:::aws-codedeploy-me-central-1/*",
        "arn:aws:s3:::aws-codedeploy-me-south-1/*",
        "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
      ]
    }
  ]
}
```

> **Note**
>
> Se você quiser usar a autorização do IAM ou os endpoints da Amazon Virtual Private Cloud (VPC) com CodeDeploy, você precisará adicionar mais permissões. Consulte Usar CodeDeploy com a Amazon Virtual Private Cloud para obter mais informações.

3. No mesmo diretório, chame o comando create-role para criar um perfil do IAM denominada **`CodeDeployDemo-EC2-Instance-Profile`**, com base nas informações do arquivo:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws iam create-role --role-name CodeDeployDemo-EC2-Instance-Profile --assume-role-policy-document file://CodeDeployDemo-EC2-Trust.json
```

4. No mesmo diretório, chame o comando put-role-policy para fornecer à função denominada **`CodeDeployDemo-EC2-Instance-Profile`** as permissões com base nas informações do segundo arquivo:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws iam put-role-policy --role-name CodeDeployDemo-EC2-Instance-Profile --policy-
name CodeDeployDemo-EC2-Permissions --policy-document file://CodeDeployDemo-EC2-
Permissions.json
```

5. Ligue attach-role-policy para dar à função Amazon EC2 Systems Manager permissões para que o SSM possa instalar o CodeDeploy agente. Esta política não é necessária se você planeja instalar o agente pelo bucket público do Amazon S3 com a linha de comando. Saiba mais sobre a instalação do agente do CodeDeploy .

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonSSMManagedInstanceCore --role-name CodeDeployDemo-EC2-Instance-Profile
```

6. Chame o comando create-instance-profile seguido do comando add-role-to-instance-profile para criar um perfil de instância do IAM denominado **`CodeDeployDemo-EC2-Instance-Profile`**. O perfil de instância permite que o Amazon EC2 transmita o perfil do IAM denominado **`CodeDeployDemo-EC2-Instance-Profile`** para uma instância do Amazon EC2 quando essa instância for iniciada pela primeira vez:

```
aws iam create-instance-profile --instance-profile-name CodeDeployDemo-EC2-
Instance-Profile
aws iam add-role-to-instance-profile --instance-profile-name CodeDeployDemo-EC2-
Instance-Profile --role-name CodeDeployDemo-EC2-Instance-Profile
```

   Se você precisar obter o nome do perfil de instância IAM, consulte list-instance-profiles-for-role na seção IAM do documento de AWS CLI Referência.

Agora, você criou um perfil de instância do IAM; a ser anexado às suas instâncias do Amazon EC2. Para obter mais informações, consulte Perfis do IAM para o Amazon EC2 no Guia do usuário do Amazon EC2.

Criar um perfil de instância do IAM; para as suas instâncias do Amazon EC2 (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 console do IAM, no painel de navegação, escolha Políticas e Criar política.
3. Na página Especificar permissões, escolha JSON.
4. Remova o `JSON` código de exemplo.
5. Cole o seguinte código:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Recomendamos que você restrinja essa política somente aos buckets do Amazon S3 que suas instâncias do Amazon EC2 precisam acessar. Certifique-se de dar acesso aos buckets do Amazon S3 que contêm o agente. CodeDeploy Caso contrário, poderá ocorrer um erro quando o CodeDeploy agente for instalado ou atualizado nas instâncias. Para conceder ao perfil da instância do IAM acesso somente a alguns buckets do kit de CodeDeploy recursos no Amazon S3, use a política a seguir, mas remova as linhas dos buckets aos quais você deseja impedir o acesso:
>
> ```
> {
>   "Version": "2012-10-17",
>   "Statement": [
>     {
> ```

```
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
        "arn:aws:s3:::aws-codedeploy-me-central-1/*",
        "arn:aws:s3:::aws-codedeploy-me-south-1/*",
        "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
      ]
    }
  ]
}
```

> **Note**
>
> Se você quiser usar a autorização do IAM ou os endpoints da Amazon Virtual Private Cloud (VPC) com CodeDeploy, você precisará adicionar mais permissões. Consulte Usar CodeDeploy com a Amazon Virtual Private Cloud para obter mais informações.

6. Escolha Próximo.
7. Na página Revisar e criar, em Nome da política, digite **`CodeDeployDemo-EC2-Permissions`**.
8. (Opcional) Em Descrição, digite uma descrição para a política.
9. Escolha Criar política.
10. No painel de navegação, escolha Perfis e Criar perfil.
11. Em Caso de uso, escolha o caso de uso do EC2.
12. Escolha Próximo.
13. Na lista de políticas, marque a caixa de seleção ao lado da política que você acabou de criar (CodeDeployDemo-EC2-Permissions). Se necessário, use a caixa de pesquisa para encontrar a política.
14. Para usar o Systems Manager para instalar ou configurar o CodeDeploy agente, marque a caixa de seleção ao lado de AmazonSSMManagedInstanceCore. Essa política AWS gerenciada permite que uma instância use a funcionalidade principal do serviço do Systems Manager. Se necessário, use a caixa de pesquisa para encontrar a política. Esta política não é necessária se você planeja instalar o agente pelo bucket público do Amazon S3 com a linha de comando. Saiba mais sobre a instalação do agente do CodeDeploy.
15. Escolha Próximo.
16. Na página Nomear, revisar e criar, em Nome do perfil, insira um nome para o perfil de serviço (por exemplo, **`CodeDeployDemo-EC2-Instance-Profile`**) e escolha Criar perfil.

    Você também pode digitar uma descrição para este perfil de serviço na caixa Descrição do perfil.

Agora, você criou um perfil de instância do IAM; a ser anexado às suas instâncias do Amazon EC2. Para obter mais informações, consulte Perfis do IAM para o Amazon EC2 no Guia do usuário do Amazon EC2.

Integrações de produtos e serviços com CodeDeploy

Por padrão, CodeDeploy se integra a vários AWS serviços e produtos e serviços de parceiros. As informações a seguir podem ajudá-lo a configurar CodeDeploy a integração com os produtos e serviços que você usa.

- Integração com outros AWS serviços
- Integração com produtos e serviços de parceiros
- Exemplos de integração da comunidade

Integração com outros AWS serviços

CodeDeploy está integrado aos seguintes AWS serviços:

Amazon CloudWatch	A Amazon CloudWatch é um serviço de monitoramento dos recursos da AWS nuvem e dos aplicativos em que você executa AWS. Você pode usar CloudWatch a Amazon para coletar e rastrear métricas, coletar e monitorar arquivos de log e definir alarmes. CodeDeploy suporta as seguintes CloudWatch ferramentas: • CloudWatch alarmes para monitorar suas implantações e interrompê-las quando suas métricas de monitoramento especificadas excederem ou ficarem abaixo dos limites especificados em uma CloudWatch regra de alarme. Para usar o monitoramento de alarmes, primeiro você configura um alarme e CloudWatch, em seguida, o adiciona CodeDeploy ao aplicativo ou grupo de implantação, onde as implantações devem parar quando o alarme for ativado. Saiba mais:

- Criação de CloudWatch alarmes de registros

- Amazon CloudWatch Events para detectar e reagir a mudanças no estado de uma instância ou implantação em suas CodeDeploy operações. Em seguida, com base nas regras que você cria, o CloudWatch Events invoca uma ou mais ações de destino quando uma implantação ou instância entra no estado especificado em uma regra.

  Saiba mais:

  - Monitoramento de implantações com Amazon Events CloudWatch

- Amazon CloudWatch Logs para monitorar os três tipos de registros criados pelo CodeDeploy agente sem precisar fazer login nas instâncias uma por vez.

  Saiba mais:

  - Exibir CodeDeploy registros no console CloudWatch de registros

Amazon EC2 Auto Scaling	CodeDeploy oferece suporte ao Amazon EC2 Auto Scaling. Esse AWS serviço pode iniciar automaticamente instâncias do Amazon EC2 com base nos critérios que você especificar, por exemplo: • Os limites excedidos para a utilização especificada da CPU. • Leituras/gravações de disco. • O tráfego de rede de entrada ou saída durante um intervalo de tempo especificado. Você pode escalar um grupo de instâncias do Amazon EC2 sempre que precisar delas e depois usá-las CodeDeploy para implantar revisões de aplicativos nelas automaticamente. O Amazon EC2 Auto Scaling encerra essas instâncias do Amazon EC2 quando elas não são mais necessárias. Saiba mais: • Integração CodeDeploy com o Amazon EC2 Auto Scaling • Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling • Sob o capô: CodeDeploy e integração com o Auto Scaling

Amazon Elastic Container Service	Você pode usar CodeDeploy para implantar um aplicativo em contêineres do Amazon ECS como um conjunto de tarefas. CodeDeploy executa uma blue/green implantação instaland o uma versão atualizada do aplicativo como um novo conjunto de tarefas de substituição. CodeDeploy redireciona o tráfego de produção do conjunto de tarefas original do aplicativo para o conjunto de tarefas de substituição. O conjunto de tarefas original é encerrado após uma implantação bem-sucedida. Para obter mais informações sobre o Amazon ECS, consulte Amazon Elastic Container Service. Você pode gerenciar a maneira como o tráfego é deslocado para o conjunto de tarefas atualizado durante uma implantação escolhend o uma configuração canário, linear ou todas de uma só vez. Para obter mais informaçõ es sobre implantações do Amazon ECS, consulte Implantações em uma plataforma de computação do Amazon ECS.

AWS CloudTrail	CodeDeploy está integrado com AWS CloudTrail. Esse serviço captura chamadas de API feitas por ou em nome de CodeDeplo y sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrailcaptura chamadas de API do CodeDeploy console, de CodeDeploy comandos por meio do ou AWS CLI diretamen te das CodeDeploy APIs. Usando as informaçõ es coletadas pelo CloudTrail, você pode determinar: • Qual solicitação foi feita para CodeDeploy. • O endereço IP de origem do qual a solicitaç ão foi feita. • Quem fez a solicitação. • Quando ela foi feita. Saiba mais: • Monitoring Deployments

AWS Cloud9	AWS Cloud9é um ambiente de desenvolv imento integrado (IDE) on-line e baseado em nuvem que você pode usar para escrever, executar, depurar e implantar código usando apenas um navegador de uma máquina conectada à Internet. AWS Cloud9 inclui um editor de código, depurador, terminal e ferramentas essenciais, como o e o Git. AWS CLI • Você pode usar o AWS Cloud9 IDE para executar, depurar e criar o código que está em um GitHub repositório. Você pode visualizar, alterar e salvar o código usando as guias do editor e a janela Environment (Ambiente) do IDE. Quando estiver pronto, você pode usar o Git na sessão do AWS Cloud9 terminal para enviar alterações de código para o seu GitHub repositório e, em seguida, usá-lo AWS CodeDeploy para implantar suas atualizações. Para obter mais informações sobre como usar AWS Cloud9 com GitHub, consulte o GitHub exemplo para AWS Cloud9. • Você pode usar o AWS Cloud9 IDE para atualizar uma AWS Lambda função. Em seguida, você pode usar AWS CodeDeplo y para criar uma implantação que transfere o tráfego para a nova versão da sua AWS Lambda função. Para obter mais informaçõ es, consulte Trabalhando com AWS Lambda funções no Ambiente de Desenvolvimento AWS Cloud9 Integrado (IDE).

	Para obter mais informações sobre AWS Cloud9, consulte O que é AWS Cloud9 e Como começar a usar AWS Cloud9.

AWS CodePipeline	AWS CodePipeline é um serviço de entrega contínua que você pode usar para modelar, visualizar e automatizar as etapas necessári as para lançar seu software em um processo de entrega contínua. Você pode usar o AWS CodePipeline para definir seu próprio processo de lançamento, para que o serviço crie, teste e implante seu código toda vez que houver uma mudança de código. Por exemplo, você pode ter três grupos de implantação para um aplicativo: Beta, Gamma e Prod. Você pode configurar um pipeline de forma que, cada vez que houver uma mudança no código-fonte, as atualizações sejam implantadas em cada grupo de implantação, uma a uma. Você pode configurar AWS CodePipeline para usar CodeDeploy para implantar: • Código para instâncias do Amazon EC2, instâncias on-premises ou ambas. • Versões da função AWS Lambda sem servidor. Você pode criar o CodeDeploy aplicativo, a implantação e o grupo de implantação para usar em uma ação de implantação em um estágio antes de criar o pipeline ou no assistent e Create Pipeline. Saiba mais: • AWS para DevOps obter um guia de introdução — Saiba como usar o CodePipel ine with CodeDeploy para entregar e implantar continuamente o código-fonte em

	CodeCommit repositórios para instâncias do Amazon EC2. • Passo a passo simples do pipeline (bucket do Amazon S3) • Passo a passo simples do pipeline (repositório) CodeCommit • Four-stagetutorial de pipeline
AWS Modelo de aplicativo sem servidor	AWS O Serverless Application Model (AWS SAM) é um modelo para definir aplicativos sem servidor. Ele se estende CloudFormation para fornecer uma forma simplificada de definir AWS Lambda funções, APIs do Amazon API Gateway e tabelas do Amazon DynamoDB exigidas por um aplicativo sem servidor. Se você já usa o AWS SAM, pode adicionar preferências de implantação para começar a usar CodeDeploy para gerenciar a forma como o tráfego é deslocado durante a implantação de um aplicativo AWS Lambda. Para obter mais informações, consulte AWS Serveless Application Model.

Elastic Load Balancing	CodeDeploy suporta o Elastic Load Balancing , um serviço que distribui o tráfego de entrada do aplicativo em várias instâncias do Amazon EC2. Para CodeDeploy implantações, os balanceadores de carga também evitam que o tráfego seja roteado para instâncias quando elas não estão prontas, estão sendo implantadas no momento ou não são mais necessárias como parte de um ambiente. Saiba mais: • Integrating CodeDeploy with Elastic Load Balancing

Tópicos

- Integração CodeDeploy com o Amazon EC2 Auto Scaling
- Integração CodeDeploy com o Elastic Load Balancing

Integração CodeDeploy com o Amazon EC2 Auto Scaling

CodeDeploy oferece suporte ao Amazon EC2 Auto Scaling AWS , um serviço que lança instâncias do Amazon EC2 automaticamente de acordo com as condições definidas por você. Essas condições podem incluir limites excedidos em um intervalo de tempo especificado para utilização da CPU, leituras ou gravações de disco ou tráfego de rede de entrada ou saída. O Amazon EC2 Auto Scaling encerra as instâncias quando elas não são mais necessárias. Para obter mais informações, consulte O que é o Amazon EC2 Auto Scaling? no Guia do usuário do Amazon EC2 Auto Scaling.

Quando novas instâncias do Amazon EC2 são lançadas como parte de um grupo do Amazon EC2 Auto Scaling CodeDeploy , você pode implantar suas revisões nas novas instâncias automaticamente. Você também pode coordenar implantações com instâncias do Amazon EC2 Auto Scaling registradas CodeDeploy com balanceadores de carga do Elastic Load Balancing. Para obter mais informações, consulte Integrating CodeDeploy with Elastic Load Balancing e Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2.

> **Note**
>
> Lembre-se de que você pode ter problemas se associar vários grupos de implantação a um único grupo do Amazon EC2 Auto Scaling. Se uma implantação falhar, por exemplo, a instância começará a ser desligada, mas as outras implementações que estavam em execução poderão demorar uma hora para atingirem o tempo limite. Para obter mais informações, consulte Evite associar vários grupos de implantação a um único grupo de Amazon EC2 Auto Scaling Under the hood: CodeDeploy and Amazon EC2 Auto Scaling integration.

Tópicos

- Implantação de CodeDeploy aplicativos em grupos do Amazon EC2 Auto Scaling
- Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling
- Como o Amazon EC2 Auto Scaling funciona com CodeDeploy
- Usando uma AMI personalizada com o CodeDeploy Amazon EC2 Auto Scaling

Implantação de CodeDeploy aplicativos em grupos do Amazon EC2 Auto Scaling

Para implantar uma revisão CodeDeploy do aplicativo em um grupo do Amazon EC2 Auto Scaling:

1. Crie ou localize um perfil de instância do IAM que permita que o grupo do Amazon EC2 Auto Scaling funcione com o Amazon S3. Para obter mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.

   > **Note**
   >
   > Você também pode usar CodeDeploy para implantar revisões de GitHub repositórios para grupos do Amazon EC2 Auto Scaling. Embora as instâncias do Amazon EC2 ainda exijam um perfil de instância do IAM, o perfil não precisa de nenhuma permissão adicional para ser implantado a partir de um GitHub repositório.

2. Crie ou use um grupo do Amazon EC2 Auto Scaling, especificando o perfil de instância do IAM no modelo ou na configuração de execução. Para obter mais informações, consulte Perfil do IAM para aplicativos que são executados em instâncias do Amazon EC2.

3. Crie ou localize uma função de serviço que permita CodeDeploy criar um grupo de implantação que contenha o grupo Amazon EC2 Auto Scaling.
4. Crie um grupo de implantação com CodeDeploy, especificando o nome do grupo Amazon EC2 Auto Scaling, a função do serviço e algumas outras opções. Para acessar mais informações, consulte Criar um grupo de implantação para uma implantação no local (console) ou Criar um grupo de implantação para uma implantação no local (console).
5. Use CodeDeploy para implantar sua revisão no grupo de implantação que contém o grupo Amazon EC2 Auto Scaling.

Para obter mais informações, consulte Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling.

Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling

Uma implantação de encerramento é um tipo de CodeDeploy implantação que é ativada automaticamente quando ocorre um evento de expansão do Auto Scaling. CodeDeploy executa a implantação de encerramento logo antes do serviço Auto Scaling encerrar a instância. Durante uma implantação de encerramento, CodeDeploy não implanta nada. Em vez disso, ele gera eventos de ciclo de vida, que você pode fazer hook com seus próprios scripts para ativar a funcionalidade de encerramento personalizada. Por exemplo, você pode fazer hook do evento de ciclo de vida `ApplicationStop` em um script que encerra a aplicação, normalmente antes da instância ser encerrada.

Para obter uma lista dos eventos do ciclo de vida que são CodeDeploy gerados durante uma implantação de encerramento, consulte. Disponibilidade de hooks de eventos de ciclo de vida

Se a implantação de encerramento falhar por qualquer motivo, CodeDeploy permitirá que o encerramento da instância continue. Isso significa que a instância será encerrada mesmo que CodeDeploy não tenha executado o conjunto completo (ou nenhum) dos eventos do ciclo de vida até a conclusão.

Se você não habilitar implantações de encerramento, o serviço Auto Scaling ainda encerrará as instâncias do Amazon EC2 quando ocorrer CodeDeploy um evento de escalabilidade, mas não gerará eventos de ciclo de vida.

> **Note**
>
> Independentemente de você ativar ou não implantações de encerramento, se o serviço Auto Scaling encerrar uma instância do Amazon EC2 enquanto a implantação estiver em andamento, poderá ocorrer CodeDeploy uma condição de corrida entre os eventos do ciclo de vida gerados pelo Auto Scaling e os serviços. CodeDeploy Por exemplo, o evento de `Terminating` ciclo de vida (gerado pelo serviço Auto Scaling) pode substituir `ApplicationStart` o evento (gerado pela implantação). CodeDeploy Nesse cenário, você pode enfrentar uma falha no encerramento da instância do Amazon EC2 ou na CodeDeploy implantação.

Para permitir a execução CodeDeploy de implantações de terminação

- Marque a caixa de seleção Adicionar um hook de encerramento aos grupos do Auto Scaling ao criar ou atualizar o grupo de implantação. Para obter instruções, consulte Criar um grupo de implantação para uma implantação no local (console) ou Crie um grupo de implantação para uma EC2/On-Premises blue/green implantação (console).

  Ativar essa caixa de seleção faz CodeDeploy com que um gancho de ciclo de vida do Auto Scaling seja instalado nos grupos do Auto Scaling que você especifica ao criar ou atualizar seu grupo de implantação. CodeDeploy Esse hook é chamado de hook de encerramento e permite implantações de encerramento.

Depois que o hook de encerramento é instalado, um evento de redução da escala horizontal (encerramento) ocorre da seguinte forma:

1. O serviço de Auto Scaling (ou simplesmente o Auto Scaling) determina que uma redução da escala horizontal precisa ocorrer e entra em contato com o serviço EC2 para encerrar uma nova instância do EC2.
2. O serviço EC2 começa a encerrar a instância do EC2. A instância se move para o estado `Terminating` e depois para o estado `Terminating:Wait`.
3. Durante`Terminating:Wait`, o Auto Scaling executa todos os ganchos do ciclo de vida conectados ao grupo Auto Scaling, incluindo o gancho de terminação instalado pelo. CodeDeploy
4. O gancho de terminação envia uma notificação para a fila do Amazon SQS que é pesquisada por. CodeDeploy

5. Ao receber a notificação, CodeDeploy analisa a mensagem, realiza alguma validação e executa uma implantação de encerramento.
6. Enquanto a implantação de encerramento está em execução, CodeDeploy envia pulsações a cada cinco minutos para o Auto Scaling para que ele saiba que a instância ainda está sendo trabalhada.
7. Até o momento, a instância do EC2 ainda está no estado `Terminating:Wait` (ou possivelmente no estado `Warmed:Pending:Wait`, se você tiver ativado Pools de aquecimento de grupos do Auto Scaling).
8. Quando a implantação é concluída, CodeDeploy indica ao Auto Scaling `CONTINUE` o processo de encerramento do EC2, independentemente de a implantação de encerramento ter sido bem-sucedida ou falhada.

Como o Amazon EC2 Auto Scaling funciona com CodeDeploy

Quando você cria ou atualiza um grupo de CodeDeploy implantação para incluir um grupo do Auto Scaling, CodeDeploy acessa o grupo do Auto Scaling usando a função de CodeDeploy serviço e, em seguida, instala os ganchos do ciclo de vida do Auto Scaling nos grupos do Auto Scaling.

> **Note**
>
> Os ganchos de ciclo de vida do Auto Scaling são diferentes dos eventos de ciclo de vida (também chamados de ganchos de eventos de ciclo de vida) gerados e descritos neste guia. CodeDeploy AppSpec seção 'ganchos'

Os ganchos do ciclo de vida do Auto Scaling que são instalados são: CodeDeploy

- Um gancho de lançamento — Esse gancho notifica CodeDeploy que um evento de expansão do Auto Scaling está em andamento e CodeDeploy que precisa iniciar uma implantação de lançamento.

  Durante uma implantação de lançamento, CodeDeploy:

  - Implanta uma revisão da aplicação na instância com aumento da escala horizontal.
  - Gera eventos do ciclo de vida para indicar o progresso da implantação. Você pode fazer hook desses eventos do ciclo de vida em seus próprios scripts para ativar a funcionalidade de inicialização personalizada. Para obter mais informações, consulte a tabela em Disponibilidade de hooks de eventos de ciclo de vida.

O hook de inicialização e a implantação de inicialização associada estão sempre ativados e não podem ser desativados.

- Um gancho de terminação — Esse gancho opcional notifica CodeDeploy que um evento de escalonamento do Auto Scaling está em andamento e CodeDeploy que precisa iniciar uma implantação de terminação.

  Durante uma implantação de encerramento, CodeDeploy gera eventos de ciclo de vida para indicar o progresso do desligamento da instância. Para obter mais informações, consulte Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling.

Tópicos

- Depois de CodeDeploy instalar os ganchos de ciclo de vida, como eles são usados?
- Como CodeDeploy nomeia os grupos do Amazon EC2 Auto Scaling
- Ordem de execução de eventos de gancho do ciclo de vida personalizados
- Scale-out eventos durante uma implantação
- Scale-in eventos durante uma implantação
- Ordem dos eventos em AWS CloudFormation scripts cfn-init

Depois de CodeDeploy instalar os ganchos de ciclo de vida, como eles são usados?

Depois que os ganchos do ciclo de vida de lançamento e encerramento são instalados, eles são usados durante os eventos de expansão e CodeDeploy expansão de grupos do Auto Scaling, respectivamente.

Um evento de aumento da escala horizontal (inicialização) ocorre da seguinte forma:

1. O serviço de auste de escala automático (ou simplesmente ajuste de escala automático) determina que um evento de aumento de escala horizontal precisa ocorrer e entra em contato com o serviço EC2 para iniciar uma nova instância do EC2.
2. O serviço do EC2 inicia uma nova instância do EC2. A instância se move para o estado `Pending` e depois para o estado `Pending:Wait`.
3. Durante`Pending:Wait`, o Auto Scaling executa todos os ganchos do ciclo de vida conectados ao grupo Auto Scaling, incluindo o gancho de lançamento instalado pelo. CodeDeploy
4. O gancho de lançamento envia uma notificação para a fila do Amazon SQS que é pesquisada por. CodeDeploy

5. Ao receber a notificação, CodeDeploy analisa a mensagem, realiza alguma validação e inicia uma implantação de lançamento.
6. Enquanto a implantação de lançamento está em execução, CodeDeploy envia pulsações a cada cinco minutos para o Auto Scaling para que ele saiba que a instância ainda está sendo trabalhada.
7. Até o momento, a instância do EC2 ainda está no estado `Pending:Wait`.
8. Quando a implantação for concluída, CodeDeploy indica ao Auto Scaling `CONTINUE` um `ABANDON` ou o processo de inicialização do EC2, dependendo se a implantação foi bem-sucedida ou falhou.
   - Se CodeDeploy indicado`CONTINUE`, o Auto Scaling continua o processo de inicialização, aguardando a conclusão de outros ganchos ou colocando a instância no estado `Pending:Proceed` e depois no estado. `InService`
   - Se CodeDeploy indicado**`ABANDON`**, o Auto Scaling encerra a instância do EC2 e reinicia o procedimento de inicialização, se necessário, para atender ao número desejado de instâncias, conforme definido na configuração de Capacidade Desejada do Auto Scaling.

Um evento de redução da escala horizontal (encerramento) ocorre da seguinte forma:

Consulte Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling.

Como CodeDeploy nomeia os grupos do Amazon EC2 Auto Scaling

Durante blue/green as implantações em uma plataforma de EC2/On-Premises computação, você tem duas opções para adicionar instâncias ao seu ambiente substituto (verde):

- Use instâncias já existentes ou que você cria manualmente.
- Use as configurações de um grupo do Amazon EC2 Auto Scaling especificado para definir e criar instâncias em um novo grupo do Amazon EC2 Auto Scaling.

Se você escolher a segunda opção, CodeDeploy provisiona um novo grupo do Amazon EC2 Auto Scaling para você. Ele usa a seguinte convenção para nomear o grupo:

```
CodeDeploy_deployment_group_name_deployment_id
```

Por exemplo, se uma implantação com ID `10` implantar um grupo de implantação denominado `alpha-deployments`, o grupo provisionado do Amazon EC2 Auto Scaling se chamará `CodeDeploy_alpha-deployments_10`. Para obter mais informações, consulte Crie um

grupo de implantação para uma EC2/On-Premises blue/green implantação (console) e GreenFleetProvisioningOption.

Ordem de execução de eventos de gancho do ciclo de vida personalizados

Você pode adicionar seus próprios ganchos de ciclo de vida aos grupos do Amazon EC2 Auto Scaling nos quais são implantados. CodeDeploy No entanto, a ordem na qual esses eventos de gancho de ciclo de vida personalizados são executados não pode ser predeterminada em relação aos eventos CodeDeploy padrão do ciclo de vida de implantação. Por exemplo, se você adicionar um gancho de ciclo de vida personalizado chamado `ReadyForSoftwareInstall` a um grupo do Amazon EC2 Auto Scaling, não poderá saber de antemão se ele será executado antes do primeiro ou depois do último evento de ciclo de vida de implantação padrão. CodeDeploy

Para saber como adicionar ganchos do ciclo de vida personalizados em um grupo do Amazon EC2 Auto Scaling, consulte Adição de ganchos de ciclo de vida no Guia do usuário do Amazon EC2 Auto Scaling.

Scale-out eventos durante uma implantação

Se um evento de aumento da escala horizontal do Auto Scaling ocorrer enquanto uma implantação estiver em andamento, as novas instâncias serão atualizadas com a revisão da aplicação que foi implantada anteriormente, não com a revisão da aplicação mais recente. Se a implantação for bem-sucedida, as instâncias antigas e as instâncias recém-aumentadas em escala hospedarão diferentes revisões de aplicativos. Para atualizar as instâncias com a revisão mais antiga, inicia CodeDeploy automaticamente uma implantação subsequente (imediatamente após a primeira) para atualizar todas as instâncias desatualizadas. Se você quiser alterar esse comportamento padrão para que as instâncias desatualizadas do EC2 sejam deixadas na revisão mais antiga, consulte Automatic updates to outdated instances.

Se você quiser suspender os processos de escalabilidade horizontal do Amazon EC2 Auto Scaling enquanto as implantações estão ocorrendo, você pode fazer isso por meio de uma configuração no script que é usada para balanceamento de `common_functions.sh` carga com. CodeDeploy Se `HANDLE_PROCS=true`, os seguintes eventos do Auto Scaling serão suspensos automaticamente durante o processo de implantação:

- AZRebalance
- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

> **Important**
>
> Somente a configuração CodeDeployDefault.OneAtATime de implantação oferece suporte a essa funcionalidade.

Para obter mais informações sobre como usar `HANDLE_PROCS=true` para evitar problemas de implantação ao usar o Amazon EC2 Auto Scaling, consulte Aviso importante AutoScaling sobre como lidar com processos em aws-codedeploy-samples em. GitHub

Scale-in eventos durante uma implantação

Se um grupo de Auto Scaling começar a escalar enquanto uma CodeDeploy implantação estiver em andamento nesse grupo de Auto Scaling, poderá ocorrer uma condição de corrida entre o processo de encerramento (incluindo os eventos do ciclo de vida da implantação de CodeDeploy encerramento) e outros eventos do ciclo de vida na instância de encerramento. CodeDeploy A implantação nessa instância específica pode falhar se a instância for encerrada antes da conclusão de todos os eventos CodeDeploy do ciclo de vida. Além disso, a CodeDeploy implantação geral pode ou não falhar, dependendo de como você definiu sua configuração de mínimo de hosts íntegros em sua configuração de implantação.

Ordem dos eventos em AWS CloudFormation scripts cfn-init

Se você usa `cfn-init` (`oucloud-init`) para executar scripts em Linux-based instâncias recém-provisionadas, suas implantações podem falhar, a menos que você controle rigorosamente a ordem dos eventos que ocorrem após o início da instância.

Essa ordem deve ser:

1. A instância recém-provisionada é iniciada.
2. Todos os scripts de inicialização `cfn-init` são executados até a conclusão.
3. O CodeDeploy agente começa.
4. A última revisão de aplicativo é implantada na instância.

Se a ordem dos eventos não for cuidadosamente controlada, o CodeDeploy agente poderá iniciar uma implantação antes que todos os scripts terminem de ser executados.

Para controlar a ordem dos eventos, use uma destas práticas recomendadas:

- Instale o CodeDeploy agente por meio de um `cfn-init` script, colocando-o depois de todos os outros scripts.
- Inclua o CodeDeploy agente em uma AMI personalizada e use um `cfn-init` script para iniciá-la, colocando-a depois de todos os outros scripts.

Para obter mais informações sobre o uso do `cfn-init`, consulte cfn-init no Guia do usuário do AWS CloudFormation .

Usando uma AMI personalizada com o CodeDeploy Amazon EC2 Auto Scaling

Existem duas opções para especificar a AMI base a ser usada quando novas instâncias do Amazon EC2 são iniciadas em um grupo do Amazon EC2 Auto Scaling:

- Você pode especificar uma AMI personalizada básica que já tenha o CodeDeploy agente instalado. Como o agente já está instalado, essa opção inicia novas instâncias do Amazon EC2 com mais rapidez do que a outra opção. No entanto, essa opção oferece uma maior probabilidade de que as implantações iniciais das instâncias do Amazon EC2 falhem, especialmente se CodeDeploy o agente estiver desatualizado. Se você escolher essa opção, recomendamos que você atualize regularmente o CodeDeploy agente em sua AMI personalizada básica.
- Você pode especificar uma AMI básica que não tenha o CodeDeploy agente instalado e tenha o agente instalado à medida que cada nova instância é executada em um grupo do Amazon EC2 Auto Scaling. Embora essa opção inicie novas instâncias do Amazon EC2 mais devagar do que a outra opção, com ela, são maiores as chances de que as implantações iniciais das instâncias sejam bem-sucedidas. Essa opção usa a versão mais recente do agente do CodeDeploy.

Integração CodeDeploy com o Elastic Load Balancing

Durante CodeDeploy as implantações, um balanceador de carga impede que o tráfego da Internet seja roteado para instâncias quando elas não estão prontas, estão sendo implantadas no momento ou não são mais necessárias como parte de um ambiente. A função exata que o balanceador de carga desempenha, no entanto, depende se ele é usado em uma blue/green implantação ou em uma implantação local.

> **Note**
>
> O uso de balanceadores de carga do Elastic Load Balancing é obrigatório nas blue/green implantações e opcional nas implantações locais.

Tipos de Elastic Load Balancing

O Elastic Load Balancing fornece três tipos de balanceadores de carga que podem ser usados em CodeDeploy implantações: Classic Load Balancers, Application Load Balancers e Network Load Balancers.

Classic Load Balancer

Rotas e balanceamentos de carga na camada de transporte (TCP/SSL) ou na camada de aplicação (HTTP/HTTPS). Ele oferece suporte a uma VPC.

> **Note**
>
> Os Classic Load Balancers não são compatíveis com implantações do Amazon ECS.

Application Load Balancer

Roteia e balanceia a carga na camada do aplicativo (HTTP/HTTPS) e oferece suporte ao roteamento baseado em caminhos. Ele pode encaminhar solicitações para portas em cada instância do EC2 ou instância de contêiner na sua nuvem privada virtual (VPC).

> **Note**
>
> Os grupos de destino do Application Load Balancer devem ter um tipo de destino `instance` para implantações em instâncias do EC2 e `IP` para implantações do Fargate. Para obter mais informações, consulte Tipo de destino.

Network Load Balancer

Rotas e balanceamentos de carga na camada de transporte (TCP/UDP Layer-4) com base nas informações de endereço extraídas do cabeçalho do pacote TCP, não do conteúdo do pacote. Os

balanceadores de carga de rede podem lidar com picos de tráfego, reter o IP de origem do cliente e usar um IP fixo para a vida útil do balanceador de carga.

Para saber mais sobre os load balancers do Elastic Load Balancing, consulte os seguintes tópicos:

- O que é Elastic Load Balancing?
- O que é um Classic Load Balancer?
- O que é um Application Load Balancer?
- O que é um Network Load Balancer?

Blue/Green implantações

O redirecionamento do tráfego da instância por trás de um load balancer do Elastic Load Balancing é fundamental para as implantações. CodeDeploy blue/green

Durante uma blue/green implantação, o balanceador de carga permite que o tráfego seja roteado para as novas instâncias em um grupo de implantação no qual a revisão mais recente do aplicativo foi implantada (o ambiente substituto), de acordo com as regras especificadas, e então bloqueia o tráfego das instâncias antigas em que a revisão anterior do aplicativo estava sendo executada (o ambiente original).

Após as instâncias em um ambiente de substituição serem registradas em um ou mais balanceadores de carga, o registro das instâncias do ambiente original é cancelado e, se você escolher, as instâncias são encerradas.

Para uma blue/green implantação, você pode especificar um ou mais grupos-alvo do Classic Load Balancers, do Application Load Balancer ou do Network Load Balancer no seu grupo de implantação. Você usa o CodeDeploy console ou adiciona AWS CLI os balanceadores de carga a um grupo de implantação.

Para obter mais informações sobre balanceadores de carga em blue/green implantações, consulte os tópicos a seguir:

- Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2
- Criar um aplicativo para uma blue/green implantação (console)
- Crie um grupo de implantação para uma EC2/On-Premises blue/green implantação (console)

In-place implantações

Em uma implantação no local, um balanceador de carga impede que o tráfego da Internet seja direcionado para uma instância durante o processo, tornando-a novamente disponível para tráfego assim que essa implantação é concluída.

Se o load balancer não for usado durante uma implantação, o tráfego da Internet ainda poderá ser direcionado para uma instância durante o processo de implantação. Como resultado, seus clientes poderão encontrar aplicativos web quebrados, incompletos ou desatualizados. Quando você usa um load balancer do Elastic Load Balancing com uma implantação local, as instâncias em um grupo de implantação são canceladas do load balancer, atualizadas com a revisão mais recente do aplicativo e, em seguida, registradas novamente no load balancer como parte do mesmo grupo de implantação após a implantação ser bem-sucedida. CodeDeploy aguardará até 1 hora para que a instância fique íntegra por trás do balanceador de carga. Se a instância não for marcada como íntegra pelo balanceador de carga durante o período de espera, CodeDeploy ela passa para a próxima instância ou falha na implantação, com base na configuração de implantação.

Para uma implantação no local, você pode especificar um ou mais grupos de destino do Classic Load Balancers, do Application Load Balancer ou do Network Load Balancer. Você pode especificar os balanceadores de carga como parte da configuração do grupo de implantação ou usar um script fornecido por CodeDeploy para implementar os balanceadores de carga.

Especificar o balanceador de carga de implantação no local usando um grupo de implantação

Para adicionar balanceadores de carga a um grupo de implantação, você usa o CodeDeploy console ou AWS CLI. Para obter informações sobre como especificar um load balancer em um grupo de implantação para as implantações no local, consulte os seguintes tópicos:

- Criar um aplicativo para uma implantação no local (console)
- Criar um grupo de implantação para uma implantação no local (console)
- Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2

Especificar o balanceador de carga de implantação no local usando um script

Use as etapas do procedimento a seguir para usar scripts de ciclo de vida de implantação para configurar o balanceamento de carga para implantações no local.

> **Note**
>
> Você deve usar a configuração de CodeDeployDefault.OneAtATime implantação somente quando estiver usando um script para configurar um balanceador de carga para uma implantação local. Não há suporte para execuções simultâneas, e a CodeDeployDefault.OneAtATime configuração garante uma execução serial dos scripts. Para obter mais informações sobre configurações de implantação, consulte Trabalhando com configurações de implantação em CodeDeploy.

No repositório CodeDeploy Samples em GitHub, fornecemos instruções e amostras que você pode adaptar para usar os balanceadores de carga do CodeDeploy Elastic Load Balancing. Esses repositórios incluem três scripts de amostra — `register_with_elb.sh`, `deregister_from_elb.sh` e `common_functions.sh` — que fornecem todo o código necessário para você poder trabalhar. Basta editar os espaços reservados nesses três scripts e fazer referência a eles no seu arquivo `appspec.yml`.

Para configurar implantações no local CodeDeploy com instâncias do Amazon EC2 registradas nos balanceadores de carga do Elastic Load Balancing, faça o seguinte:

1. Baixe as amostras para o tipo de balanceador de carga que deseja usar para uma implantação no local:
   - Classic Load Balancer
   - Application Load Balancer ou Network Load Balancer (o mesmo script pode ser usado para ambos os tipos)
2. Certifique-se de que cada uma de suas instâncias de destino do Amazon EC2 tenha o AWS CLI instalado.
3. Certifique-se de cada uma das suas instâncias de destino do Amazon EC2 tenha um perfil de instância do IAM conectado pelo menos com as permissões elasticloadbalancing:* e autoscaling:*.
4. Inclua no diretório do código-fonte do seu aplicativo os scripts de evento de ciclo de vida de implantação (`register_with_elb.sh`, `deregister_from_elb.sh` e `common_functions.sh`).
5. Em `appspec.yml` Para a revisão do aplicativo, forneça instruções CodeDeploy para executar o `register_with_elb.sh` script durante o ApplicationStartevento e o `deregister_from_elb.sh` script durante o ApplicationStopevento.

6. Se a instância fizer parte de um grupo do Amazon EC2 Auto Scaling, você poderá ignorar essa etapa.

   No script`common_functions.sh`:

   - Se estiver usando o Classic Load Balancer, especifique os nomes dos balanceadores de carga do Elastic Load Balancing em `ELB_LIST=""` e faça as alterações necessárias nas outras configurações de implantação no arquivo.
   - Se estiver usando o Application Load Balancer ou o Network Load Balancer, especifique os nomes dos grupos de destino do Elastic Load Balancing em `TARGET_GROUP_LIST=""` e faça as alterações necessárias nas outras configurações de implantação no arquivo.
7. Forneça o código-fonte do seu aplicativo, o `appspec.yml` e os scripts de evento de ciclo de vida de implantação em uma revisão de aplicativo e faça o upload da revisão. Implante a revisão nas instâncias do Amazon EC2. Durante a implantação, os scripts de evento de ciclo de vida de implantação cancelarão o registro da instância do Amazon EC2 no balanceador de carga, aguardará a drenagem da conexão e depois registrará novamente a instância do Amazon EC2 no balanceador de carga após a conclusão da implantação.

Integração com produtos e serviços de parceiros

CodeDeploy tem integração integrada para os seguintes produtos e serviços de parceiros:

Ansible	Se você já tem um conjunto de manuais do Ansible, mas só precisa de um lugar para executá-los, veja o modelo do Ansible e CodeDeploy demonstra como alguns ganchos de implantação simples podem garantir que o Ansible esteja disponível na instância de implantação local e execute os playbooks. Se você já tem um processo para criar e manter seu inventário, também há um módulo do Ansible que você pode usar para instalar e executar o CodeDeploy agente. Saiba mais: • Ansible e CodeDeploy

Atlassian – Bamboo e Bitbucket	A CodeDeploy tarefa do Bamboo compacta o diretório que contém um AppSpec arquivo em um arquivo.zip, carrega o arquivo no Amazon S3 e, em seguida, inicia a implantação de acordo com a configuração fornecida no aplicativo. CodeDeploy O suporte do Atlassian Bitbucket CodeDeploy permite que você envie código para instâncias do Amazon EC2 diretamente da interface do usuário do Bitbucket, sob demanda, para qualquer um dos seus grupos de implantação. Isso significa que, depois de atualizar o código no seu repositório do Bitbucket, você não precisa fazer login na sua plataforma de integração contínua (CI) ou em instâncias do Amazon EC2 para executar um processo de implantação manual. Saiba mais: • Usando a CodeDeploy tarefa para o Bamboo • Anunciando o suporte do Atlassian Bitbucket para CodeDeploy

Chef	AWS fornece dois exemplos de modelos para integrar o Chef e. CodeDeploy O primeiro é um livro de receitas do Chef que instala e inicia o CodeDeploy agente. Isso permite que você continue gerenciando sua infraestrutura de host com o Chef enquanto usa o CodeDeploy. O segundo modelo de amostra demonstra como usar para orquestrar CodeDeploy a execução de livros de receitas e receitas com chef-solo em cada nó. Saiba mais: • Chef e CodeDeploy
CircleCI	O CircleCI fornece um conjunto de ferrament as de teste e integração contínua e implantaç ão automatizada. Depois de criar uma função do IAM AWS para usar com o CircleCI e configurar seus parâmetros de implantação em seu arquivo circle.yml, você pode usar o CodeDeploy CircleCI with para criar revisões de aplicativos, carregá-las em um bucket do Amazon S3 e, em seguida, iniciar e monitorar suas implantações. Saiba mais: • Usando um CircleCI Orb para implantar aplicativos em AWS CodeDeploy

CloudBees	Você pode usar o plug-in CodeDeploy Jenkins, disponível no CloudBeesDEV @cloud, como uma ação pós-compilação. Por exemplo, no final de um pipeline de entrega contínua, você pode usá-lo para implantar uma revisão de aplicativo na sua frota de servidores. Saiba mais: • CodeDeploy Plugin Jenkins agora disponível no DEV @cloud
Codeship	Você pode usar o Codeship para implantar revisões de aplicativos por meio de. CodeDeploy Você pode usar a UI do Codeship para adicionar o CodeDeploy a um pipeline de implantação para uma ramificação. Saiba mais: • Implemente em CodeDeploy • CodeDeploy integração no Codeship
GitHub	Você pode usar CodeDeploy para implantar revisões de aplicativos a partir de GitHubrepositórios. Você também pode acionar uma implantação a partir de um GitHub repositório sempre que o código-fonte desse repositório for alterado. Saiba mais: • Integrando com CodeDeploy GitHub • Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub • Implemente automaticamente a partir do GitHub uso CodeDeploy

HashiCorp Cônsul	Você pode usar a ferramenta HashiCorp Consul de código aberto para ajudar a garantir a integridade e a estabilidade do seu ambiente de aplicativos ao implantar aplicativos em. CodeDeploy Você pode usar o Consul para registrar aplicativos que serão descobertos durante a implantação, colocar aplicativos e nós no modo de manutenção para omiti-los de implantações e interromper implantações se as instâncias de destino se tornarem não íntegras. Saiba mais: • CodeDeploy implantações com Consul HashiCorp
Jenkins	O plug-in CodeDeploy Jenkins fornece uma etapa de pós-construção para seu projeto Jenkins. Após uma compilação bem-sucedida, ele compacta o espaço de trabalho, faz upload no Amazon S3 e inicia uma nova implantação. Saiba mais: • CodeDeployPlug-in Jenkins • Configurando o plug-in Jenkins para CodeDeploy

Puppet Labs	AWS fornece modelos de amostra para Puppet e. CodeDeploy O primeiro é um módulo Puppet que instala e inicia o CodeDeploy agente. Isso permite que você continue gerencian do sua infraestrutura de host com o Puppet enquanto usa o CodeDeploy. O segundo modelo de amostra demonstra como usar para orquestrar CodeDeploy a execução de módulos e manifestos com uma marionete sem mestre em cada nó. Saiba mais: • Marionete e CodeDeploy
SaltStack	Você pode integrar a SaltStackinfraestr utura com CodeDeploy. Você pode usar o CodeDeploy módulo para instalar e executar o CodeDeploy agente em seus minions ou, com alguns ganchos de implantação simples, você pode usar CodeDeploy para orquestrar a execução de seus Salt States. Saiba mais: • SaltStack and CodeDeploy

TeamCity	Você pode usar o plug-in CodeDeploy Runner para implantar aplicativos diretamente do TeamCity. O plug-in adiciona uma etapa de TeamCity construção que prepara e carrega uma revisão do aplicativo em um bucket do Amazon S3, registra a revisão em CodeDeploy um aplicativo, cria CodeDeploy uma implantaç ão e, se você escolher, espera que a implantaç ão seja concluída. Saiba mais: • CodeDeploy Runner (Baixar) • CodeDeploy Plugin Runner (Documentação)
Travis CI	Você pode configurar o Travis CI para acionar uma implantação CodeDeploy após uma compilação bem-sucedida. Saiba mais: • Travis CI e implantações CodeDeploy

Tópicos

- Integrando com CodeDeploy GitHub

Integrando com CodeDeploy GitHub

CodeDeploy supports GitHub, um serviço de hospedagem e compartilhamento de código baseado na web. CodeDeploy pode implantar revisões de aplicativos armazenadas em GitHub repositórios ou buckets do Amazon S3 em instâncias. CodeDeploy suporta somente GitHub para EC2/On-Premises implantações.

Tópicos

- Implantando CodeDeploy revisões de GitHub
- GitHub comportamentos com CodeDeploy

Implantando CodeDeploy revisões de GitHub

Para implantar uma revisão do aplicativo de um GitHub repositório para instâncias:

1. Crie uma revisão que seja compatível com CodeDeploy o tipo de instância do Amazon EC2 na qual você implantará.

   Para criar uma revisão compatível, siga as instruções em Planeje uma revisão para CodeDeploy e Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.
2. Use uma GitHub conta para adicionar sua revisão a um GitHub repositório.

   Para criar uma GitHub conta, consulte Entrar GitHub. Para criar um GitHub repositório, consulte Criar um repositório.
3. Use a página Criar implantação no CodeDeploy console ou o AWS CLI create-deployment comando para implantar sua revisão do seu GitHub repositório para instâncias de destino configuradas para uso em CodeDeploy implantações.

   Se quiser chamar o create-deployment comando, você deve primeiro usar a página Criar implantação do console para dar CodeDeploy permissão para interagir GitHub em nome de sua GitHub conta preferida para o aplicativo especificado. Você só precisa fazer isso uma vez por aplicativo.

   Para saber como usar a página Criar implantação para implantar a partir de um GitHub repositório, consulteCrie uma implantação com CodeDeploy.

   Para saber como chamar o create-deployment comando para implantar a partir de um GitHub repositório, consulteCrie uma implantação da EC2/On-Premises Compute Platform (CLI).

   Para saber como preparar instâncias para uso em CodeDeploy implantações, consulteTrabalhando com instâncias para CodeDeploy.

Para obter mais informações, consulte Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub.

GitHub comportamentos com CodeDeploy

Tópicos

- GitHub autenticação com aplicativos em CodeDeploy
- CodeDeploy interação com GitHub repositórios públicos e privados

- CodeDeploy interação com repositórios gerenciados pela organização GitHub
- Implemente automaticamente a partir de CodePipeline com CodeDeploy

GitHub autenticação com aplicativos em CodeDeploy

Depois de dar CodeDeploy permissão para interagir com GitHub, a associação entre essa GitHub conta e o aplicativo é armazenada em CodeDeploy. Você pode vincular o aplicativo a uma GitHub conta diferente. Você também pode revogar a permissão CodeDeploy para interagir com GitHub.

Para vincular uma GitHub conta a um aplicativo no CodeDeploy

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
3. Escolha o aplicativo que você deseja vincular a uma GitHub conta diferente.
4. Se o aplicativo não tiver um grupo de implantação, selecione Criar grupo de implantação para criar um. Para obter mais informações, consulte Crie um grupo de implantação com CodeDeploy. Um grupo de implantação é necessário para selecionar Create deployment (Criar implantação) na próxima etapa.
5. Em Deployments (Implantações), selecione Create deployment (Criar implantação).

   > **Note**
   >
   > Não é necessário criar uma nova implantação. Atualmente, essa é a única maneira de vincular uma GitHub conta diferente a um aplicativo.

6. Em Configurações de implantação, para Tipo de revisão, escolha Meu aplicativo está armazenado em GitHub.
7. Execute um destes procedimentos:

   - Para criar uma conexão entre AWS CodeDeploy aplicativos e uma GitHub conta, saia GitHub em uma guia separada do navegador. Em nome do GitHub token, digite um nome para

identificar essa conexão e escolha Conectar GitHub a. A página da web solicita que você autorize CodeDeploy a interação com seu GitHub aplicativo. Continue na etapa 10.

- Para usar uma conexão que você já criou, em nome do GitHub token, selecione seu nome e escolha Conectar GitHub a. Continue na etapa 8.
- Para criar uma conexão com uma GitHub conta diferente, saia GitHub em uma guia separada do navegador. Em Nome do GitHub token, digite um nome para identificar a conexão e escolha Conectar GitHub a. A página da web solicita que você autorize CodeDeploy a interação com seu GitHub aplicativo. Continue na etapa 10.

8. Se você ainda não estiver conectado GitHub, siga as instruções na página de login para fazer login com a GitHub conta à qual você deseja vincular o aplicativo.
9. Escolha Autorizar aplicativo. GitHub dá CodeDeploy permissão para interagir GitHub em nome da GitHub conta conectada do aplicativo selecionado.
10. Se não quiser criar uma implantação, escolha Cancelar.

Para revogar a permissão CodeDeploy para interagir com GitHub

1. Faça login GitHub usando as credenciais da GitHub conta na qual você deseja revogar a permissão AWS CodeDeploy .
2. Abra a página GitHub Aplicativos, localize CodeDeployna lista de aplicativos autorizados e siga o GitHub procedimento para revogar a autorização de um aplicativo.

CodeDeploy interação com GitHub repositórios públicos e privados

CodeDeploy suporta a implantação de aplicativos de GitHub repositórios públicos e privados. Quando você der CodeDeploy permissão para acessar GitHub em seu nome, CodeDeploy terá acesso de leitura e gravação a todos os GitHub repositórios privados aos quais sua GitHub conta tem acesso. No entanto, CodeDeploy só lê dos GitHub repositórios. Ele não gravará em nenhum dos seus GitHub repositórios privados.

CodeDeploy interação com repositórios gerenciados pela organização GitHub

Por padrão, GitHub os repositórios gerenciados por uma organização (em oposição aos repositórios privados ou públicos da sua conta) não concedem acesso a aplicativos de terceiros, inclusive. CodeDeploy Sua implantação falhará se as restrições de aplicativos de terceiros de uma organização estiverem habilitadas GitHub e você tentar implantar o código de seu GitHub repositório. Há duas maneiras de resolver esse problema.

- Como membro da organização, você pode solicitar ao proprietário da organização que aprove o acesso ao CodeDeploy. As etapas para solicitar esse acesso dependem de você já ter autorizado CodeDeploy sua conta individual:
  - Se você autorizou o acesso à CodeDeploy sua conta, consulte Solicitando a aprovação da organização para suas inscrições autorizadas.
  - Se você ainda não autorizou o acesso à CodeDeploy sua conta, consulte Solicitação de aprovação da organização para aplicativos de terceiros.
- O proprietário da organização pode desabilitar todas as restrições de aplicativos de terceiros para a organização. Para obter informações, consulte Desabilitando restrições de aplicativos de terceiros para sua organização.

Para obter mais informações, consulte Sobre restrições de aplicativos de terceiros.

Implemente automaticamente a partir de CodePipeline com CodeDeploy

Você pode acionar uma implantação a partir de um CodePipeline sempre que o código-fonte for alterado. Para obter mais informações, consulte CodePipeline.

Exemplos de integração da comunidade

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

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

Publicações no blog

- Automatizando o provisionamento em CodeDeploy CloudFormation

  Saiba como provisionar a implantação de um aplicativo CodeDeploy usando CloudFormation.

  Publicado em janeiro de 2016

- AWS Toolkit for Eclipse Integração com CodeDeploy (Parte 1)

  AWS Toolkit for Eclipse Integração com CodeDeploy (Parte 2)

  AWS Toolkit for Eclipse Integração com CodeDeploy (Parte 3)

  Saiba como os desenvolvedores Java podem usar o CodeDeploy plug-in do Eclipse para implantar aplicativos web AWS diretamente dos ambientes de desenvolvimento do Eclipse.

  Publicado em fevereiro de 2015
- Implemente automaticamente a partir do GitHub uso CodeDeploy

  Saiba como as implantações automáticas de GitHub até CodeDeploy podem ser usadas para criar um pipeline de ponta a ponta, do controle de origem aos ambientes de teste ou produção.

  Publicado em dezembro de 2014

CodeDeploy tutoriais

Esta seção inclui alguns tutoriais para ajudá-lo a aprender a usar o CodeDeploy.

Os procedimentos desses tutoriais fornecem sugestões sobre o local no qual armazenar arquivos (por exemplo, c:\temp) e os nomes a serem dados aos buckets, subpastas ou arquivos (por exemplo, amzn-s3-demo-bucket e CodeDeployDemo-EC2-Trust.json, respectivamente) HelloWorldApp, mas você não precisa usá-los. Certifique-se de substituir os nomes e locais dos seus arquivos enquanto realiza os procedimentos.

Tópicos

- Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix)
- Tutorial: implantar um aplicativo "Hello, World!" aplicativo com CodeDeploy (Windows Server)
- Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux)
- Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling
- Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub
- Tutorial: Implantar um aplicativo no Amazon ECS
- Tutorial: Implantar um serviço do Amazon ECS com um teste de validação
- Tutorial: Implemente uma função Lambda atualizada com CodeDeploy e o AWS Modelo de aplicativo sem servidores

Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix)

Neste tutorial, você implanta WordPress uma ferramenta de blog e um sistema de gerenciamento de conteúdo de código aberto baseados em PHP e MySQL em uma única instância do Amazon EC2 executando o Amazon Linux ou o Red Hat Enterprise Linux (RHEL).

Não é o que você está procurando?

- Para praticar a implantação em uma instância do Amazon EC2 que executa o Windows Server, consulte Tutorial: implantar um aplicativo "Hello, World!" aplicativo com CodeDeploy (Windows Server).
- Para praticar a implantação em uma instância on-premises, em vez de uma instância do Amazon EC2, consulte Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).

As etapas deste tutorial são apresentadas sob a perspectiva de uma máquina de desenvolvimento local executando o Linux, macOS ou Unix. Embora possa concluir a maioria dessas etapas em uma máquina local executando o Windows, você precisa adaptar as etapas que fazem referência a comandos, como chmod e wget, a aplicativos, como o sed, e a caminhos de diretório, como `/tmp`.

Antes de começar este tutorial, você deve atender aos pré-requisitos em Começando com CodeDeploy. Isso inclui configurar um usuário, instalar ou atualizar o AWS CLI e criar um perfil de instância do IAM e uma função de serviço.

Tópicos

- Etapa 1: Iniciar e configurar uma instância Amazon Linux ou Red Hat Enterprise Linux Amazon EC2
- Etapa 2: configurar seu conteúdo de origem para ser implantado na instância Amazon Linux ou Red Hat Enterprise Linux instância do Amazon EC2
- Etapa 3: Faça o upload do seu WordPress aplicativo para o Amazon S3
- Etapa 4: implantar seu WordPress aplicativo
- Etapa 5: atualizar e reimplantar seu aplicativo WordPress
- Etapa 6: Limpe seu WordPress aplicativo e os recursos relacionados

Etapa 1: Iniciar e configurar uma instância Amazon Linux ou Red Hat Enterprise Linux Amazon EC2

Para implantar o WordPress aplicativo CodeDeploy, você precisará de uma instância do Amazon EC2 executando o Amazon Linux ou o Red Hat Enterprise Linux (RHEL). A instância do Amazon EC2 requer uma nova regra de segurança de entrada que permita conexões HTTP. Essa regra é necessária para visualizar a WordPress página em um navegador após sua implantação bem-sucedida.

Siga as instruções em Crie uma instância do Amazon EC2 para CodeDeploy. Quando você chegar na parte dessas instruções sobre como atribuir uma tag de instância do Amazon EC2 à instância, certifique-se de especificar a chave da tag de **`Name`** e o valor da tag de **`CodeDeployDemo`**. (Se você especificar uma chave de tag ou um valor de tag diferente, as instruções em Etapa 4: implantar seu WordPress aplicativo poderão produzir resultados inesperados.)

Depois de seguir as instruções para iniciar a instância do Amazon EC2, retorne a essa página e continue na próxima seção. Não continue em Crie um aplicativo com CodeDeploy como a próxima etapa.

Conectar-se à instância do Amazon Linux ou RHEL Amazon EC2

Depois que a sua nova instância do Amazon EC2 for iniciada, siga essas instruções para praticar a conexão com ela.

1. Use o ssh comando (ou um emulador de SSH-capable terminal como PuTTY) para se conectar à sua instância Amazon Linux ou RHEL do Amazon EC2. Você precisará do endereço DNS público da instância e da chave privada para o par de chaves usado na inicialização da instância do Amazon EC2. Para obter mais informações, consulte Conectar-se à sua instância.

   Por exemplo, se o endereço DNS público for **`ec2-01-234-567-890.compute-1.amazonaws.com`**, e o par de chaves da sua instância do Amazon EC2 para acesso SSH for denominado **`codedeploydemo.pem`**, você poderá digitar:

   ```
   ssh -i /path/to/codedeploydemo.pem ec2-
   user@ec2-01-234-567-890.compute-1.amazonaws.com
   ```

   Substitua */path/to/*`codedeploydemo.pem` pelo caminho para o seu arquivo `.pem` e substitua o endereço DNS de exemplo pelo endereço de sua instância do Amazon EC2 do Amazon Linux ou RHEL.

   > **Note**
   >
   > Se receber um erro de que as permissões do seu arquivo de chave são muito abertas, será necessário restringir suas permissões para dar acesso apenas ao usuário atual (você). Por exemplo, com o comando chmod no Linux, macOS ou Unix, digite:

```
chmod 400 /path/to/codedeploydemo.pem
```

2. Depois de entrar, você verá o banner da AMI da instância do Amazon EC2. Para o Amazon Linux, deve ficar assim:

```
       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|
```

3. Agora, você pode sair da instância do Amazon EC2 em execução.

> **Warning**
>
> Não interrompa ou encerre a instância do Amazon EC2. Caso contrário, CodeDeploy não será possível implantá-lo.

Adicione uma regra de entrada que permita tráfego HTTP à instância do Amazon EC2 do Amazon Linux ou do RHEL

A próxima etapa confirma que sua instância do Amazon EC2 tem uma porta HTTP aberta para que você possa ver a página inicial do aplicativo WordPress implantado em um navegador.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. Escolha Instâncias e, em seguida, escolha sua instância.
3. Na guia Descrição, em Grupos de segurança, escolha exibir regras de entrada.

   Você deve ver uma lista de regras em seu grupo de segurança como a seguinte:

```
Security Groups associated with i-1234567890abcdef0
 Ports     Protocol     Source     launch-wizard-N
 22        tcp          0.0.0.0/0          #
```

4. Em Grupos de segurança, escolha o grupo de segurança para a sua instância do Amazon EC2. Ele pode se chamar **`launch-wizard-`*`N`***. O ***`N`*** no nome é um número atribuído ao grupo de segurança quando a instância foi criada.

   Escolha a guia Entrada. Se o grupo de segurança da sua instância está configurado corretamente, você deve ver uma regra com os seguintes valores:

   - Tipo: HTTP
   - Protocolo: TCP
   - Port Range: 80
   - Fonte: 0.0.0. 0/0

5. Se você não encontrar uma regra com esses valores, use os procedimentos em Adicionar regras a um grupo de segurança para adicioná-las a uma nova regra de segurança.

Etapa 2: configurar seu conteúdo de origem para ser implantado na instância Amazon Linux ou Red Hat Enterprise Linux instância do Amazon EC2

Agora é hora de configurar o conteúdo de origem do seu aplicativo, para que você tenha algo para implantar na instância.

Tópicos

- Obter o código-fonte
- Criar scripts para executar seu aplicativo
- Adicionar um arquivo de especificação de aplicativo

Obter o código-fonte

Neste tutorial, você implanta a plataforma de publicação de WordPress conteúdo da sua máquina de desenvolvimento na instância de destino do Amazon EC2. Para obter o WordPress código-fonte, você pode usar chamadas de linha de comando integradas. Ou, se tiver o Git instalado na sua máquina de desenvolvimento, poderá usá-lo como alternativa.

Para essas etapas, presumimos que você baixou uma cópia do WordPress código-fonte para o `/tmp` diretório em sua máquina de desenvolvimento. (É possível escolher qualquer diretório desejado,

mas lembre-se de substituir `/tmp` pela localização sempre que esse diretório for especificado nas etapas.)

Escolha uma das duas opções a seguir para copiar os arquivos de WordPress origem para sua máquina de desenvolvimento. A primeira opção usa chamadas integradas de linha de comando. A segunda opção usa o Git.

Tópicos

- Para obter uma cópia do WordPress código-fonte (chamadas de linha de comando integradas)
- Para obter uma cópia do WordPress código-fonte (Git)

Para obter uma cópia do WordPress código-fonte (chamadas de linha de comando integradas)

1. Chame o wget comando para baixar uma cópia do WordPress código-fonte, como um arquivo.zip, para o diretório atual:

   ```
   wget https://github.com/WordPress/WordPress/archive/master.zip
   ```

2. Chame os comandos unzip, mkdir, cp e rm para:

   - Descompacte o `master.zip` arquivo no `/tmp/WordPress_Temp` diretório (pasta).
   - Copiar seu conteúdo descompactado na pasta de destino `/tmp/WordPress`.
   - Excluir a pasta temporária `/tmp/WordPress_Temp` e o arquivo `master`.

   Execute os comandos um de cada vez:

   ```
   unzip master -d /tmp/WordPress_Temp
   ```

   ```
   mkdir -p /tmp/WordPress
   ```

   ```
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   ```

   ```
   rm -rf /tmp/WordPress_Temp
   ```

   ```
   rm -f master
   ```

Isso deixa você com um conjunto limpo de arquivos de WordPress código-fonte na `/tmp/WordPress` pasta.

Para obter uma cópia do WordPress código-fonte (Git)

1. Baixe e instale o Git na sua máquina de desenvolvimento.
2. Na pasta `/tmp/WordPress`, chame o comando git init.
3. Chame o git clone comando para clonar o WordPress repositório público, fazendo sua própria cópia dele na pasta de `/tmp/WordPress` destino:

   ```
   git clone https://github.com/WordPress/WordPress.git /tmp/WordPress
   ```

   Isso deixa você com um conjunto limpo de arquivos de WordPress código-fonte na `/tmp/WordPress` pasta.

Criar scripts para executar seu aplicativo

Em seguida, crie uma pasta e scripts no diretório. CodeDeploy usa esses scripts para configurar e implantar a revisão do seu aplicativo na instância de destino do Amazon EC2. É possível usar qualquer editor de texto para criar os scripts.

1. Crie um diretório de scripts em sua cópia do WordPress código-fonte:

   ```
   mkdir -p /tmp/WordPress/scripts
   ```

2. Crie um arquivo `install_dependencies.sh` no `/tmp/WordPress/scripts`. Adicione as seguintes linhas ao arquivo. Este script `install_dependencies.sh` instala Apache, MySQL e PHP. Ele também adiciona suporte do MySQL ao PHP.

   ```
   #!/bin/bash
   sudo amazon-linux-extras install php7.4
   sudo yum install -y httpd mariadb-server php
   ```

3. Crie um arquivo `start_server.sh` no `/tmp/WordPress/scripts`. Adicione as seguintes linhas ao arquivo. Este script `start_server.sh` inicia Apache e MySQL.

```
#!/bin/bash
systemctl start mariadb.service
systemctl start httpd.service
systemctl start php-fpm.service
```

4. Crie um arquivo `stop_server.sh` no `/tmp/WordPress/scripts`. Adicione as seguintes linhas ao arquivo. Este script `stop_server.sh` para Apache e MySQL.

```
#!/bin/bash
isExistApp="pgrep httpd"
if [[ -n $isExistApp ]]; then
systemctl stop httpd.service
fi
isExistApp=pgrep mysqld
if [[ -n $isExistApp ]]; then
systemctl stop mariadb.service
fi
isExistApp=pgrep php-fpm
if [[ -n $isExistApp ]]; then
systemctl stop php-fpm.service

fi
```

5. Crie um arquivo `create_test_db.sh` no `/tmp/WordPress/scripts`. Adicione as seguintes linhas ao arquivo. Esse `create_test_db.sh` script usa o MySQL para criar um **`test`** banco de dados WordPress para uso.

```
#!/bin/bash
mysql -uroot <<CREATE_TEST_DB
CREATE DATABASE IF NOT EXISTS test;
CREATE_TEST_DB
```

6. Finalmente, crie um script `change_permissions.sh` em `/tmp/WordPress/scripts`. Ele é usado para alterar as permissões de pastas no Apache.

> **Important**
>
> Este script mostra permissões atualizadas na pasta `/tmp/WordPress` para que qualquer pessoa possa gravar nele. Isso é necessário para que ele WordPress possa

gravar em seu banco de dados durante[Etapa 5: atualizar e reimplantar seu aplicativo WordPress](). Depois que o WordPress aplicativo estiver configurado, execute o comando a seguir para atualizar as permissões para uma configuração mais segura:

```
chmod -R 755 /var/www/html/WordPress
```

```
#!/bin/bash
chmod -R 777 /var/www/html/WordPress
```

7. Dê todas as permissões executáveis de scripts. Na linha de comando, digite:

```
chmod +x /tmp/WordPress/scripts/*
```

Adicionar um arquivo de especificação de aplicativo

Em seguida, adicione um arquivo de especificação do aplicativo (AppSpec arquivo), um arquivo formatado em YAML usado por to: CodeDeploy

- Mapear os arquivos de origem na sua revisão de aplicativo para os respectivos destinos na instância do Amazon EC2 de destino.
- Especificar permissões personalizadas para arquivos implantados.
- Especificar os scripts a serem executados na instância do Amazon EC2 de destino durante a implantação.

O AppSpec arquivo deve ser nomeado`appspec.yml`. Ele deve ser colocado no diretório raiz do código-fonte do aplicativo. Neste tutorial, o diretório raiz é `/tmp/WordPress`.

Com seu editor de texto, crie um arquivo chamado `appspec.yml`. Adicione as seguintes linhas ao arquivo:

```
version: 0.0
os: linux
files:
```

```
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

CodeDeploy usa esse AppSpec arquivo para copiar todos os arquivos da `/tmp/WordPress` pasta na máquina de desenvolvimento para a `/var/www/html/WordPress` pasta na instância do Amazon EC2 de destino. Durante a implantação, CodeDeploy executa os scripts especificados como `root` na `/var/www/html/WordPress/scripts` pasta na instância de destino do Amazon EC2 em eventos específicos durante o ciclo de vida da implantação, como e. **`BeforeInstall`** **`AfterInstall`** Se algum desses scripts levar mais de 300 segundos (5 minutos) para ser executado, CodeDeploy interrompe a implantação e marca a implantação como falhada.

Para obter mais informações sobre essas configurações, consulte o CodeDeploy AppSpec referência de arquivo.

> **Important**
>
> Os locais e números de espaços entre cada um dos itens desse arquivo são importantes. Se o espaçamento estiver incorreto, CodeDeploy gera um erro que pode ser difícil de depurar. Para obter mais informações, consulte AppSpec Espaçamento de arquivos.

Etapa 3: Faça o upload do seu WordPress aplicativo para o Amazon S3

Agora você preparará e enviará seu conteúdo de origem para um local a partir do qual CodeDeploy possa implantá-lo. As instruções a seguir mostram como provisionar um bucket do Amazon S3, preparar os arquivos da revisão de aplicativo para o bucket, agrupar os arquivos da revisão e, em seguida, enviá-la para esse bucket.

> **Note**
>
> Embora não seja abordado neste tutorial, você pode usá-lo CodeDeploy para implantar aplicativos de GitHub repositórios para instâncias. Para obter mais informações, consulte Integrando com CodeDeploy GitHub.

Tópicos

- Provisionar um bucket do Amazon S3
- Preparar os arquivos do aplicativo para o bucket
- Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento

Provisionar um bucket do Amazon S3

Crie um contêiner de armazenamento, ou bucket, no Amazon S3, ou use um bucket existente. Certifique-se de que você possa carregar a revisão no bucket e de que as instâncias do Amazon EC2 usadas nas implantações possam baixá-la desse bucket.

Você pode usar o AWS CLI console do Amazon S3 ou as APIs do Amazon S3 para criar um bucket do Amazon S3. Depois de criar o bucket, certifique-se de dar permissões de acesso a ele e à sua conta do AWS .

> **Note**
>
> Os nomes dos buckets devem ser exclusivos no Amazon S3 para todas as contas AWS . Se não for possível usar **`amzn-s3-demo-bucket`**, tente um nome de bucket diferente, como **`amzn-s3-demo-bucket`**, seguido por um traço e suas iniciais ou algum outro identificador exclusivo. Em seguida, certifique-se de substituir o nome do bucket por **`amzn-s3-demo-bucket`** sempre que essa identificação aparecer neste tutorial.
> O bucket do Amazon S3 deve ser criado na mesma AWS região em que suas instâncias de destino do Amazon EC2 são lançadas. Por exemplo, se você criar o bucket na região Leste

dos EUA (Norte da Virgínia), suas instâncias do Amazon EC2 de destino devem ser lançadas na região Leste dos EUA (Norte da Virgínia).

Tópicos

- Como criar um bucket do Amazon S3 (CLI)
- Como criar um bucket do Amazon S3 (console)
- Conceda permissões ao bucket do Amazon S3 e AWS account

Como criar um bucket do Amazon S3 (CLI)

Chame o comando mb para criar um bucket do Amazon S3 denominado **`amzn-s3-demo-bucket`**:

```
aws s3 mb s3://amzn-s3-demo-bucket --region region
```

Como criar um bucket do Amazon S3 (console)

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. No console do Amazon S3, escolha Criar bucket.
3. Na caixa Nome do bucket, digite um nome para o bucket.
4. Na lista Região, escolha a Região de destino e, em seguida, escolha Criar.

Conceda permissões ao bucket do Amazon S3 e AWS account

Você deve ter permissões para fazer upload no bucket do Amazon S3. Você pode especificar essas permissões por meio de uma política de bucket do Amazon S3. Por exemplo, na seguinte política de bucket do Amazon S3, o uso do caractere curinga (*) permite que `111122223333` a AWS conta faça upload de arquivos para qualquer diretório no bucket do Amazon S3 chamado: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
```

```
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Para ver o ID AWS da sua conta, consulte Encontrando o ID AWS da sua conta.

Agora é um bom momento para verificar se o bucket do Amazon S3 permitirá solicitações de download de cada instância participante do Amazon EC2. É possível especificar isso por meio de uma política do bucket do Amazon S3. Por exemplo, na seguinte política de bucket do Amazon S3, usar o caractere curinga (*) permite que qualquer instância do Amazon EC2 com um perfil de instância anexado do IAM contendo o ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` faça o download de arquivos de qualquer diretório no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`:

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:role/CodeDeployDemo"
                ]
            }
        }
    ]
}
```

Para obter informações sobre como gerar e anexar uma política de bucket do Amazon S3, consulte Exemplos de políticas de bucket.

Para obter informações sobre como criar e anexar uma política do IAM, consulte Trabalhando com políticas.

Preparar os arquivos do aplicativo para o bucket

Certifique-se de que os arquivos do WordPress aplicativo, o AppSpec arquivo e os scripts estejam organizados em sua máquina de desenvolvimento de forma semelhante à seguinte:

```
/tmp/
  |--WordPress/
      |-- appspec.yml
      |-- scripts/
      |    |-- change_permissions.sh
      |    |-- create_test_db.sh
      |    |-- install_dependencies.sh
      |    |-- start_server.sh
      |    |-- stop_server.sh
      |-- wp-admin/
      |    |-- (various files...)
      |-- wp-content/
      |    |-- (various files...)
      |-- wp-includes/
      |    |-- (various files...)
      |-- index.php
      |-- license.txt
      |-- readme.html
      |-- (various files ending with .php...)
```

Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento

Agrupe os arquivos do WordPress aplicativo e o AppSpec arquivo em um arquivo compactado (conhecido como revisão do aplicativo).

> **Note**
>
> Você pode ser cobrado por armazenar objetos em um bucket e por transferir revisões de aplicativos para dentro e fora de um bucket. Para obter mais informações, consulte Preço do Amazon S3.

1. Na máquina de desenvolvimento, mude para a pasta na qual os arquivos estão armazenados:

   ```
   cd /tmp/WordPress
   ```

> **Note**
>
> Se você não mudar para essa pasta, o empacotamento de arquivos começará na sua pasta atual. Por exemplo, se a sua pasta atual for `/tmp` em vez de `/tmp/WordPress`, o empacotamento começará com arquivos e subpastas na pasta `tmp`, o que pode incluir mais do que a subpasta `WordPress`.

2. Chame o comando create-application para registrar um novo aplicativo denominado **`WordPress_App`**:

```
aws deploy create-application --application-name WordPress_App
```

3. Chame o comando CodeDeploy push para agrupar os arquivos, fazer o upload das revisões para o Amazon S3 e registrar informações sobre a revisão CodeDeploy carregada, tudo em uma única ação.

```
aws deploy push \
  --application-name WordPress_App \
  --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \
  --ignore-hidden-files
```

Esse comando agrupa os arquivos do diretório atual (excluindo os arquivos ocultos) em um único arquivo chamado**`WordPressApp.zip`**, carrega a revisão no **`amzn-s3-demo-bucket`** bucket e registra as informações CodeDeploy sobre a revisão carregada.

Etapa 4: implantar seu WordPress aplicativo

Agora você implanta o exemplo de revisão do WordPress aplicativo que você carregou no Amazon S3. Você pode usar o console AWS CLI ou o CodeDeploy console para implantar a revisão e monitorar o progresso da implantação. Depois que a revisão do aplicativo for implementada com êxito, confira os resultados.

Tópicos

- Implante a revisão do seu aplicativo com CodeDeploy
- Monitorar sua implantação e solucionar problemas relacionados
- Verificar sua implantação

Implante a revisão do seu aplicativo com CodeDeploy

Use o console AWS CLI ou o console para implantar a revisão do seu aplicativo.

Tópicos

- Para implantar sua revisão de aplicativo (CLI)
- Para implantar sua revisão de aplicativo (console)

Para implantar sua revisão de aplicativo (CLI)

1. A implantação precisa de um grupo de implantação. No entanto, antes de criar o grupo de implantação, você precisa de um ARN de um perfil de serviço. Um perfil de serviço é um perfil do IAM que dá a um serviço permissão para agir em seu nome. Nesse caso, a função de serviço dá CodeDeploy permissão para acessar suas instâncias do Amazon EC2 para expandir (ler) suas tags de instância do Amazon EC2.

   Você já deve ter seguido as instruções em Criar um perfil de serviço (CLI) para criar um perfil de serviço. Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (CLI).

2. Agora que você tem o ARN do perfil de serviço, chame o comando create-deployment-group para criar um grupo de implantação chamado **`WordPress_DepGroup`**, associado ao aplicativo chamado **`WordPress_App`**, usando a tag do Amazon EC2 chamada **`CodeDeployDemo`** e a configuração de implantação chamada **`CodeDeployDefault.OneAtATime`**:

```
aws deploy create-deployment-group \
  --application-name WordPress_App \
  --deployment-group-name WordPress_DepGroup \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \
  --service-role-arn serviceRoleARN
```

> **Note**
>
> O comando create-deployment-group fornece suporte para a criação de acionadores que resultam no envio de notificações do Amazon SNS aos assinantes do tópico sobre eventos específicos em implantações e instâncias. O comando também oferece suporte a opções para reverter automaticamente as implantações e configurar alarmes para interromper as implantações quando os limites de monitoramento nos alarmes

da CloudWatch Amazon forem atingidos. Os comandos para essas ações não estão incluídos neste tutorial.

3. Antes de criar uma implantação, as instâncias do seu grupo de implantação devem ter o CodeDeploy agente instalado. Você pode instalar o agente a partir da linha de comando do AWS Systems Manager , com o seguinte comando:

```
aws ssm create-association \
  --name AWS-ConfigureAWSPackage \
  --targets Key=tag:Name,Values=CodeDeployDemo \
  --parameters action=Install,name=AWSCodeDeployAgent \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Esse comando cria uma associação no Systems Manager State Manager que instalará o CodeDeploy agente e tentará atualizá-lo às 2:00 de cada manhã de domingo. Para obter mais informações sobre o CodeDeploy agente, consulte Trabalhando com o CodeDeploy agente. Para obter mais informações sobre o Systems Manager, consulte O que é o AWS Systems Manager.

4. Agora, chame o comando create-deployment para criar uma implantação associada ao aplicativo **`WordPress_App`**, a configuração de implantação **`CodeDeployDefault.OneAtATime`** e o grupo de implantação **`WordPress_DepGroup`**, usando a revisão de aplicativo **`WordPressApp.zip`** no bucket **`amzn-s3-demo-bucket`**:

```
aws deploy create-deployment \
  --application-name WordPress_App \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --deployment-group-name WordPress_DepGroup \
  --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=WordPressApp.zip
```

Para implantar sua revisão de aplicativo (console)

1. Antes de usar o CodeDeploy console para implantar a revisão do aplicativo, você precisa de um ARN de função de serviço. Um perfil de serviço é um perfil do IAM que dá a um serviço permissão para agir em seu nome. Nesse caso, a função de serviço dá CodeDeploy permissão para acessar suas instâncias do Amazon EC2 para expandir (ler) suas tags de instância do Amazon EC2.

Você já deve ter seguido as instruções em Criar um perfil de serviço (console) para criar um perfil de serviço. Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (console).

2. Agora que você tem o ARN, use o CodeDeploy console para implantar a revisão do seu aplicativo:

   Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

3. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
4. Na lista de aplicativos, escolha WordPress_App.
5. Na guia Grupos de implantação, selecione Criar um grupo de implantação.
6. Em Nome do grupo de implantação, insira **`WordPress_DepGroup`**.
7. Em Tipo de implantação, escolha In-placeimplantação.
8. Em Configuração do ambiente, selecione Instâncias do Amazon EC2.
9. Em Configuração do agente com AWS Systems Manager, mantenha os padrões.
10. Em Chave, insira **`Name`**.
11. Em Valor, informe **`CodeDeployDemo`**.

    > **Note**
    >
    > Depois de digitar**`CodeDeployDemo`**, um 1 deve aparecer em Matching instances para confirmar que CodeDeploy encontrou uma instância correspondente do Amazon EC2.

12. Em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime.
13. Em Service role ARN (ARN da função de serviço), selecione o ARN da função de serviço e, em seguida, escolha Create deployment group (Criar grupo de implantação).
14. Escolha Criar implantação.

15. Em Deployment group (Grupo de implantação), selecione **`WordPress_DepGroup`**.
16. Ao lado de Tipo de repositório, selecione Meu aplicativo está armazenado no Amazon S3. Em Local da revisão, insira o local do exemplo de revisão do WordPress aplicativo que você enviou anteriormente para o Amazon S3. Para obter a localização:

    a. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.

    b. Na lista de buckets, selecione amzn-s3-demo-bucket (ou o nome do bucket no qual você fez upload da sua revisão de aplicação).

    c. Na lista de objetos, selecione WordPressApp.zip.

    d. Na guia Visão geral, copie o valor do campo Link na área de transferência.

    A aparência pode ser semelhante a esta:

    `https://s3.amazonaws.com/amzn-s3-demo-bucket/WordPressApp.zip`

    e. Retorne ao CodeDeploy console e, em Local da revisão, cole o valor do campo Link.
17. Se uma mensagem aparecer na lista File type (Tipo de arquivo), indicando que o tipo de arquivo não pôde ser detectado, escolha .zip.
18. (Opcional) Digite um comentário na caixa Deployment description.
19. Expanda Deployment group overrides (Substituições de grupos de implantação) e em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime.
20. Selecione Iniciar implantação. As informações sobre a implantação recém-criada aparecem na página Implantações.

Monitorar sua implantação e solucionar problemas relacionados

Use o console AWS CLI ou o console para monitorar e solucionar problemas de sua implantação.

Tópicos

- Para monitorar e solucionar problemas de implantação (CLI)
- Para monitorar e solucionar problemas de implantação (console)

Para monitorar e solucionar problemas de implantação (CLI)

1. Obtenha o ID de implantação chamando o comando list-deployments para o aplicativo chamado **`WordPress_App`** e o grupo de implantação chamado **`WordPress_DepGroup`**:

```
aws deploy list-deployments --application-name WordPress_App --deployment-group-
name WordPress_DepGroup --query 'deployments' --output text
```

2. Chame o comando get-deployment com o ID da implantação:

```
aws deploy get-deployment --deployment-id deploymentID --query
 'deploymentInfo.status' --output text
```

3. O comando retorna o status geral da implantação. Se a implantação for bem-sucedida, o valor será `Succeeded`.

   Se o status geral for `Failed`, você poderá chamar comandos como list-deployment-instances e get-deployment-instance para solucionar problemas. Para mais opções de solução de problemas, consulte Analisando arquivos de log para investigar falhas de implantação em instâncias.

Para monitorar e solucionar problemas de implantação (console)

Na página Implantações no CodeDeploy console, você pode monitorar o status da sua implantação na coluna Status.

Para obter mais informações sobre a implantação, especialmente se o valor da coluna Status tiver qualquer outro valor que não Succeeded:

1. Na tabela Deployments (Implantações), selecione o nome da implantação. Depois que uma implantação falhar, uma mensagem descrevendo o motivo da falha será exibida.
2. Na Instance activity (Atividade da instância), mais informações sobre a implantação são exibidas. Depois que uma implantação apresentar falha, você possivelmente conseguirá determinar em quais instâncias do Amazon EC2 e em qual etapa ocorreu a falha.
3. Se quiser realizar outros procedimentos de solução de problemas, use uma técnica como a descrita em View Instance Details. Você também pode analisar os arquivos de log de implantação em uma instância do Amazon EC2. Para obter mais informações, consulte Analisando arquivos de log para investigar falhas de implantação em instâncias.

Verificar sua implantação

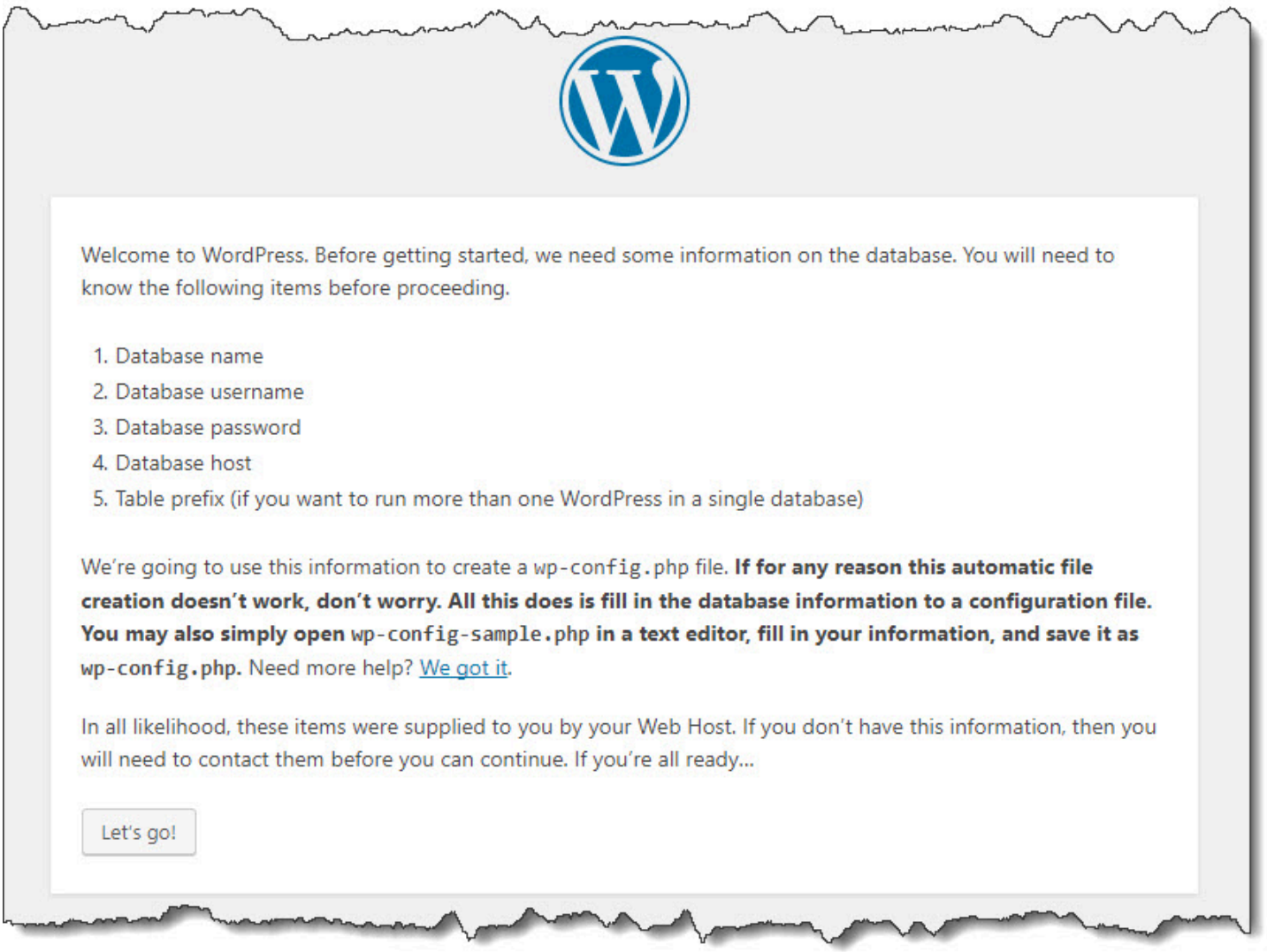
Depois que sua implantação for bem-sucedida, verifique se WordPress a instalação está funcionando. Use o endereço DNS público da instância do Amazon EC2, seguido por `/WordPress`,

para visualizar seu site em um navegador da Web. (Para obter o valor DNS público, no console do Amazon EC2, escolha a instância do Amazon EC2 e, na guia Descrição, procure o valor de DNS público.)

Por exemplo, se o endereço DNS público da sua instância do Amazon EC2 for **`ec2-01-234-567-890.compute-1.amazonaws.com`**, você usará o seguinte URL:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress
```

Ao visualizar o site em seu navegador, você deve ver uma página de WordPress boas-vindas semelhante à seguinte:



Se sua instância do Amazon EC2 não tiver uma regra de entrada HTTP adicionada ao seu grupo de segurança, a página de WordPress boas-vindas não aparecerá. Se aparecer uma mensagem

indicando que o servidor remoto não está respondendo, certifique-se de que o grupo de segurança de sua instância do Amazon EC2 tenha a regra de entrada. Para obter mais informações, consulte Adicione uma regra de entrada que permita tráfego HTTP à instância do Amazon EC2 do Amazon Linux ou do RHEL.

Etapa 5: atualizar e reimplantar seu aplicativo WordPress

Agora que você implantou com sucesso a revisão do aplicativo, atualize o WordPress código na máquina de desenvolvimento e use-o CodeDeploy para reimplantar o site. Em seguida, você deve ver as alterações de código na instância do Amazon EC2.

Tópicos

- Configurar o WordPress site
- Modificar o site
- Reimplantar o site

Configurar o WordPress site

Para ver os efeitos da alteração do código, conclua a configuração do WordPress site para que você tenha uma instalação totalmente funcional.

1. Digite a URL do seu site no navegador. A URL é o endereço DNS público da instância do Amazon EC2 mais a extensão `/WordPress`. Para este WordPress site de exemplo (e exemplo de endereço DNS público da instância Amazon EC2), a URL é. **`http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress`**
2. Se você ainda não configurou o site, a página de boas-vindas WordPress padrão é exibida. Escolha Let's go!.
3. Na página de configuração do banco de dados, digite os seguintes valores para usar o banco de dados MySQL padrão:
   - Nome do banco de dados: **`test`**
   - Nome de usuário: **`root`**
   - Senha: deixe em branco.
   - Host do banco de dados: **`localhost`**
   - Prefixo da tabela: **`wp_`**

Escolha Submit para configurar o banco de dados.

4. Continue a configuração do site. Na página de boas-vindas, preencha os valores desejados e escolha Instalar WordPress. Quando a instalação estiver concluída, faça login no seu painel.

> **Important**
>
> Durante a implantação do WordPress aplicativo, o **`change_permissions.sh`** script atualizou as permissões da `/tmp/WordPress` pasta para que qualquer pessoa possa gravar nela. Agora é um bom momento para executar o comando a seguir e restringir permissões para que apenas você, o proprietário, possa gravar nele:
>
> ```
> chmod -R 755 /var/www/html/WordPress
> ```

Modificar o site

Para modificar o WordPress site, acesse a pasta do aplicativo em sua máquina de desenvolvimento:

```
cd /tmp/WordPress
```

Para modificar algumas das cores do site, no arquivo `wp-content/themes/twentyfifteen/style.css`, use um editor de texto ou sed para alterar `#fff` para `#768331`.

No Linux ou outros sistemas com o GNU sed, use:

```
sed -i 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

No macOS, Unix ou outros sistemas com o BSD sed, use:

```
sed -i '' 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

Reimplantar o site

Agora que você modificou o código do site, use o Amazon S3 CodeDeploy para reimplantar o site.

Agrupe e carregue as alterações no Amazon S3, conforme descrito em Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento. (Ao seguir essas instruções, lembre-se de que não é necessário criar um aplicativo.) Especifique a mesma chave de antes para a nova revisão (**`WordPressApp.zip`**). Faça upload dela no mesmo bucket do Amazon S3 criado anteriormente (por exemplo, **`amzn-s3-demo-bucket`**).

Use o AWS CLI CodeDeploy console ou as CodeDeploy APIs para reimplantar o site.

Tópicos

- Para redistribuir o site (CLI)
- Para reimplantar o site (console)

Para redistribuir o site (CLI)

Chame o comando create-deployment para criar uma implantação com base na revisão recém-carregada. Use o aplicativo **`WordPress_App`**, a configuração de implantação **`CodeDeployDefault.OneAtATime`**, o grupo de implantação **`WordPress_DepGroup`**, e a revisão **`WordPressApp.zip`** no bucket **`amzn-s3-demo-bucket`**:

```
aws deploy create-deployment \
 --application-name WordPress_App \
 --deployment-config-name CodeDeployDefault.OneAtATime \
 --deployment-group-name WordPress_DepGroup \
 --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=WordPressApp.zip
```

Você pode verificar o status da implantação conforme descrito em Monitorar sua implantação e solucionar problemas relacionados.

Depois CodeDeploy de reimplantar o site, visite novamente o site em seu navegador para verificar se as cores foram alteradas. (Talvez seja necessário atualizar o navegador.) Se as cores mudaram, parabéns! Você modificou e reimplantou seu site com sucesso!

Para reimplantar o site (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
3. Na lista de aplicativos, escolha WordPress_App.
4. Na guia Grupos de implantação, escolha **`WordPress_DepGroup`**.
5. Escolha Criar implantação.
6. Na página Create deployment:

   a. Em Deployment group (Grupo de implantação), selecione **`WordPress_DepGroup`**.

   b. Na área Tipo de repositório, selecione Meu aplicativo está armazenado no Amazon S3 e copie o link do Amazon S3 da revisão para a caixa Local de revisão. Para encontrar o valor do link:

      i. Em uma guia separada do navegador:

         Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/

         Navegue até amzn-s3-demo-bucket, abra-o e, depois, selecione sua revisão: **`WordPressApp.zip`**.

      ii. Se o painel Propriedades não estiver visível no console do Amazon S3, escolha o botão Propriedades.

      iii. No painel Propriedades, copie o valor do campo Link na caixa Local da revisão no CodeDeploy console.

   c. Se uma mensagem aparecer informando que o tipo de arquivo não pôde ser detectado, escolha .zip.

   d. Deixe a caixa Descrição da implantação em branco.

   e. Expanda Deployment group overrides (Substituições de grupos de implantação) e em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime.

   f. Selecione Iniciar implantação. As informações sobre a implantação recém-criada aparecem na página Implantações.

g. Você pode verificar o status da implantação conforme descrito em Monitorar sua implantação e solucionar problemas relacionados.

   Depois CodeDeploy de reimplantar o site, visite novamente o site em seu navegador para verificar se as cores foram alteradas. (Talvez seja necessário atualizar o navegador.) Se as cores mudaram, parabéns! Você modificou e reimplantou seu site com sucesso!

Etapa 6: Limpe seu WordPress aplicativo e os recursos relacionados

Agora você atualizou o WordPress código com sucesso e reimplantou o site. Para evitar cobranças contínuas pelos recursos criados para este tutorial, você deve excluir o seguinte:

- Qualquer CloudFormation pilha (ou encerre qualquer instância do Amazon EC2, se você as criou fora do). CloudFormation
- Qualquer bucket do Amazon S3.
- O aplicativo `WordPress_App` no CodeDeploy.
- A associação AWS Systems Manager do gerente estadual para o CodeDeploy agente.

Você pode usar o AWS CLI, o Amazon S3 CloudFormation, o Amazon EC2 CodeDeploy e os consoles ou as APIs para realizar AWS a limpeza.

Tópicos

- Para limpar recursos (CLI)
- Para limpar recursos (console)
- Próximas etapas

Para limpar recursos (CLI)

1. Se você usou nosso CloudFormation modelo para este tutorial, chame o delete-stack comando na pilha chamada**`CodeDeployDemoStack`**. Isso encerrará todas as instâncias do Amazon EC2 acompanhantes e excluirá todos os perfis do IAM acompanhantes que essa pilha criou:

   ```
   aws cloudformation delete-stack --stack-name CodeDeployDemoStack
   ```

2. Para excluir o bucket do Amazon S3, chame o comando rm com a opção --recursive para o bucket denominado **`amzn-s3-demo-bucket`**. Isso excluirá o bucket e todos os seus objetos:

```
aws s3 rm s3://amzn-s3-demo-bucket --recursive --region region
```

3. Para excluir o aplicativo `WordPress_App`, chame o comando delete-application. Isso também excluirá todos os registros do grupo de implantação associados e os registros de implantação do aplicativo:

   ```
   aws deploy delete-application --application-name WordPress_App
   ```

4. Para excluir a associação do gerenciador de estados do Systems Manager, chame o comando delete-association.

   ```
   aws ssm delete-association --assocation-id association-id
   ```

   Você pode obter o *association-id* chamando o describe-association comando.

   ```
   aws ssm describe-association --name AWS-ConfigureAWSPackage --targets
    Key=tag:Name,Values=CodeDeployDemo
   ```

Se você não usou a CloudFormation pilha para este tutorial, chame o terminate-instances comando para encerrar todas as instâncias do Amazon EC2 que você criou manualmente. Forneça o ID da instância do Amazon EC2 a ser encerrada:

```
aws ec2 terminate-instances --instance-ids instanceId
```

Para limpar recursos (console)

Se você usou nosso CloudFormation modelo para este tutorial, exclua a CloudFormation pilha associada.

1. Faça login no Console de gerenciamento da AWS e abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. Na caixa Filtro, digite o nome da CloudFormation pilha que você criou anteriormente (por exemplo,**`CodeDeployDemoStack`**).
3. Marque a caixa ao lado do nome da pilha. No menu Ações, escolha Excluir pilha.

   CloudFormation exclui a pilha, encerra todas as instâncias acompanhantes do Amazon EC2 e exclui todas as funções do IAM associadas.

Para encerrar instâncias do Amazon EC2 que você criou fora de CloudFormation uma pilha:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. Na lista INSTÂNCIAS, escolha Instâncias.
3. Na caixa de pesquisa, digite o nome da instância do Amazon EC2 que você deseja encerrar (por exemplo, **`CodeDeployDemo`**) e depois pressione Enter.
4. Escolha o nome da instância do Amazon EC2
5. No menu Ações, selecione Estado da instância e escolha Encerrar. Quando solicitado, escolha Sim, Encerrar.

Repita essas etapas para cada instância.

Para excluir o bucket do Amazon S3:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/
2. Na lista de buckets, navegue e escolha o nome do bucket do Amazon S3 que você criou anteriormente (por exemplo, **`amzn-s3-demo-bucket`**).
3. Antes de poder excluir um bucket, você deve excluir seu conteúdo. Selecione todos os arquivos no bucket, como **`WordPressApp.zip`**. No menu Ações, escolha Excluir. Quando solicitado a confirmar a exclusão, escolha OK.
4. Depois que o bucket estiver vazio, você poderá excluí-lo. Na lista de buckets, escolha a linha do bucket (mas não o nome do bucket). Escolha Excluir bucket e quando solicitado a confirmar, escolha OK.

Para excluir o `WordPress_App` aplicativo de CodeDeploy:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
3. Na lista de aplicativos, escolha WordPress_App.
4. Na página Detalhes da aplicação, escolha Excluir aplicativo.
5. Quando solicitado, digite o nome do aplicativo para confirmar que você deseja excluí-lo e, em seguida, escolha Excluir.

Para excluir a associação do gerenciador de estados do Systems Manager:

1. Abra o AWS Systems Manager console em https://console.aws.amazon.com/systems-manager.
2. No painel de navegação, escolha State Manager.
3. Escolha a associação criada e selecione Excluir.

Próximas etapas

Se você chegou até aqui, parabéns! Você concluiu com êxito uma implantação do CodeDeploy e, depois, atualizou o código do seu site e o reimplantou.

Tutorial: implantar um aplicativo "Hello, World!" aplicativo com CodeDeploy (Windows Server)

Neste tutorial, você implanta uma única página da web em uma única instância do Amazon EC2 do Windows Server que executa o IIS (Serviços de Informações da Internet) como seu servidor web. Essa página da web exibe um "Hello, World!" simples mensagem.

Não é o que você está procurando?

- Em vez disso, para praticar a implantação em uma instância Amazon EC2 do Amazon Linux ou Red Hat Enterprise Linux (RHEL), consulte Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix).
- Para praticar a implantação em uma instância local, consulte Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).

As etapas deste tutorial são apresentadas sob a perspectiva do Windows. Embora você possa concluir a maioria dessas etapas em uma máquina local executando Linux, macOS ou Unix, é

necessário adaptar aquelas que abrangem caminhos de Windows-based diretório, como. `c:\temp` Além disso, se quiser se conectar à instância do Amazon EC2, você precisará de um aplicativo cliente que possa conectar via protocolo RDP à instância do Amazon EC2 que executa o Windows Server. (O Windows inclui um aplicativo cliente de conexão RDP por padrão.)

Antes de começar este tutorial, você deve preencher os pré-requisitosComeçando com CodeDeploy, incluindo configurar seu usuário, instalar ou atualizar o e criar um perfil de instância do AWS CLI IAM e uma função de serviço.

Tópicos

- Etapa 1: execute uma instância do Amazon EC2 do Windows Server
- Etapa 2: configurar seu conteúdo de origem para implantar na instância do Amazon EC2 do Windows Server
- Etapa 3: carregar seu aplicativo "Hello, World!" no Amazon S3
- Etapa 4: implantar seu aplicativo Hello World
- Etapa 5: atualizar e reimplantar o aplicativo "Hello, World!" aplicação
- Etapa 6: limpar seu aplicativo "Hello, World!" e os recursos relacionados

Etapa 1: execute uma instância do Amazon EC2 do Windows Server

Para implantar o aplicativo Hello World com CodeDeploy, você precisa de uma instância do Amazon EC2 executando o Windows Server.

Siga as instruções em Crie uma instância do Amazon EC2 para CodeDeploy. Quando estiver pronto para atribuir uma tag de instância do Amazon EC2 à instância, especifique a chave da tag de **`Name`** e o valor da tag de **`CodeDeployDemo`** (Se você especificar uma chave de tag ou um valor de tag diferente, as instruções em Etapa 4: implantar seu aplicativo Hello World poderão produzir resultados inesperados.)

Depois de iniciar a instância do Amazon EC2, retorne a essa página e continue na próxima seção. Não continue em Crie um aplicativo com CodeDeploy como próxima etapa.

Conecte-se à sua instância Amazon EC2

Depois que a sua instância do Amazon EC2 for iniciada, siga essas instruções para praticar a conexão com ela.

> **Note**
>
> Nestas instruções, partimos do princípio de que você esteja executando o Windows e o aplicativo cliente de Conexão de Área de Trabalho do Windows. Para obter informações, consulte Estabelecendo uma conexão com a sua instância do Windows usando o RDP. Talvez seja necessário adaptar essas instruções para outros sistemas operacionais ou outros aplicativos cliente de conexão RDP.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação, em Instâncias, escolha Instâncias.
3. Navegue e escolha sua instância do Windows Server na lista.
4. Selecione Conectar.
5. Selecione Get Password (Obter senha) e, em seguida, selecione Choose File (Selecionar arquivo).
6. Procure e escolha o arquivo de par de chaves da instância do Amazon EC2 associado à instância do Amazon EC2 no Windows Server e escolha Abrir.
7. Escolha Decrypt Password. Anote a senha que é exibida. Você vai precisar dele na etapa 10.
8. Escolha Download Remote Desktop File e abra o arquivo.
9. Se você for solicitado a se conectar, continue mesmo que o editor da conexão remota não possa ser identificado.
10. Digite a senha que você anotou na etapa 7 e continue. (Se seu aplicativo cliente de conexão RDP solicitar um nome de usuário, digite **`Administrator`**.)
11. Se você for solicitado a se conectar, continue mesmo que a identidade do computador remoto não possa ser verificada.
12. Quando você estiver conectado, a área de trabalho da instância do Amazon EC2 que executa o Windows Server será exibida.
13. Agora você pode desconectar-se da instância do Amazon EC2.

> **Warning**
>
> Não interrompa ou encerre a instância. Caso contrário, não será CodeDeploy possível implantá-lo.

Adicionar uma regra de entrada que permita tráfego HTTP à instância do Amazon EC2 do Windows Server.

A próxima etapa confirma se a instância do Amazon EC2 tem uma porta HTTP aberta para que você possa ver a página da web implantada em sua instância do Amazon EC2 do Windows Server em um navegador.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. Escolha Instâncias e, em seguida, escolha sua instância.
3. Na guia Descrição, em Grupos de segurança, escolha exibir regras de entrada.

   Você deve ver uma lista de regras em seu grupo de segurança como a seguinte:

   ```
   Security Groups associated with i-1234567890abcdef0
    Ports     Protocol     Source     launch-wizard-N
    22        tcp          0.0.0.0/0          #
   ```

4. Em Grupos de segurança, escolha o grupo de segurança para a sua instância do Amazon EC2. Ele pode se chamar **`launch-wizard-N`**. O ***N*** no nome é um número atribuído ao grupo de segurança quando a instância foi criada.

   Escolha a guia Entrada. Se o grupo de segurança da sua instância está configurado corretamente, você deve ver uma regra com os seguintes valores:

   - Tipo: HTTP
   - Protocolo: TCP
   - Port Range: 80
   - Fonte: 0.0.0. 0/0
5. Se você não encontrar uma regra com esses valores, use os procedimentos em Adicionar regras a um grupo de segurança para adicioná-las a uma nova regra de segurança.

Etapa 2: configurar seu conteúdo de origem para implantar na instância do Amazon EC2 do Windows Server

Agora é hora de configurar o conteúdo de origem do seu aplicativo, para que você tenha algo que possa implantar na instância do Amazon EC2. Para este tutorial, você implantará uma única página

da Web na instância do Amazon EC2 que executa o Windows Server, que executará o IIS (Serviços de Informações da Internet) como seu servidor web. Essa página da Web exibirá um simples "Hello, World!" mensagem.

Tópicos

- Criar página da Web
- Crie um script para executar seu aplicativo
- Adicionar um arquivo de especificação de aplicativo

Criar página da Web

1. Crie um subdiretório (subpasta) chamado `HelloWorldApp` na pasta `c:\temp` e, em seguida, mude para essa pasta.

```
mkdir c:\temp\HelloWorldApp
cd c:\temp\HelloWorldApp
```

> **Note**
>
> Não é necessário usar a localização `c:\temp` ou o nome da subpasta `HelloWorldApp`. Se você usar uma localização ou um nome de subpasta diferente, certifique-se de usá-lo ao longo de todo este tutorial.

2. Use um editor de texto para criar um arquivo dentro da pasta. Nomeie o arquivo `index.html`.

```
notepad index.html
```

3. Adicione o seguinte código HTML ao arquivo e salve-o em seguida.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
TR/html4/loose.dtd">
<html>
<head>
  <title>Hello, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #0188cc;
```

```
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello, World!</h1></div>
  <div align="center"><h2>You have successfully deployed an application using
 CodeDeploy</h2></div>
  <div align="center">
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/
codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Crie um script para executar seu aplicativo

Em seguida, você criará um script que CodeDeploy será usado para configurar o servidor web na instância de destino do Amazon EC2.

1. Na mesma subpasta em que o arquivo `index.html` está salvo, use um editor de texto para criar outro arquivo. Nomeie o arquivo `before-install.bat`.

   ```
   notepad before-install.bat
   ```

2. Adicione o seguinte código script em lotes ao arquivo e salve-o em seguida.

   ```
   REM Install Internet Information Server (IIS).
   c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Import-Module -
   Name ServerManager
   c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Install-
   WindowsFeature Web-Server
   ```

Adicionar um arquivo de especificação de aplicativo

Em seguida, você adicionará um arquivo de especificação do aplicativo (AppSpec arquivo), além da página da Web e do arquivo de script em lote. O AppSpec arquivo é um arquivo formatado em YAML usado por para: CodeDeploy

- Mapear os arquivos de origem na sua revisão de aplicativo para os respectivos destinos na instância.
- Especifique os scripts a serem executados na instância durante a implantação.

O AppSpec arquivo deve ser nomeado`appspec.yml`. Ele deve ser colocado na pasta raiz do código-fonte do aplicativo.

1. Na mesma subpasta em que os arquivos `index.html` e `before-install.bat` estão salvos, use um editor de texto para criar outro arquivo. Nomeie o arquivo `appspec.yml`.

   ```
   notepad appspec.yml
   ```

2. Adicione o seguinte código YAML ao arquivo e salve-o em seguida.

   ```
   version: 0.0
   os: windows
   files:
     - source: \index.html
       destination: c:\inetpub\wwwroot
   hooks:
     BeforeInstall:
       - location: \before-install.bat
         timeout: 900
   ```

CodeDeploy usará esse AppSpec arquivo para copiar o `index.html` arquivo na pasta raiz do código-fonte do aplicativo para a `c:\inetpub\wwwroot` pasta na instância do Amazon EC2 de destino. Durante a implantação, CodeDeploy executará o script em `before-install.bat` lote na instância de destino do Amazon EC2 durante o evento do ciclo de vida da **`BeforeInstall`** implantação. Se esse script levar mais de 900 segundos (15 minutos) para ser executado, CodeDeploy interromperá a implantação e marcará a implantação na instância do Amazon EC2 como falha.

Para obter mais informações sobre essas configurações, consulte o CodeDeploy AppSpec referência de arquivo.

> **Important**
>
> Os locais e números de espaços entre cada um dos itens desse arquivo são importantes. Se o espaçamento estiver incorreto, CodeDeploy gerará um erro que pode ser difícil de depurar. Para obter mais informações, consulte AppSpec Espaçamento de arquivos.

Etapa 3: carregar seu aplicativo "Hello, World!" no Amazon S3

Agora você preparará e enviará seu conteúdo de origem para um local a partir do qual CodeDeploy possa implantá-lo. As instruções a seguir mostram como provisionar um bucket do Amazon S3, preparar os arquivos da revisão de aplicativo para o bucket, agrupar os arquivos da revisão e, em seguida, enviá-la para esse bucket.

> **Note**
>
> Embora não seja abordado neste tutorial, você pode usá-lo CodeDeploy para implantar aplicativos de GitHub repositórios para instâncias. Para obter mais informações, consulte Integrando com CodeDeploy GitHub.

Tópicos

- Provisionar um bucket do Amazon S3
- Preparar os arquivos do aplicativo para o bucket
- Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento

Provisionar um bucket do Amazon S3

Crie um contêiner de armazenamento, ou bucket, no Amazon S3, ou use um bucket existente. Certifique-se de que você possa carregar a revisão no bucket e de que as instâncias do Amazon EC2 usadas nas implantações possam baixá-la desse bucket.

Você pode usar o AWS CLI console do Amazon S3 ou as APIs do Amazon S3 para criar um bucket do Amazon S3. Depois de criar o bucket, certifique-se de dar permissões de acesso a ele e ao seu usuário do CodeDeploy.

> **Note**
>
> Os nomes dos buckets devem ser exclusivos no Amazon S3 para todas as contas AWS . Se não for possível usar **`amzn-s3-demo-bucket`**, tente um nome de bucket diferente, como **`amzn-s3-demo-bucket`**, seguido por um traço e suas iniciais ou algum outro identificador exclusivo. Em seguida, certifique-se de substituir o nome do bucket por **`amzn-s3-demo-bucket`** sempre que essa identificação aparecer neste tutorial.
> O bucket do Amazon S3 deve ser criado na mesma AWS região em que suas instâncias de destino do Amazon EC2 são lançadas. Por exemplo, se você criar o bucket na região Leste dos EUA (Norte da Virgínia), suas instâncias do Amazon EC2 de destino devem ser lançadas na região Leste dos EUA (Norte da Virgínia).

Tópicos

- Como criar um bucket do Amazon S3 (CLI)
- Como criar um bucket do Amazon S3 (console)
- Conceda permissões ao bucket do Amazon S3 e ao seu AWS account

Como criar um bucket do Amazon S3 (CLI)

Chame o comando mb para criar um bucket do Amazon S3 denominado **`amzn-s3-demo-bucket`**:

```
aws s3 mb s3://amzn-s3-demo-bucket --region region
```

Como criar um bucket do Amazon S3 (console)

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. No console do Amazon S3, escolha Criar bucket.
3. Na caixa Nome do bucket, digite um nome para o bucket.
4. Na lista Região, escolha a Região de destino e, em seguida, escolha Criar.

Conceda permissões ao bucket do Amazon S3 e ao seu AWS account

Você deve ter permissões para fazer upload no bucket do Amazon S3. Você pode especificar essas permissões por meio de uma política de bucket do Amazon S3. Por exemplo, na seguinte política de

bucket do Amazon S3, o uso do caractere curinga (*) permite que `111122223333` a AWS conta faça upload de arquivos para qualquer diretório no bucket do Amazon S3 chamado: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Para ver o ID AWS da sua conta, consulte Encontrando o ID AWS da sua conta.

Agora é um bom momento para verificar se o bucket do Amazon S3 permitirá solicitações de download de cada instância participante do Amazon EC2. É possível especificar isso por meio de uma política do bucket do Amazon S3. Por exemplo, na seguinte política de bucket do Amazon S3, usar o caractere curinga (*) permite que qualquer instância do Amazon EC2 com um perfil de instância anexado do IAM contendo o ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` faça o download de arquivos de qualquer diretório no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`:

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
```

```
                    "arn:aws:iam::444455556666:role/CodeDeployDemo"
                ]
            }
        }
    ]
}
```

Para obter informações sobre como gerar e anexar uma política de bucket do Amazon S3, consulte Exemplos de políticas de bucket.

O usuário CodeDeploy administrativo que você criou também Etapa 1: configuração deve ter permissão para carregar a revisão no bucket do Amazon S3. Uma forma de especificar isso é por meio de uma política do IAM, que você adiciona ao conjunto de permissões do usuário, ou de um perfil do IAM (que você permite que o usuário assuma). A seguinte política do IAM permite que o usuário faça upload de revisões em qualquer lugar do bucket do Amazon S3 denominado `amzn-s3-demo-bucket`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":["s3:PutObject"],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

Para obter informações sobre como criar uma política do IAM, consulte Criação de políticas do IAM no Guia do usuário do IAM. Para obter informações sobre como adicionar uma política a um conjunto de permissões, consulte Criar um conjunto de permissões no Guia do usuário do Centro de Identidade do AWS IAM .

Preparar os arquivos do aplicativo para o bucket

Certifique-se de que a página da web, o AppSpec arquivo e o script estejam organizados em sua máquina de desenvolvimento da seguinte forma:

```
c:\
  |-- temp\
        |--HelloWorldApp\
             |-- appspec.yml
             |-- before-install.bat
             |-- index.html
```

Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento

Agrupe os arquivos em um arquivamento (conhecido como uma revisão de aplicativo).

> **Note**
>
> Você pode ser cobrado por armazenar objetos em um bucket e por transferir revisões de aplicativos para dentro e fora de um bucket. Para obter mais informações, consulte Preço do Amazon S3.

1. Na máquina de desenvolvimento, mude para a pasta na qual os arquivos estão armazenados:

   ```
   cd c:\temp\HelloWorldApp
   ```

   > **Note**
   >
   > Se você não mudar para essa pasta, o empacotamento de arquivos começará na sua pasta atual. Por exemplo, se a sua pasta atual for `c:\temp`, e não `c:\temp\HelloWorldApp`, o empacotamento começará com os arquivos e as subpastas da pasta `c:\temp`, o que poderá incluir mais do que a subpasta `HelloWorldApp`.

2. Chame o create-application comando para registrar um novo aplicativo chamado **`HelloWorld_App`** com CodeDeploy:

   ```
   aws deploy create-application --application-name HelloWorld_App
   ```

3. Chame o comando CodeDeploy push para agrupar os arquivos, fazer o upload das revisões para o Amazon S3 e registrar informações sobre a revisão CodeDeploy carregada, tudo em uma única ação.

```
aws deploy push --application-name HelloWorld_App --s3-location s3://amzn-s3-demo-
bucket/HelloWorld_App.zip --ignore-hidden-files
```

Esse comando agrupa os arquivos do diretório atual (excluindo os arquivos ocultos) em um único arquivo chamado`HelloWorld_App.zip`, carrega a revisão no **`amzn-s3-demo-bucket`** bucket e registra as informações CodeDeploy sobre a revisão carregada.

Etapa 4: implantar seu aplicativo Hello World

Agora, você implanta o exemplo da revisão de aplicativo Hello World! que fez o upload no Amazon S3. Você usa o AWS CLI ou o CodeDeploy console para implantar a revisão e monitorar o progresso da implantação. Depois que a revisão do aplicativo for implementada com êxito, confira os resultados.

Tópicos

- Implemente a revisão do seu aplicativo com CodeDeploy
- Monitorar sua implantação e solucionar problemas relacionados
- Verificar sua implantação

Implemente a revisão do seu aplicativo com CodeDeploy

Você pode implantar seu aplicativo usando a CLI ou o console.

Tópicos

- Para implantar sua revisão de aplicativo (CLI)
- Para implantar sua revisão de aplicativo (console)

Para implantar sua revisão de aplicativo (CLI)

1. Primeiro, a implantação precisa de um grupo de implantação. No entanto, antes de criar o grupo de implantação, você precisa de um ARN de um perfil de serviço. Um perfil de serviço é um perfil do IAM que dá a um serviço permissão para agir em seu nome. Nesse caso, a função de serviço dá CodeDeploy permissão para acessar suas instâncias do Amazon EC2 para expandir (ler) suas tags de instância do Amazon EC2.

Você já deve ter seguido as instruções em Criar um perfil de serviço (CLI) para criar um perfil de serviço. Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (CLI).

2. Agora que você tem o ARN de perfil de serviço, chame o comando create-deployment-group para criar um grupo de implantação chamado **HelloWorld_DepGroup**, associado ao aplicativo chamado **HelloWorld_App**, usando a tag de instância do Amazon EC2 chamada **CodeDeployDemo** e a configuração de implantação chamada **CodeDeployDefault.OneAtATime**, com o ARN de perfil de serviço:

```
aws deploy create-deployment-group --application-name HelloWorld_App
 --deployment-group-name HelloWorld_DepGroup --deployment-
config-name CodeDeployDefault.OneAtATime --ec2-tag-filters
 Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE --service-role-arn serviceRoleARN
```

> **Note**
>
> O comando create-deployment-group fornece suporte para a criação de acionadores que resultam no envio de notificações do Amazon SNS aos assinantes do tópico sobre eventos específicos em implantações e instâncias. O comando também oferece suporte a opções para reverter automaticamente as implantações e configurar alarmes para interromper as implantações quando os limites de monitoramento nos alarmes da CloudWatch Amazon forem atingidos. Os comandos para essas ações não estão incluídos neste tutorial.

3. Antes de criar uma implantação, as instâncias do seu grupo de implantação devem ter o CodeDeploy agente instalado. Você pode instalar o agente a partir da linha de comando do AWS Systems Manager , com o seguinte comando:

```
aws ssm create-association --name AWS-ConfigureAWSPackage
 --targets Key=tag:Name,Values=CodeDeployDemo --parameters
 action=Install,name=AWSCodeDeployAgent --schedule-expression "cron(0 2 ? * SUN
 *)"
```

Esse comando cria uma associação no Systems Manager State Manager que instalará o CodeDeploy agente e tentará atualizá-lo às 2:00 todos os domingos de manhã. Para obter mais informações sobre o CodeDeploy agente, consulte Trabalhando com o CodeDeploy agente. Para obter mais informações sobre o Systems Manager, consulte O que é o AWS Systems Manager.

4. Agora, chame o comando create-deployment para criar uma implantação associada ao aplicativo **`HelloWorld_App`**, a configuração de implantação **`CodeDeployDefault.OneAtATime`** e o grupo de implantação **`HelloWorld_DepGroup`**, usando a revisão de aplicativo **`HelloWorld_App.zip`** no bucket **`amzn-s3-demo-bucket`**:

   ```
   aws deploy create-deployment --application-name HelloWorld_App --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
   ```

Para implantar sua revisão de aplicativo (console)

1. Antes de usar o CodeDeploy console para implantar a revisão do aplicativo, você precisa de um ARN de função de serviço. Um perfil de serviço é um perfil do IAM que dá a um serviço permissão para agir em seu nome. Nesse caso, a função de serviço dá CodeDeploy permissão para acessar suas instâncias do Amazon EC2 para expandir (ler) suas tags de instância do Amazon EC2.

   Você já deve ter seguido as instruções em Criar um perfil de serviço (console) para criar um perfil de serviço. Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (console).
2. Agora que você tem o ARN, você pode usar o CodeDeploy console para implantar a revisão do seu aplicativo.

   Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.
3. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
4. Escolha HelloWorld_App.
5. Na guia Grupos de implantação, selecione Criar um grupo de implantação.
6. Em Nome do grupo de implantação, insira **`HelloWorld_DepGroup`**.
7. Em Service Role (Função do serviço), selecione o nome da função do serviço.

8. Em Tipo de implantação, escolha In-place.
9. Em Configuração do ambiente, selecione Instâncias do Amazon EC2.
10. Em Configuração do agente com AWS Systems Manager, mantenha os padrões.
11. Em Chave, insira **`Name`**.
12. Em Valor, informe **`CodeDeployDemo`**.
13. Em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime.
14. Em Balanceador de carga, desmarque Habilitar balanceamento de carga.
15. Selecione Criar grupo de implantação.
16. Escolha Criar implantação.
17. No grupo de implantação, escolha HelloWorld_ DepGroup
18. Em Tipo de revisão, selecione Meu aplicativo está armazenado no Amazon S3 e, em seguida, em Local de revisão, digite o local da revisão do aplicativo de exemplo Hello World! que você fez upload no Amazon S3 anteriormente. Para obter a localização:
    a. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
    b. Na lista de buckets, selecione amzn-s3-demo-bucket (ou o nome do bucket no qual você fez upload da sua revisão de aplicação).
    c. Na lista de objetos, escolha HelloWorld_ App.zip.
    d. Na guia Overview (Visão geral), selecione Copy path (Copiar caminho).
    e. Retorne ao CodeDeploy console e, em Local da revisão, cole o valor do campo Link.
19. Em Tipo de arquivo de revisão, selecione .zip.
20. (Opcional) Insira um comentário em Deployment description (Descrição da implantação).
21. Escolha Criar implantação. As informações sobre a implantação recém-criada aparecem na página Implantações.

Monitorar sua implantação e solucionar problemas relacionados

Use o console AWS CLI ou o console para monitorar e solucionar problemas de sua implantação.

Tópicos

- Para monitorar e solucionar problemas de implantação (CLI)

- Para monitorar e solucionar problemas de implantação (console)

Para monitorar e solucionar problemas de implantação (CLI)

1. Obtenha o ID de implantação chamando o comando list-deployments para o aplicativo chamado **`HelloWorld_App`** e o grupo de implantação chamado **`HelloWorld_DepGroup`**:

   ```
   aws deploy list-deployments --application-name HelloWorld_App --deployment-group-
   name HelloWorld_DepGroup --query "deployments" --output text
   ```

2. Chame o comando get-deployment com o ID da implantação:

   ```
   aws deploy get-deployment --deployment-id deploymentID --query
    "deploymentInfo.status" --output text
   ```

3. O comando retorna o status geral da implantação. Se a implantação for bem-sucedida, o valor será `Succeeded`.

   Se o status geral for `Failed`, você poderá chamar comandos como list-deployment-instances e get-deployment-instance para solucionar problemas. Para mais opções de solução de problemas, consulte Analisando arquivos de log para investigar falhas de implantação em instâncias.

Para monitorar e solucionar problemas de implantação (console)

Na página Implantações no CodeDeploy console, você pode monitorar o status da sua implantação na coluna Status.

Para obter mais informações sobre a implantação, especialmente se o valor da coluna Status tiver qualquer outro valor que não Succeeded:

1. Na tabela Deployments (Implantações), selecione o ID da implantação. Depois que uma implantação apresentar falha, uma mensagem descrevendo o motivo da falha será exibida na página de detalhes da implantação.
2. Serão exibidas mais informações sobre as instâncias de implantação. Depois que uma implantação apresentar falha, você possivelmente conseguirá determinar em quais instâncias do Amazon EC2 e em qual etapa ocorreu a falha.
3. Se quiser realizar outros procedimentos de solução de problemas, use uma técnica como View Instance Details. Você também pode analisar os arquivos de log de implantação em uma

instância do Amazon EC2. Para obter mais informações, consulte Analisando arquivos de log para investigar falhas de implantação em instâncias.

Verificar sua implantação

Depois que a sua implantação for bem-sucedida, verifique se a sua instalação está funcionando. Use o endereço DNS público da instância do Amazon EC2 para visualizar a página da Web em um navegador da Web. (Para obter o valor DNS público, no console do Amazon EC2, escolha a instância do Amazon EC2 e, na guia Description, procure o valor de DNS público.)

Por exemplo, se o endereço DNS público da sua instância do Amazon EC2 for **`ec2-01-234-567-890.compute-1.amazonaws.com`**, você usará o seguinte URL:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Se o procedimento for bem-sucedido, você verá uma página da web Hello World!.

Etapa 5: atualizar e reimplantar o aplicativo "Hello, World!" aplicação

Agora que você implantou com sucesso a revisão do seu aplicativo, na máquina de desenvolvimento, atualize o código da página da Web e use-o CodeDeploy para reimplantar o site. Após a redistribuição, você poderá ver as alterações na instância do Amazon EC2.

Tópicos

- Modificar a página da web
- Reimplantar o site

Modificar a página da web

1. Vá para a subpasta `c:\temp\HelloWorldApp` e use um editor de texto para modificar o arquivo `index.html`:

   ```
   cd c:\temp\HelloWorldApp
   notepad index.html
   ```

2. Revise o conteúdo do arquivo `index.html` para alterar a cor do plano de fundo e parte do texto na página da web e depois salve o arquivo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
TR/html4/loose.dtd">
<html>
<head>
  <title>Hello Again, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #66cc00;
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello Again, World!</h1></div>
  <div align="center"><h2>You have successfully deployed a revision of an
 application using CodeDeploy</h2></div>
  <div align="center">
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/
codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Reimplantar o site

Agora que você modificou o código, use o Amazon S3 e reimplante CodeDeploy a página da web.

Agrupe e carregue as alterações no Amazon S3, conforme descrito em Agrupar os arquivos do aplicativo em um único arquivamento e enviar o arquivamento. (Ao seguir essas instruções, você não precisa criar um novo aplicativo.) Especifique a mesma chave de antes para a revisão (**`HelloWorld_App.zip`**). Faça upload dela no mesmo bucket do Amazon S3 criado anteriormente (por exemplo, **`amzn-s3-demo-bucket`**).

Use o console AWS CLI ou o CodeDeploy console para reimplantar o site.

Tópicos

- Para redistribuir o site (CLI)
- Para reimplantar o site (console)

Para redistribuir o site (CLI)

Chame o comando create-deployment para criar uma implantação baseada na revisão que fez upload, novamente usando o aplicativo **HelloWorld_App**, a configuração de implantação **CodeDeployDefault.OneAtATime**, o grupo de implantação **HelloWorld_DepGroup** e a revisão **HelloWorld_App.zip** no bucket **amzn-s3-demo-bucket**:

```
 aws deploy create-deployment --application-name HelloWorld_App --deployment-config-
name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-
location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
```

Você pode verificar o status da nova implantação conforme descrito em Monitorar sua implantação e solucionar problemas relacionados.

Quando CodeDeploy tiver reimplantado o site, visite novamente o site em seu navegador para verificar se a cor de fundo e o texto na página da Web foram alterados. (Talvez seja necessário atualizar o navegador.) Se a cor de fundo e o texto tiverem sido alterados, parabéns! Você modificou e reimplantou seu site!

Para reimplantar o site (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, selecione Applications (Aplicativos).
3. Na lista Aplicativos, escolha HelloWorld_App.
4. Na guia Deployments (Implantações), escolha Create deployment (Criar implantação).

   a. Na lista de grupos de implantação, escolha HelloWorld_ DepGroup.
   b. Em Local de revisão, digite o link Amazon S3 da revisão.

      Para encontrar o valor do link:

      i. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/

Navegue até amzn-s3-demo-bucket, abra-o e, depois, selecione a revisão, **`HelloWorld_App.zip`**, no console do Amazon S3.

ii. Se o painel Propriedades não estiver visível no console do Amazon S3, escolha o botão Propriedades.

iii. No painel Properties (Propriedades), copie o valor do campo Link.

iv. Retorne ao CodeDeploy console e cole o link no local da revisão.

v.

c. Em Tipo de arquivo de revisão, se uma mensagem aparecer indicando que o tipo de arquivo não pôde ser detectado, selecione .zip.

d. Deixe Descrição da implantação em branco.

e. Expanda Deployment group overrides (Revogações do grupo de implantação)Na lista Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime e, em seguida, escolha Create deployment (Criar implantação).

Você pode verificar o status da implantação conforme descrito em Monitorar sua implantação e solucionar problemas relacionados.

Quando CodeDeploy tiver reimplantado o site, visite novamente o site em seu navegador para verificar se a cor de fundo e o texto na página da Web foram alterados. (Talvez seja necessário atualizar o navegador.) Se a cor de fundo e o texto tiverem sido alterados, parabéns! Você modificou e reimplantou seu site!

Etapa 6: limpar seu aplicativo "Hello, World!" e os recursos relacionados

Agora, você fez uma atualização bem-sucedida no código de "Hello, World!" e reimplantou o site. Para evitar cobranças contínuas pelos recursos criados neste tutorial, exclua o seguinte:

- Qualquer CloudFormation pilha (ou encerre qualquer instância do Amazon EC2, se você as criou fora do). CloudFormation
- Qualquer bucket do Amazon S3.
- O aplicativo `HelloWorld_App` no CodeDeploy.
- A associação AWS Systems Manager do gerente estadual para o CodeDeploy agente.

Você pode usar o AWS CLI, o Amazon S3 CloudFormation, o Amazon EC2 CodeDeploy e os consoles ou as APIs para realizar AWS a limpeza.

Tópicos

- Para usar recursos de limpeza (CLI)
- Para limpar recursos (console)
- Próximas etapas

Para usar recursos de limpeza (CLI)

1. Se você usou a CloudFormation pilha neste tutorial, exclua a pilha chamando o delete-stack comando na pilha chamada. **`CodeDeployDemoStack`** Isso encerra todas as instâncias do Amazon EC2 acompanhantes e exclui todos os perfis do IAM acompanhantes originalmente criados pela pilha.

   ```
   aws cloudformation delete-stack --stack-name CodeDeployDemoStack
   ```

2. Para excluir o bucket do Amazon S3, chame o comando rm com a opção --recursive para o bucket denominado **`amzn-s3-demo-bucket`**. Isso exclui o bucket e todos os seus objetos.

   ```
   aws s3 rm s3://amzn-s3-demo-bucket --recursive --region region
   ```

3. Para excluir o `HelloWorld_App` aplicativo de CodeDeploy, chame o delete-application comando. Isso exclui todos os registros do grupo de implantação associados e os registros de implantação do aplicativo.

   ```
   aws deploy delete-application --application-name HelloWorld_App
   ```

4. Para excluir a associação do gerenciador de estados do Systems Manager, chame o comando delete-association.

   ```
   aws ssm delete-association --assocation-id association-id
   ```

   Você pode obter o *`association-id`* chamando o describe-association comando.

   ```
   aws ssm describe-association --name AWS-ConfigureAWSPackage --targets
    Key=tag:Name,Values=CodeDeployDemo
   ```

5. Se você não usou a CloudFormation pilha para este tutorial, chame o terminate-instances comando para encerrar as instâncias do Amazon EC2 que você criou manualmente. Forneça o ID da instância do Amazon EC2 a ser encerrada.

```
aws ec2 terminate-instances --instance-ids instanceId
```

Para limpar recursos (console)

Se você usou nosso CloudFormation modelo para este tutorial, exclua a CloudFormation pilha associada.

1. Faça login no Console de gerenciamento da AWS e abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. Na caixa de pesquisa, digite o nome da CloudFormation pilha (por exemplo,**`CodeDeployDemoStack`**).
3. Marque a caixa ao lado do nome da pilha.
4. No menu Ações, escolha Excluir pilha. Isso exclui a pilha, encerrará todas as instâncias do Amazon EC2 acompanhantes e excluirá todos os perfis do IAM acompanhantes.

Para encerrar instâncias do Amazon EC2 que você criou fora de CloudFormation uma pilha:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. Na área Instâncias, escolha Instâncias.
3. Na caixa de pesquisa, digite o nome da instância do Amazon EC2 que você deseja encerrar e pressione Enter.
4. Escolha a instância do Amazon EC2.
5. Escolha Ações, escolha Estado da instância e escolha Encerrar. Quando solicitado, escolha Sim, Encerrar. Repita essas etapas para qualquer instância do Amazon EC2 adicional.

Para excluir o bucket do Amazon S3:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/

2. Na lista de buckets, navegue e escolha o nome do bucket do Amazon S3 (por exemplo, **`amzn-s3-demo-bucket`**)
3. Antes de poder excluir um bucket, você deve excluir seu conteúdo. Selecione todos os arquivos no bucket, como **`HelloWorld_App.zip`**. No menu Ações, escolha Excluir. Quando solicitado a confirmar a exclusão, escolha OK.
4. Depois que o bucket estiver vazio, você poderá excluí-lo. Na lista de buckets, escolha a linha do bucket (mas não o nome do bucket). Escolha Excluir bucket e quando solicitado a confirmar, escolha OK.

Para excluir o `HelloWorld_App` aplicativo de CodeDeploy:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Selecione **`HelloWorld_App`**.
4. Selecione Excluir aplicativo.
5. Quando solicitado, digite **`Delete`** e escolha Excluir.

Para excluir a associação do gerenciador de estados do Systems Manager:

1. Abra o AWS Systems Manager console em https://console.aws.amazon.com/systems-manager.
2. No painel de navegação, escolha State Manager.
3. Escolha a associação criada e selecione Excluir.

Próximas etapas

Se você chegou aqui, concluiu com êxito uma implantação com CodeDeploy. Parabéns!

Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux)

Este tutorial ajuda você a ganhar experiência, CodeDeploy orientando você na implantação de uma amostra de revisão de aplicativo em uma única instância local, ou seja, um dispositivo físico que não é uma instância do Amazon EC2, executando Windows Server, Ubuntu Server ou Red Hat Enterprise Linux (RHEL). Para obter informações sobre instâncias locais e como elas funcionam CodeDeploy, consulteWorking with On-Premises Instances.

Não é o que você está procurando?

- Para praticar a implantação em uma instância do Amazon EC2 executando o Amazon Linux ou RHEL, consulte Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix).
- Para praticar a implantação em uma instância do Amazon EC2 executando o Windows Server, consulte Tutorial: implantar um aplicativo "Hello, World!" aplicativo com CodeDeploy (Windows Server).

Tópicos

- Pré-requisitos
- Etapa 1: configurar a instância on-premises
- Etapa 2: criar uma revisão de aplicativo de amostra
- Etapa 3: agrupar e carregar sua revisão de aplicativo no Amazon S3
- Etapa 4: implantar sua revisão de aplicativo
- Etapa 5: verificar sua implantação
- Etapa 6: limpar os recursos

Pré-requisitos

Antes de iniciar este tutorial, você deve preencher os pré-requisitos emComeçando com CodeDeploy, que incluem configurar um usuário, instalar ou atualizar o e criar uma função de AWS CLI serviço. Você não precisa criar um perfil de instância do IAM conforme descrito nos pré-requisitos. On-premises as instâncias não usam perfis de instância do IAM.

O dispositivo físico que você configurará como uma instância local deve estar executando um dos sistemas operacionais listados em Sistemas operacionais suportados pelo CodeDeploy agente.

Etapa 1: configurar a instância on-premises

Para poder implantar na sua instância local, você deve configurá-la. Siga as instruções em Working with On-Premises Instances e retorne para esta página.

Instale o CodeDeploy agente

Depois de configurar sua instância local, siga as etapas para instâncias locais em Instalar o CodeDeploy agente e volte para esta página.

Etapa 2: criar uma revisão de aplicativo de amostra

Nesta etapa, você cria uma revisão do aplicativo de amostra para implantar na instância no local.

Como é difícil saber quais softwares e atributos já estão instalados (ou podem ser instalados de acordo com as políticas da sua organização) na sua instância on-premises, a revisão de aplicativo de exemplo que oferecemos aqui usa simplesmente scripts em lote (para o Windows Server) ou scripts shell (para o Ubuntu Server e o RHEL) para gravar arquivos de texto em um localização na sua instância on-premises. Um arquivo é gravado para cada um dos vários eventos do ciclo de vida da CodeDeploy implantação, incluindo Instalação, AfterInstallApplicationStart, e. ValidateService Durante o evento do ciclo de vida da BeforeInstallimplantação, um script será executado para remover arquivos antigos gravados durante implantações anteriores dessa amostra e criar um local na sua instância local para gravar os novos arquivos.

> **Note**
>
> A implantação dessa revisão de aplicativo de amostra poderá falhar se qualquer uma das seguintes condições for verdadeira:
>
> - O usuário que inicia o CodeDeploy agente na instância local não tem permissão para executar scripts.
> - O usuário não tem permissão para criar ou excluir pastas nas localizações listadas nos scripts.
> - O usuário não tem permissão para criar arquivos de texto nas localizações listadas nos scripts.

> **Note**
>
> Se você tiver configurado uma instância do Windows Server e quiser implantar uma amostra diferente, convém usar aquela em Etapa 2: configurar seu conteúdo de origem para implantar na instância do Amazon EC2 do Windows Server no tutorial Tutorial: implantar um aplicativo "Hello, World!" aplicativo com CodeDeploy (Windows Server).
> Se você tiver configurado uma instância do RHEL e quiser implantar uma amostra diferente, convém usar aquela em Etapa 2: configurar seu conteúdo de origem para ser implantado na instância Amazon Linux ou Red Hat Enterprise Linux instância do Amazon EC2 no tutorial Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix).
> Atualmente, não existe uma amostra alternativa para o Ubuntu Server.

1. Na sua máquina de desenvolvimento, crie um subdiretório (subpasta) chamado `CodeDeployDemo-OnPrem` que armazenará os arquivos da revisão de aplicativo de amostra e, em seguida, mude para essa subpasta. Para este exemplo, assumimos que você usará `c:\temp` como a pasta raiz para o Windows Server ou a pasta `/tmp` como a pasta raiz para o Ubuntu Server e o RHEL. Se usar uma pasta diferente, certifique-se de usá-la no lugar das pastas acima neste tutorial:

   Para Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem
   cd c:\temp\CodeDeployDemo-OnPrem
   ```

   Para Linux, macOS ou Unix:

   ```
   mkdir /tmp/CodeDeployDemo-OnPrem
   cd /tmp/CodeDeployDemo-OnPrem
   ```

2. Na raiz da subpasta `CodeDeployDemo-OnPrem`, use um editor de texto para criar os dois seguintes arquivos `appspec.yml` e `install.txt`:

   `appspec.yml` para Windows Server:

   ```
   version: 0.0
   os: windows
   files:
   ```

```
  - source: .\install.txt
    destination: c:\temp\CodeDeployExample
hooks:
  BeforeInstall:
    - location: .\scripts\before-install.bat
      timeout: 900
  AfterInstall:
    - location: .\scripts\after-install.bat
      timeout: 900
  ApplicationStart:
    - location: .\scripts\application-start.bat
      timeout: 900
  ValidateService:
    - location: .\scripts\validate-service.bat
      timeout: 900
```

`appspec.yml` para Ubuntu Server e RHEL:

```
version: 0.0
os: linux
files:
  - source: ./install.txt
    destination: /tmp/CodeDeployExample
hooks:
  BeforeInstall:
    - location: ./scripts/before-install.sh
      timeout: 900
  AfterInstall:
    - location: ./scripts/after-install.sh
      timeout: 900
  ApplicationStart:
    - location: ./scripts/application-start.sh
      timeout: 900
  ValidateService:
    - location: ./scripts/validate-service.sh
      timeout: 900
```

Para obter mais informações sobre AppSpec arquivos, consulte Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy CodeDeploy AppSpec referência de arquivo e.

`install.txt`:

```
The Install deployment lifecycle event successfully completed.
```

3. Na raiz da subpasta `CodeDeployDemo-OnPrem`, crie uma subpasta `scripts` e, em seguida, mude para ela:

   Para Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
   cd c:\temp\CodeDeployDemo-OnPrem\scripts
   ```

   Para Linux, macOS ou Unix:

   ```
   mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
   cd /tmp/CodeDeployDemo-OnPrem/scripts
   ```

4. Na raiz da subpasta `scripts`, use um editor de texto para criar os quatro seguintes arquivos `before-install.bat`, `after-install.bat`, `application-start.bat` e `validate-service.bat` para Windows Server, ou `before-install.sh`, `after-install.sh`, `application-start.sh` e `validate-service.sh` para Ubuntu Server e RHEL:

   Para Windows Server:

   `before-install.bat`:

   ```
   set FOLDER=%HOMEDRIVE%\temp\CodeDeployExample

   if exist %FOLDER% (
     rd /s /q "%FOLDER%"
   )

   mkdir %FOLDER%
   ```

   `after-install.bat`:

   ```
   cd %HOMEDRIVE%\temp\CodeDeployExample

   echo The AfterInstall deployment lifecycle event successfully completed. > after-
   install.txt
   ```

`application-start.bat`:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ApplicationStart deployment lifecycle event successfully completed. >
 application-start.txt
```

`validate-service.bat`:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ValidateService deployment lifecycle event successfully completed. >
 validate-service.txt
```

Para Ubuntu Server e RHEL:

`before-install.sh`:

```
#!/bin/bash
export FOLDER=/tmp/CodeDeployExample

if [ -d $FOLDER ]
then
 rm -rf $FOLDER
fi

mkdir -p $FOLDER
```

`after-install.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The AfterInstall deployment lifecycle event successfully completed." > after-
install.txt
```

`application-start.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample
```

```
echo "The ApplicationStart deployment lifecycle event successfully completed." >
 application-start.txt
```

`validate-service.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ValidateService deployment lifecycle event successfully completed." >
 validate-service.txt

unset FOLDER
```

5. Somente para o Ubuntu Server e o RHEL, certifique-se de que os quatro scripts de shell tenham permissões de execução:

```
chmod +x ./scripts/*
```

Etapa 3: agrupar e carregar sua revisão de aplicativo no Amazon S3

Para poder implantar sua revisão de aplicativo, você precisará agrupar os arquivos e depois carregar o pacote de arquivos em um bucket do Amazon S3. Siga as instruções em Crie um aplicativo com CodeDeploy e Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações). (Embora você possa dar qualquer nome ao aplicativo e ao grupo de implantação, recomendamos usar `CodeDeploy-OnPrem-App` para o nome do aplicativo e `CodeDeploy-OnPrem-DG` para o nome do grupo de implantação.) Depois de concluir essas instruções, retorna para esta página.

> **Note**
>
> Como alternativa, você pode carregar o pacote de arquivos em um GitHub repositório e implantá-lo a partir daí. Para obter mais informações, consulte Integrando com CodeDeploy GitHub.

Etapa 4: implantar sua revisão de aplicativo

Depois de carregar sua revisão de aplicativo em um bucket do Amazon S3, tente implantá-la na sua instância on-premises. Siga as instruções em Crie uma implantação com CodeDeploy e retorne para esta página.

Etapa 5: verificar sua implantação

Para verificar se a implantação foi bem-sucedida, siga as instruções em Exibir detalhes CodeDeploy da implantação e retorne para esta página.

Se a implantação foi bem-sucedida, você encontrará quatro arquivos de texto na pasta `c:\temp\CodeDeployExample` (para Windows Server) ou `/tmp/CodeDeployExample` (para Ubuntu Server e RHEL).

Se a implantação tiver falhado, siga as etapas de solução de problemas em View Instance Details e Solução de problemas de instância. Faça as correções necessárias, reagrupe e carregue sua revisão de aplicativo. Em seguida, tente a implantação novamente.

Etapa 6: limpar os recursos

Para evitar cobranças contínuas pelos recursos criados para este tutorial, exclua o bucket do Amazon S3 caso ele não seja mais utilizado. Você também pode limpar os recursos associados, como os registros do aplicativo e do grupo de implantação CodeDeploy e a instância local.

Você pode usar o AWS CLI ou uma combinação dos consoles Amazon S3 CodeDeploy e do Amazon S3 AWS CLI para limpar recursos.

Limpar recursos (CLI)

Para excluir o bucket do Amazon S3

- Chame o comando rm junto à opção `--recursive` para o bucket (por exemplo, `amzn-s3-demo-bucket`). O bucket e todos os objetos dentro dele serão excluídos.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Para excluir os registros do aplicativo e do grupo de implantação no CodeDeploy

- Chame o comando delete-application para o aplicativo (por exemplo, `CodeDeploy-OnPrem-App`). Os registros da implantação e do grupo de implantação serão excluídos.

```
aws deploy delete-application --application-name your-application-name
```

Para cancelar o registro da instância on-premises e excluir o usuário do IAM

- Chame o comando deregister para a instância on-premises e a região:

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --
region your-region
```

> **Note**
>
> Se não quiser excluir o usuário do IAM associado a essa instância on-premises, use a opção `--no-delete-iam-user` como alternativa.

Para desinstalar o CodeDeploy agente e remover o arquivo de configuração da instância local

- Na instância on-premises, chame o comando desinstalar:

```
aws deploy uninstall
```

Você concluiu todas as etapas para limpar os recursos usados para esse tutorial.

Limpar recursos (console)

Para excluir o bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/
2. Escolha o ícone ao lado do bucket que deseja excluir (por exemplo, `amzn-s3-demo-bucket`), mas não escolha o bucket propriamente dito.
3. Escolha Ações e, em seguida, escolha Excluir.

4. Quando solicitado a excluir o bucket, escolha OK.

Para excluir os registros do aplicativo e do grupo de implantação no CodeDeploy

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação selecione Applications (Aplicativos).
3. Escolha o nome do aplicativo que você deseja excluir (por exemplo, `CodeDeploy-OnPrem-App`) e escolha Delete application (Excluir aplicativo).
4. Quando solicitado, digite o nome do aplicativo para confirmar que você deseja excluí-lo e, em seguida, escolha Excluir.

Você não pode usar o AWS CodeDeploy console para cancelar o registro da instância local ou desinstalar o agente. CodeDeploy Siga as instruções em Para cancelar o registro da instância on-premises e excluir o usuário do IAM.

Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling

Neste tutorial, você usará CodeDeploy para implantar uma revisão do aplicativo em um grupo do Auto Scaling. O Amazon EC2 Auto Scaling lança instâncias do Amazon EC2 usando condições predefinidas e, em seguida, encerra essas instâncias quando elas não são mais necessárias. O Amazon EC2 Auto Scaling CodeDeploy pode ajudar a escalar garantindo que sempre tenha o número correto de instâncias do Amazon EC2 disponíveis para lidar com a carga das implantações. Para obter informações sobre a CodeDeploy integração do Amazon EC2 Auto Scaling com, consulte. Integração CodeDeploy com o Amazon EC2 Auto Scaling

Tópicos

- Pré-requisitos
- Etapa 1: criar e configurar o grupo do Auto Scaling

- Etapa 2: implantar o aplicativo no grupo do Auto Scaling
- Etapa 3: verificar os resultados
- Etapa 4: Aumentar o número de instâncias do Amazon EC2 no grupo do Auto Scaling.
- Etapa 5: verificar novamente os resultados
- Etapa 6: limpar

Pré-requisitos

Para acompanhar esse tutorial:

- Conclua todas as etapasComeçando com CodeDeploy, incluindo configurar AWS CLI e criar um perfil de instância do IAM (**`CodeDeployDemo-EC2-Instance-Profile`**) e uma função de serviço (**`CodeDeployDemo`**). Um perfil de serviço é um tipo especial de perfil do IAM; que dá a um serviço permissão para agir em seu nome.
- Se criou grupo do Auto Scaling com um modelo de execução, você deverá adicionar as seguintes permissões:
  - `ec2:RunInstances`
  - `ec2:CreateTags`
  - `iam:PassRole`

  Para obter mais informações sobre como criar modelos de execução, consulte Etapa 2: Criar um perfil de serviço, Criar um modelo de execução para um grupo do Auto Scaling e Executar suporte de modelo no Guia do usuário do Amazon EC2 Auto Scaling.
- Crie e use uma revisão compatível com uma instância do Ubuntu Server CodeDeploy e. Para a revisão, é possível fazer um dos seguintes procedimentos:
  - Crie e use a amostra de revisão Etapa 2: criar uma revisão de aplicativo de amostra no tutorial Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).
  - Crie uma revisão por conta própria. Consulte Trabalhando com revisões de aplicativos para CodeDeploy.
- Crie um grupo de segurança chamado **`CodeDeployDemo-AS-SG`** com a seguinte regra de entrada:
  - Tipo: HTTP
  - Origem: Qualquer lugar

Isso é necessário para visualizar seu aplicativo e verificar o sucesso da implantação. Para obter informações sobre como criar um grupo de segurança, consulte Criar um grupo de segurança no Guia do usuário do Amazon EC2.

Etapa 1: criar e configurar o grupo do Auto Scaling

Nesta etapa, você criará um grupo do Auto Scaling que contém uma única instância Amazon Linux, RHEL ou Windows Server Amazon EC2. Em uma etapa posterior, você instruirá o Amazon EC2 Auto Scaling a adicionar mais uma instância do Amazon EC2 CodeDeploy e implantará sua revisão nela.

Tópicos

- Como criar e configurar o grupo do Auto Scaling (CLI)
- Como criar e configurar o grupo do Auto Scaling (console)

Como criar e configurar o grupo do Auto Scaling (CLI)

1. Use o comando create-launch-template para criar um modelo de execução do Amazon EC2.

   Antes de chamar esse comando, você precisa do ID de uma AMI que funcione para este tutorial, representado pelo espaço *image-id* reservado. Você também precisa do nome de um par de chaves de instância do Amazon EC2 para permitir o acesso à instância do Amazon EC2, representado pelo espaço reservado. *key-name*

   Para obter o ID de uma AMI que funcione com este tutorial:

   a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
   b. No painel de navegação, em Instâncias, escolha Instâncias e, em seguida, escolha Iniciar instância.
   c. Na guia Início rápido da página Escolha uma imagem de máquina da Amazon, anote o ID da AMI ao lado do Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS ou Microsoft Windows Server 2012 R2.

      > **Note**
      >
      > Se você tiver uma versão personalizada de uma AMI compatível com o CodeDeploy, selecione-a aqui em vez de navegar pela guia Quick Start (Início

rápido). Para obter informações sobre o uso de uma AMI personalizada com CodeDeploy o Amazon EC2 Auto Usando uma AMI personalizada com o CodeDeploy Amazon EC2 Auto Scaling Scaling, consulte.

Para o par de chaves da instância do Amazon EC2, use o nome do par de chaves da sua instância do Amazon EC2.

Chame o comando create-launch-template.

Em máquinas Linux, macOS ou Unix locais:

```
aws ec2 create-launch-template \
  --launch-template-name CodeDeployDemo-AS-Launch-Template \
  --launch-template-data file://config.json
```

Visualize o conteúdo do arquivo `config.json`:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Em máquinas locais do Windows:

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-
Template --launch-template-data file://config.json
```

Visualize o conteúdo do arquivo `config.json`:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
```

```
  "KeyName":"key-name"
}
```

Esses comandos, junto com o `config.json` arquivo, criam um modelo de lançamento do Amazon EC2 com o nome do seu grupo CodeDeployDemo-AS-Launch-Template de Auto Scaling, que será criado na etapa seguinte com base no tipo de instância t1.micro do Amazon EC2. Com base em sua entrada para`ImageId`, `IamInstanceProfile`, e `KeyName`, o modelo de execução também especifica a ID da AMI, o nome do perfil de instância associado ao perfil do IAM a ser passado para as instâncias na execução e o par de chaves do Amazon EC2 a ser usado ao se conectar às instâncias.

2. Chame o comando create-auto-scaling-group para criar um grupo do Auto Scaling. Você precisará do nome de uma das zonas de disponibilidade em uma das regiões listadas em Região e dos endpoints na Referência geral da AWS, representados pelo espaço *`availability-zone`* reservado.

> **Note**
>
> Para ver uma lista de zonas de disponibilidade em uma região, ligue para:
>
> ```
> aws ec2 describe-availability-zones --region region-name
> ```
>
> Por exemplo, para ver uma lista de Zonas de disponibilidade na região Oeste dos EUA (Oregon), chame:
>
> ```
> aws ec2 describe-availability-zones --region us-west-2
> ```
>
> Para obter uma lista de identificadores de nomes de regiões, consulteNomes de bucket do kit de recursos por região.

Em máquinas Linux, macOS ou Unix locais:

```
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name CodeDeployDemo-AS-Group \
  --launch-template CodeDeployDemo-AS-Launch-Template,Version='$Latest' \
  --min-size 1 \
  --max-size 1 \
  --desired-capacity 1 \
```

```
  --availability-zones availability-zone \
  --tags Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Em máquinas locais do Windows:

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name
 CodeDeployDemo-AS-Group --launch-template LaunchTemplateName=CodeDeployDemo-
AS-Launch-Template,Version="$Latest" --min-size 1 --max-size 1 --
desired-capacity 1 --availability-zones availability-zone --tags
 Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Esses comandos criam um grupo do Auto Scaling chamado **CodeDeployDemo-AS-Group** com base no modelo de execução do Amazon EC2 chamado **CodeDeployDemo-AS-Launch-Template**. Esse grupo do Auto Scaling tem apenas uma instância do Amazon EC2 e é criado na Zona de disponibilidade especificada. Cada instância nesse grupo do Auto Scaling terá a tag `Name=CodeDeployDemo`. A tag será usada ao instalar o CodeDeploy agente posteriormente.

3. Chame o comando describe-auto-scaling-groups contra **CodeDeployDemo-AS-Group**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Não prossiga até que os valores retornados mostrem `Healthy` e `InService`.

4. As instâncias em seu grupo de Auto Scaling devem ter o CodeDeploy agente instalado para serem usadas em CodeDeploy implantações. Instale o CodeDeploy agente chamando o create-association comando AWS Systems Manager com as tags que foram adicionadas quando o grupo Auto Scaling foi criado.

```
aws ssm create-association \
  --name AWS-ConfigureAWSPackage \
  --targets Key=tag:Name,Values=CodeDeployDemo \

 --parameters action=Install, name=AWSCodeDeployAgent \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Esse comando cria uma associação no Systems Manager State Manager que instalará o CodeDeploy agente em todas as instâncias do grupo Auto Scaling e tentará atualizá-lo às 2h da manhã de domingo. Para obter mais informações sobre o CodeDeploy agente, consulte

Trabalhando com o CodeDeploy agente. Para obter mais informações sobre o Systems Manager, consulte O que é o AWS Systems Manager.

Como criar e configurar o grupo do Auto Scaling (console)

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Na barra de navegação global, verifique se uma das regiões listadas em Região e os endpoints na Referência geral da AWSestão selecionados. Os recursos do Amazon EC2 Auto Scaling estão vinculados à região que você especifica CodeDeploy e são suportados somente em regiões selecionadas.
3. Na barra de navegação, escolha Instances e, em seguida, Launch Templates.
4. Escolha Criar modelo de execução.
5. Na caixa de diálogo Nome e descrição do modelo de execução, para Nome do modelo de execução, insira **`CodeDeployDemo-AS-Launch-Template`**. Mantenha os valores padrão em todos os outros campos.
6. Na caixa de diálogo Imagem de máquina da Amazon (AMI), clique no menu suspenso em AMI e escolha uma AMI que funcione com este tutorial:

   - Na guia Início rápido do menu suspenso AMI, escolha uma das seguintes opções: Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS ou Microsoft Windows Server 2012 R2.

     > **Note**
     >
     > Se você tiver uma versão personalizada de uma AMI compatível com o CodeDeploy, selecione-a aqui em vez de navegar pela guia Quick Start (Início rápido). Para obter informações sobre o uso de uma AMI personalizada com CodeDeploy o Amazon EC2 Auto Usando uma AMI personalizada com o CodeDeploy Amazon EC2 Auto Scaling Scaling, consulte.

7. Em Tipo de instância, selecione a lista suspensa e escolha t1.micro. É possível usar a barra de pesquisa para encontrá-la mais rapidamente.
8. Na caixa de diálogo Par de chaves (login), selecione Escolher um par de chaves existente. Na lista suspensa Selecione um par de chaves, selecione o par de chaves de instância do Amazon EC2 criado ou usado nas etapas anteriores.
9. Na caixa de diálogo Configurações de rede, escolha Virtual Public Cloud (VPC).

No menu suspenso Grupos de segurança, escolha o grupo de segurança que você criou na seção de pré-requisitos do tutorial (**CodeDeployDemo-AS-SG**).

10. Expanda a caixa de diálogo Detalhes avançados. No menu suspenso Perfil de instância do IAM, selecione o perfil do IAM que você criou anteriormente (**CodeDeployDemo-EC2-Instance-Profile**) em Perfil de instância do IAM.

    Deixe o restante dos padrões.

11. Escolha Criar modelo de execução.
12. Na caixa de diálogo Próximas etapas, escolha Criar grupo do Auto Scaling.
13. Na página Escolher modelo ou configuração de execução, para Nome do grupo do Auto Scaling, insira **CodeDeployDemo-AS-Group**.
14. Na caixa de diálogo Modelo de execução, seu modelo de execução (**CodeDeployDemo-AS-Launch-Template**) deve ser preenchido; caso contrário, selecione-o no menu suspenso. Deixe os padrões e escolha Próximo.
15. Na página Escolher as opções de execução da instância na seção Rede, para VPC, selecione a VPC padrão. Em seguida, para Zonas de disponibilidade e sub-redes, escolha uma sub-rede padrão. Você deve criar uma VPC se não puder escolher a padrão. Para obter mais informações, consulte Conceitos básicos do Amazon VPC.
16. Na seção Instance type requirements (Requisitos de tipo de instância), use a configuração padrão para simplificar essa etapa. (Não substitua o modelo de execução.) Neste tutorial, você executará somente On-Demand instâncias usando o tipo de instância especificado em seu modelo de execução.
17. Selecione Next (Próximo) para ir até a página Configure advanced options (Configurar opções avançadas).
18. Mantenha os outros valores padrão e escolha Próximo.
19. Na página Configurar tamanho do grupo e políticas de escalabilidade, mantenha os valores padrão de tamanho do grupo de 1. Escolha Próximo.
20. Ignore a etapa de configuração de notificações e escolha Próximo.
21. Na página Adicionar tags, adicione uma tag a ser usada ao instalar o CodeDeploy agente posteriormente. Escolha Adicionar Tag.

    a. Em Chave, insira **Name**.

    b. Em Valor, informe **CodeDeployDemo**.

Escolha Próximo.

22. Na página Revisar, reveja as configurações do grupo do Auto Scaling e escolha Criar grupo do Auto Scaling.
23. Na barra de navegação, com a opção Grupos do Auto Scaling selecionada, escolha **`CodeDeployDemo-AS-Group`** e, em seguida, a guia Gerenciamento de instâncias. Não continue até que o valor de InServiceapareça na coluna Lifecycle e o valor de Healthy apareça na coluna Health Status.
24. Instale o CodeDeploy agente seguindo as etapas em Instalar o CodeDeploy agente e usando as tags da `Name=CodeDeployDemo` instância.

Etapa 2: implantar o aplicativo no grupo do Auto Scaling

Nesta etapa, você implantará a revisão na instância única do Amazon EC2 no grupo do Auto Scaling.

Tópicos

- Para criar a implantação (CLI)
- Para criar a implantação (console)

Para criar a implantação (CLI)

1. Chame o comando create-application para criar um aplicativo chamado **`SimpleDemoApp`**:

```
aws deploy create-application --application-name SimpleDemoApp
```

2. Você já deve ter criado um perfil de serviço seguindo as instruções em Etapa 2: criar uma função de serviço para CodeDeploy. A função de serviço dará CodeDeploy permissão para acessar suas instâncias do Amazon EC2 para expandir (ler) suas tags. Você precisará do ARN da função de serviço. Para obter o ARN do perfil de serviço, siga as instruções em Obter o ARN do perfil de serviço (CLI).
3. Agora que você tem um ARN de perfil de serviço, chame o comando create-deployment-group para criar um grupo de implantação chamado **`SimpleDemoDG`**, associado ao aplicativo chamado **`SimpleDemoApp`**, usando o grupo do Auto Scaling chamado **`CodeDeployDemo-AS-Group`** e a configuração de implantação chamada **`CodeDeployDefault.OneAtATime`**, com o ARN de perfil de serviço especificado.

> **Note**
>
> O comando create-deployment-group fornece suporte para a criação de acionadores que resultam no envio de notificações do Amazon SNS aos assinantes do tópico sobre eventos específicos em implantações e instâncias. O comando também oferece suporte a opções para reverter automaticamente as implantações e configurar alarmes para interromper as implantações quando os limites de monitoramento nos alarmes da CloudWatch Amazon forem atingidos. Os comandos para essas ações não estão incluídos neste tutorial.

Em máquinas Linux, macOS ou Unix locais:

```
aws deploy create-deployment-group \
  --application-name SimpleDemoApp \
  --auto-scaling-groups CodeDeployDemo-AS-Group \
  --deployment-group-name SimpleDemoDG \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --service-role-arn service-role-arn
```

Em máquinas locais do Windows:

```
aws deploy create-deployment-group --application-name SimpleDemoApp --auto-scaling-
groups CodeDeployDemo-AS-Group --deployment-group-name SimpleDemoDG --deployment-
config-name CodeDeployDefault.OneAtATime --service-role-arn service-role-arn
```

4. Chame o comando create-deployment para criar uma implantação associada ao aplicativo **`SimpleDemoApp`**, a configuração de implantação **`CodeDeployDefault.OneAtATime`**, o grupo de implantação **`SimpleDemoDG`**, usando a revisão na localização especificada.

   Para instâncias Amazon Linux e RHEL do Amazon EC2, chamadas de máquinas Linux, macOS ou Unix locais

```
aws deploy create-deployment \
  --application-name SimpleDemoApp \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --deployment-group-name SimpleDemoDG \
```

```
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/
SampleApp_Linux.zip
```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit para sua região. Por exemplo, para a região Leste dos EUA (Ohio), *bucket-name* substitua por`aws-codedeploy-us-east-2`. Para obter uma lista de nomes de buckets, consulte Nomes de bucket do kit de recursos por região.

Para instâncias do Amazon Linux e RHEL do Amazon EC2, chamadas de máquinas Windows locais

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-
name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-
location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Linux.zip
```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit para sua região. Por exemplo, para a região Leste dos EUA (Ohio), *bucket-name* substitua por`aws-codedeploy-us-east-2`. Para obter uma lista de nomes de buckets, consulte Nomes de bucket do kit de recursos por região.

Para instâncias do Windows Server Amazon EC2, chamadas de máquinas Linux, macOS ou Unix locais

```
aws deploy create-deployment \
  --application-name SimpleDemoApp \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --deployment-group-name SimpleDemoDG \
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/
SampleApp_Windows.zip
```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit para sua região. Por exemplo, para a região Leste dos EUA (Ohio), *bucket-name* substitua por`aws-codedeploy-us-east-2`. Para obter uma lista de nomes de buckets, consulte Nomes de bucket do kit de recursos por região.

Para instâncias do Windows Server Amazon EC2, chamadas de máquinas Windows locais

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-
name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-
location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Windows.zip
```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit para sua região. Por exemplo, para a região Leste dos EUA (Ohio), *bucket-name* substitua por`aws-codedeploy-us-east-2`. Para obter uma lista de nomes de buckets, consulte Nomes de bucket do kit de recursos por região.

> **Note**
>
> Atualmente, CodeDeploy não fornece uma revisão de amostra para implantação em instâncias do Amazon EC2 do Ubuntu Server. Para criar uma revisão por conta própria, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

5. Chame o comando get-deployment para verificar se a implantação foi bem-sucedida.

   Antes de chamar esse comando, será necessário o ID de implantação, que deve ter sido retornado pela chamada para o comando create-deployment. Se precisar obter o ID de implantação novamente, chame o comando list-deployments contra o aplicativo chamado **`SimpleDemoApp`** e o grupo de implantação chamado **`SimpleDemoDG`**.

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Agora, chame o comando get-deployment usando o ID de implantação:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query
    "deploymentInfo.status" --output text
   ```

   Não continue até que o valor retornado seja `Succeeded`.

Para criar a implantação (console)

1. Você já deve ter criado um perfil de serviço seguindo as instruções em Etapa 2: criar uma função de serviço para CodeDeploy. A função de serviço dará CodeDeploy permissão para acessar suas instâncias para expandir (ler) suas tags. Antes de usar o CodeDeploy console

para implantar a revisão do aplicativo, você precisará do ARN da função de serviço. Para obter o ARN do perfil de serviço, siga as instruções em Obter o ARN do perfil de serviço (console).

2. Agora que você tem o ARN da função de serviço, você pode usar o CodeDeploy console para implantar a revisão do seu aplicativo.

   Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

3. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
4. Escolha Criar aplicativo.
5. Selecione Custom application (Aplicativo personalizado).
6. Em Nome do aplicativo, insira **`SimpleDemoApp`**.
7. Na plataforma de computação, escolha EC2/On-premises.
8. Escolha Criar aplicativo.
9. Na guia Grupos de implantação, selecione Criar um grupo de implantação.
10. Em Nome do grupo de implantação, insira **`SimpleDemoDG`**.
11. Em Service Role (Função do serviço), selecione o nome da função do serviço.
12. Em Tipo de implantação, escolha In-place.
13. Em Configuração do ambiente, selecione Grupos do Auto Scaling e escolha **`CodeDeployDemo-AS-Group`**.
14. Em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.OneAtATime.
15. Desmarque Enable load balancing (Habilitar o balanceamento de carga).
16. Selecione Criar grupo de implantação.
17. Na página do grupo de implantação, selecione Create deployment (Criar implantação).
18. Em Tipo de revisão, escolha Meu aplicativo está armazenado no Amazon S3.
19. Em Local da revisão, insira o local do aplicativo de amostra para seu sistema operacional e sua região.

Para instâncias Amazon Linux e RHEL do Amazon EC2

Região	Local do aplicativo de amostra
Região Leste dos EUA (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip`
Região Leste dos EUA (Norte da Virgínia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip`
Região Oeste dos EUA (Norte da Califórnia)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip`
Região Oeste dos EUA (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip`
Região Canadá (Central)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip`
Região Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip`
Região Europa (Londres)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip`

Região	Local do aplicativo de amostra
Região Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip`
Região Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip`
Região de Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Hong Kong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Tóquio)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Seul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip`

Região	Local do aplicativo de amostra
Região Ásia-Pacífico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Região Ásia-Pacífico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip`
Região América do Sul (São Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip`

Para instâncias do Windows Server Amazon EC2

Região	Local do aplicativo de amostra
Região Leste dos EUA (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip`
Região Leste dos EUA (Norte da Virgínia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip`

Região	Local do aplicativo de amostra
Região Oeste dos EUA (Norte da Califórnia)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip`
Região Oeste dos EUA (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip`
Região Canadá (Central)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip`
Região Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip`
Região Europa (Londres)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip`
Região Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip`
Região Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip`

Região	Local do aplicativo de amostra
Região de Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Hong Kong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Seul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Região Ásia-Pacífico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip`

Região	Local do aplicativo de amostra
Região América do Sul (São Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip`

Para instâncias do Ubuntu Server Amazon EC2

Digite o local da sua revisão de aplicativo personalizada armazenada no Amazon S3.

20. Deixe Descrição da implantação em branco.
21. Expanda Advanced.
22. Escolha Criar implantação.

> **Note**
>
> Se Falha aparecer no lugar de Bem-sucedido, convém tentar algumas das técnicas em Monitorar sua implantação e solucionar problemas relacionados (usando o nome de aplicativo **`SimpleDemoApp`** e o nome do grupo de implantação **`SimpleDemoDG`**).

Etapa 3: verificar os resultados

Nesta etapa, você verificará se CodeDeploy instalou a **`SimpleDemoApp`** revisão em uma única instância do Amazon EC2 no grupo Auto Scaling.

Tópicos

- Para verificar os resultados (CLI)
- Para verificar os resultados (console)

Para verificar os resultados (CLI)

Primeiro, você precisará do DNS público da instância do Amazon EC2.

Use o AWS CLI para obter o DNS público da instância do Amazon EC2 no grupo Auto Scaling chamando o comando. describe-instances

Antes de chamar esse comando, você precisará da ID da instância do Amazon EC2. Para obter o ID, chame describe-auto-scaling-groups contra **CodeDeployDemo-AS-Group**, conforme feito anteriormente:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Agora chame o comando describe-instances:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

O valor retornado é o DNS público da instância do Amazon EC2.

Usando um navegador da web, mostre a SimpleDemoApp revisão implantada nessa instância do Amazon EC2, usando uma URL como a seguinte:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Se você ver a página de parabéns, você usou com sucesso CodeDeploy a implantação de uma revisão em uma única instância do Amazon EC2 em um grupo de Auto Scaling!

Em seguida, adicione uma instância do Amazon EC2 ao grupo do Auto Scaling. Depois que o Amazon EC2 Auto Scaling adicionar a instância do Amazon EC2, implantará sua revisão na nova CodeDeploy instância.

Para verificar os resultados (console)

Primeiro, você precisará do DNS público da instância do Amazon EC2.

Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.

No painel de navegação do Amazon EC2, em Auto Scaling, selecione Grupos do Auto Scaling e, em seguida, a entrada. **CodeDeployDemo-AS-Group**.

Na guia Instâncias, escolha a ID da instância do Amazon EC2 na lista.

Na página Instâncias, na guia Descrição, anote o valor DNS público. A aparência será semelhante à seguinte: **ec2-01-234-567-890.compute-1.amazonaws.com**.

Usando um navegador da web, mostre a SimpleDemoApp revisão implantada nessa instância do Amazon EC2, usando uma URL como a seguinte:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Se você ver a página de parabéns, você usou com sucesso CodeDeploy a implantação de uma revisão em uma única instância do Amazon EC2 em um grupo de Auto Scaling!

Em seguida, adicione uma instância do Amazon EC2 ao grupo do Auto Scaling. Depois que o Amazon EC2 Auto Scaling adicionar a instância do Amazon EC2, implantará sua revisão na nova CodeDeploy instância do Amazon EC2.

Etapa 4: Aumentar o número de instâncias do Amazon EC2 no grupo do Auto Scaling.

Nesta etapa, você instrui o grupo do Auto Scaling para criar uma instância adicional do Amazon EC2. Depois que o Amazon EC2 Auto Scaling criar a CodeDeploy instância, implanta sua revisão nela.

Tópicos

- Para aumentar a escala horizontalmente o número de instâncias do Amazon EC2 no grupo do Auto Scaling (CLI)
- Para aumentar a escala horizontalmente o número de instâncias do Amazon EC2 no grupo de implantação (console)

Para aumentar a escala horizontalmente o número de instâncias do Amazon EC2 no grupo do Auto Scaling (CLI)

1. Chame o comando update-auto-scaling-group para aumentar as instâncias do Amazon EC2 no grupo do Auto Scaling chamado **`CodeDeployDemo-AS-Group`** de um para dois.

   Em máquinas Linux, macOS ou Unix locais:

   ```
   aws autoscaling update-auto-scaling-group \
     --auto-scaling-group-name CodeDeployDemo-AS-Group \
     --min-size 2 \
     --max-size 2 \
     --desired-capacity 2
   ```

Em máquinas locais do Windows:

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-
AS-Group --min-size 2 --max-size 2 --desired-capacity 2
```

2. Verifique se o grupo do Auto Scaling possui duas instâncias do Amazon EC2. Chame o comando describe-auto-scaling-groups contra **`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Não prossiga até que ambos os valores retornados mostrem `Healthy` e `InService`.

Para aumentar a escala horizontalmente o número de instâncias do Amazon EC2 no grupo de implantação (console)

1. No painel de navegação do Amazon EC2, em Auto Scaling selecione Grupos do Auto Scaling e, em seguida, a entrada. **`CodeDeployDemo-AS-Group`**.
2. Escolha Ações e, em seguida, escolha Editar.
3. Na guia Details (Detalhes), nas caixas Desired (Desejado), Min (Mín.) e Max (Máx.), digite **2** e, em seguida, selecione Save (Salvar).
4. Escolha a guia Instâncias. A nova instância do Amazon EC2 aparecerá na lista. (Se a instância não aparecer, talvez seja necessário escolher o botão Atualizar algumas vezes). Não continue até que o valor de InServiceapareça na coluna Lifecycle e o valor de Healthy apareça na coluna Health Status.

Etapa 5: verificar novamente os resultados

Nesta etapa, você verificará se a SimpleDemoApp revisão foi CodeDeploy instalada na nova instância no grupo Auto Scaling.

Tópicos

- Para verificar os resultados da implantação automática (CLI)
- Para verificar os resultados da implantação automática (console)

Para verificar os resultados da implantação automática (CLI)

1. Antes de chamar o comando get-deployment, será necessário o ID de implantação automática. Para obter o ID, chame o comando list-deployments contra o aplicativo chamado **`SimpleDemoApp`** e o grupo de implantação chamado **`SimpleDemoDG`**.

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Deve haver dois ID de implantação. Use aquele que você ainda não usou em uma chamada para o comando get-deployment:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
   [status, creator]" --output text
   ```

   Além do status de implantação, você verá `autoScaling` na saída do comando. (`autoScaling` significa que o Amazon EC2 Auto Scaling criou a implantação.)

   Não prossiga até que o estado da implantação mostre `Succeeded`.

2. Antes de chamar o comando describe-instances, você precisará da ID da nova instância do Amazon EC2. Para obter este ID, faça outra chamada para o comando describe-auto-scaling-groups contra **`CodeDeployDemo-AS-Group`**:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
    CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --
   output text
   ```

   Agora faça uma chamada para o comando describe-instances:

   ```
   aws ec2 describe-instances --instance-id instance-id --query
    "Reservations[0].Instances[0].PublicDnsName" --output text
   ```

   Na saída do comando describe-instances, anote o DNS público da nova instância do Amazon EC2.

3. Usando um navegador da Web, mostre a revisão do `SimpleDemoApp` implantada nessa instância do Amazon EC2 usando uma URL como a seguinte:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Se a página de parabéns aparecer, você costumava CodeDeploy implantar uma revisão em uma instância ampliada do Amazon EC2 em um grupo de Auto Scaling!

Para verificar os resultados da implantação automática (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Implantações.

3. Selecione a ID de implantação da implantação que o Amazon EC2 Auto Scaling criou.

   .

4. A página Deployment (Implantação) exibe informações sobre a implantação. Normalmente, você criaria uma implantação por conta própria, mas o Amazon EC2 Auto Scaling criou uma para você, para implantar sua revisão na nova instância do Amazon EC2.
5. Depois que Succeeded (Bem-sucedida) for exibido na parte superior da página, verifique os resultados na instância. Em primeiro lugar, é necessário obter o DNS público da instância:
6. No painel de navegação do Amazon EC2, em Auto Scaling, selecione Grupos do Auto Scaling e, em seguida, a entrada. **`CodeDeployDemo-AS-Group`**.
7. Na guia Instâncias, escolha a ID da nova instância do Amazon EC2.
8. Na página Instâncias, na guia Descrição, anote o valor DNS público. A aparência será semelhante à seguinte: **`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Mostre a revisão de `SimpleDemoApp` implantada na instância usando uma URL como a seguinte:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Se a página de parabéns aparecer, você costumava CodeDeploy implantar uma revisão em uma instância ampliada do Amazon EC2 em um grupo de Auto Scaling!

Etapa 6: limpar

Nesta etapa, você excluirá o grupo do Auto Scaling para evitar cobranças contínuas pelos recursos usados durante este tutorial. Opcionalmente, você pode excluir a configuração do Auto Scaling CodeDeploy e os registros do componente de implantação.

Tópicos

- Para limpar recursos (CLI)
- Para limpar recursos (console)

Para limpar recursos (CLI)

1. Excluir o grupo do Auto Scaling chamando o comando delete-auto-scaling-group contra **`CodeDeployDemo-AS-Group`**. Isso também terminará as instâncias do Amazon EC2.

   ```
   aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --force-delete
   ```

2. Opcionalmente, exclua a configuração de execução do Auto Scaling chamando o comando delete-launch-template contra a configuração de execução chamada **`CodeDeployDemo-AS-Launch-Template`**:

   ```
   aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template
   ```

3. Opcionalmente, exclua o aplicativo CodeDeploy chamando o delete-application comando no aplicativo chamado**`SimpleDemoApp`**. Isso também excluirá todos os registros associados de implantação, grupo de implantação e revisão.

   ```
   aws deploy delete-application --application-name SimpleDemoApp
   ```

4. Para excluir a associação do gerenciador de estados do Systems Manager, chame o comando delete-association.

   ```
   aws ssm delete-association --assocation-id association-id
   ```

Você pode obter o *association-id* chamando o describe-association comando.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets
 Key=tag:Name,Values=CodeDeployDemo
```

Para limpar recursos (console)

Para excluir o grupo do Auto Scaling, que também encerra as instâncias do Amazon EC2:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação do Amazon EC2, em Auto Scaling, selecione Grupos do Auto Scaling e, em seguida, a entrada. **CodeDeployDemo-AS-Group**.
3. Escolha Ações, Excluir e, em seguida, Sim, excluir.

(Opcional) Excluir o modelo de execução:

1. Na barra de navegação, em Ajuste de escala automático, selecione Configurações de execução e, em seguida, escolha **CodeDeployDemo-AS-Launch-Template**.
2. Escolha Ações, Excluir configuração de execução e, em seguida, Sim, excluir.

1. Opcionalmente, exclua o aplicativo de CodeDeploy. Isso também excluirá todos os registros associados de implantação, grupo de implantação e revisão. Abra o console do CodeDeploy em https://console.aws.amazon.com/codedeploy.
2. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

No painel de navegação, expanda a opção Implantar e escolha Aplicações.

3. Na lista de aplicativos, selecione SimpleDemoApp.
4. Na página Detalhes da aplicação, escolha Excluir aplicativo.
5. Quando solicitado, digite **`Delete`** e escolha Excluir.

Para excluir a associação do gerenciador de estados do Systems Manager:

1. Abra o AWS Systems Manager console em https://console.aws.amazon.com/systems-manager.
2. No painel de navegação, escolha State Manager.
3. Escolha a associação criada e selecione Excluir.

Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub

Neste tutorial, você usa CodeDeploy para implantar uma amostra de revisão do GitHub aplicativo em uma única instância do Amazon EC2 executando o Amazon Linux, uma única instância do Red Hat Enterprise Linux (RHEL) ou uma única instância do Windows Server. Para obter informações sobre GitHub integração com CodeDeploy, consulteIntegrando com CodeDeploy GitHub.

> **Note**
>
> Você também pode usar CodeDeploy para implantar uma revisão do aplicativo GitHub em uma instância do Ubuntu Server. Você pode usar a revisão de amostra descrita Etapa 2: criar uma revisão de aplicativo de amostra emTutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux), ou você pode criar uma revisão compatível com uma instância do Ubuntu Server CodeDeploy e. Para criar sua própria revisão, consulte Planeje uma revisão para CodeDeploy e Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

Tópicos

- Pré-requisitos
- Etapa 1: configurar uma GitHub conta
- Etapa 2: criar um GitHub repositório
- Etapa 3: Carregar um aplicativo de amostra no seu GitHub repositório

- Etapa 4: provisionar uma instância
- Etapa 5: criar um aplicativo e um grupo de implantação
- Etapa 6: implantar o aplicativo na instância
- Etapa 7: monitorar e verificar a implantação
- Etapa 8: Limpeza

Pré-requisitos

Antes de começar este tutorial, faça o seguinte:

- Instale o Git na sua máquina local. Para instalar o Git, consulte Downloads do Git.
- Conclua as etapas em Começando com CodeDeploy, incluindo a instalação e a configuração da AWS CLI. Isso é especialmente importante se você quiser usar o AWS CLI para implantar uma revisão na instância. GitHub

Etapa 1: configurar uma GitHub conta

Você precisará de uma GitHub conta para criar um GitHub repositório onde a revisão será armazenada. Se você já tem uma GitHub conta, vá paraEtapa 2: criar um GitHub repositório.

1. Acesse https://github.com/join.
2. Digite um nome de usuário, seu endereço de e-mail e uma senha.
3. Escolha Inscrever-se GitHub em e siga as instruções.

Etapa 2: criar um GitHub repositório

Você precisará de um GitHub repositório para armazenar a revisão.

Se você já tem um GitHub repositório, certifique-se de substituir seu nome por **`CodeDeployGitHubDemo`** todo este tutorial e, em seguida, vá para. Etapa 3: Carregar um aplicativo de amostra no seu GitHub repositório

1. Na página GitHub inicial, faça o seguinte:
   - Em Seus repositórios, escolha Novo repositório.
   - Na barra de navegação, escolha Criar novo (+) e, em seguida, escolha Novo repositório.

2. Na página Create a new repository, faça o seguinte:

   - Na caixa de texto Repository name (Nome do repositório), digite **`CodeDeployGitHubDemo`**.
   - Selecione Public.

   > **Note**
   >
   > Selecionar a opção Público padrão significa que qualquer pessoa pode ver esse repositório. Você pode selecionar a opção Private (Privado) para limitar quem pode ver e confirmar no repositório.

   - Desmarque a caixa de seleção Inicializar este repositório com um LEIA-ME. Você criará um arquivo `README.md` manualmente na próxima etapa.
   - Escolha Criar repositório.

3. Siga as instruções para o seu tipo de máquina local de modo a usar a linha de comando para criar o repositório.

   > **Note**
   >
   > Se você ativou a autenticação de dois fatores GitHub, certifique-se de inserir seu token de acesso pessoal em vez de sua senha de GitHub login, caso seja solicitada uma senha. Para obter informações, consulte Como fornecer seu código de autenticação de dois fatores.

Em máquinas Linux, macOS ou Unix locais:

1. No terminal, execute os seguintes comandos, um por vez, onde *`user-name`* está seu nome de GitHub usuário:

```
mkdir /tmp/CodeDeployGitHubDemo
```

```
cd /tmp/CodeDeployGitHubDemo
```

```
touch README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```

```
git push -u origin master
```

2. Deixe o terminal aberto na localização `/tmp/CodeDeployGitHubDemo`.

Em máquinas locais do Windows:

1. Em um prompt de comando em execução como administrador, execute os seguintes comandos, um por vez:

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. No bloco de notas, salve o arquivo `README.md`. Feche o Bloco de notas. Execute os seguintes comandos, um por vez, onde *user-name* está seu nome de GitHub usuário:

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```

```
git push -u origin master
```

3. Deixe a solicitação de comando aberta na localização `c:\temp\CodeDeployGitHubDemo`.

Etapa 3: Carregar um aplicativo de amostra no seu GitHub repositório

Nesta etapa, você copiará uma revisão de amostra de um bucket público do Amazon S3 para o seu GitHub repositório. (Para simplificar, as revisões de amostra fornecidas para este tutorial são páginas da Web exclusivas.)

> **Note**
>
> Se você usar uma das suas revisões em vez da nossa revisão de amostra, sua revisão deverá:
>
> - Seguir as diretrizes em Planeje uma revisão para CodeDeploy e Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.
> - Trabalhar com o tipo de instância correspondente.
> - Esteja acessível a partir do seu GitHub painel.
>
> Se a sua revisão atender a esses requisitos, avance para Etapa 5: criar um aplicativo e um grupo de implantação.
> Se você estiver implantando em uma instância do Ubuntu Server, precisará enviar para o seu GitHub repositório uma revisão compatível com uma instância do Ubuntu Server e. CodeDeploy Para obter mais informações, consulte Planeje uma revisão para CodeDeploy e Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

Tópicos

- Envie uma amostra de revisão com uma máquina Linux, macOS ou Unix local
- Enviar uma revisão de amostra de uma máquina Windows local

Envie uma amostra de revisão com uma máquina Linux, macOS ou Unix local

Com seu terminal ainda aberto, por exemplo, na localização `/tmp/CodeDeployGitHubDemo`, execute os seguintes comandos, um de cada vez:

> **Note**
>
> Se você planeja implantar em uma instância do Windows Server, substitua `SampleApp_Windows.zip` por `SampleApp_Linux.zip` nos comandos.

```
(Amazon S3 copy command)
```

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Onde *(Amazon S3 copy command)* está uma das seguintes opções:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2` para a região Leste dos EUA (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1` para a região Leste dos EUA (Norte da Virgínia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1` para a região Oeste dos EUA (Norte da Califórnia)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2` para a região Oeste dos EUA (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1` para a região do Canadá (Central)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`para a região da Europa (Irlanda)

- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`para a região da Europa (Londres)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`para a região da Europa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1` para a região da Europa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1` para a região de Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`para a região Ásia-Pacífico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`para a região Ásia-Pacífico (Tóquio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2`para a região Ásia-Pacífico (Seul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1` para a região Ásia-Pacífico (Singapura)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2`para a região Ásia-Pacífico (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4`para a região Ásia-Pacífico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1`para a região Ásia-Pacífico (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1` para a região da América do Sul (São Paulo)

Enviar uma revisão de amostra de uma máquina Windows local

Com o prompt de comando ainda aberto, por exemplo, no `c:\temp\CodeDeployGitHubDemo` local, execute os seguintes comandos, um por vez:

> **Note**
>
> Se você planeja implantar em uma instância do Amazon Linux ou RHEL, substitua `SampleApp_Linux.zip` por `SampleApp_Windows.zip` nos comandos.

```
(Amazon S3 copy command)
```

Descompacte o conteúdo do arquivo ZIP `the` diretamente no diretório local (por exemplo `c:\temp\CodeDeployGitHubDemo`), e não em um novo subdiretório.

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Onde *(Amazon S3 copy command)* está uma das seguintes opções:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2` para a região Leste dos EUA (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1` para a região Leste dos EUA (Norte da Virgínia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1` para a região Oeste dos EUA (Norte da Califórnia)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2` para a região Oeste dos EUA (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1` para a região do Canadá (Central)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1`para a região da Europa (Irlanda)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2`para a região da Europa (Londres)

- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3`para a região da Europa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1` para a região da Europa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1` para a região de Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1`para a região Ásia-Pacífico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1`para a região Ásia-Pacífico (Tóquio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2`para a região Ásia-Pacífico (Seul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1` para a região Ásia-Pacífico (Singapura)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2`para a região Ásia-Pacífico (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4`para a região Ásia-Pacífico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1`para a região Ásia-Pacífico (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1` para a região da América do Sul (São Paulo)

Para enviar sua própria revisão a uma instância do Ubuntu Server, copie-a no seu repositório local e depois chame o seguinte:

```
git add .
git commit -m "Added Ubuntu app"
git push
```

Etapa 4: provisionar uma instância

Nesta etapa, você criará ou configurará a instância na qual implantará o aplicativo de amostra. Você pode implantar em uma instância do Amazon EC2 ou em uma instância local que esteja executando um dos sistemas operacionais suportados pelo. CodeDeploy Para obter mais informações, consulte Sistemas operacionais suportados pelo CodeDeploy agente. (Se você já tiver uma instância configurada para uso em CodeDeploy implantações, vá para a próxima etapa.)

Como provisionar uma instância

1. Siga as instruções em Iniciar uma instância do Amazon EC2 (console) para provisionar uma instância.
2. Ao iniciar a instância, lembre-se de especificar uma tag na página Adicionar tags. Para obter detalhes sobre como especificar a tag, consulte Iniciar uma instância do Amazon EC2 (console).

Para verificar se o CodeDeploy agente está sendo executado na instância

- Siga as instruções em Verifique se o CodeDeploy agente está em execução para verificar se o agente está em execução.

Depois de provisionar a instância com sucesso e verificar se o CodeDeploy agente está em execução, vá para a próxima etapa.

Etapa 5: criar um aplicativo e um grupo de implantação

Nesta etapa, você usará o CodeDeploy console ou o AWS CLI para criar um aplicativo e um grupo de implantação para implantar a revisão de amostra do seu GitHub repositório.

Criar um aplicativo e um grupo de implantação (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.

3. Escolha Create application (Criar aplicativo) e selecione Custom application (Aplicativo personalizado).
4. Em Nome do aplicativo, insira **`CodeDeployGitHubDemo-App`**.
5. Em Compute Platform, escolha EC2/On-premises.
6. Escolha Criar aplicativo.
7. Na guia Grupos de implantação, selecione Criar um grupo de implantação.
8. Em Nome do grupo de implantação, insira **`CodeDeployGitHubDemo-DepGrp`**.
9. Em Função de serviço, escolha o nome da sua função de CodeDeploy serviço que você criou em Criar uma função de serviço para CodeDeploy.
10. Em Tipo de implantação, escolha In-place.
11. Em Configuração do ambiente, dependendo do tipo de instância que você está usando, escolha instâncias ou On-premises instâncias do Amazon EC2. Em Key (Chave) e Value (Valor), digite a chave de tag da instância e o valor que foi aplicado à instância como parte de Etapa 4: provisionar uma instância.
12. Em Deployment configuration (Configuração de implantação), selecione CodeDeployDefault.AllatOnce.
13. Em Balanceador de carga, desmarque Habilitar balanceamento de carga.
14. Expanda Advanced.
15. Em Alarms (Alarmes), selecione Ignore alarm configuration (Ignorar configuração de alarme).
16. Selecione Create deployment group (Criar grupo de implantação) e continue na próxima etapa.

Criar um aplicativo e um grupo de implantação (CLI)

1. Chame o comando create-application para criar um aplicativo no CodeDeploy chamado `CodeDeployGitHubDemo-App`:

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Chame o comando create-deployment-group para criar um grupo de implantação chamado `CodeDeployGitHubDemo-DepGrp`:

- Se você estiver implantando em uma instância do Amazon EC2*ec2-tag-key*, é a chave de tag da instância do Amazon EC2 que foi aplicada à sua instância do Amazon EC2 como parte do. Etapa 4: provisionar uma instância
- Se você estiver implantando em uma instância do Amazon EC2*ec2-tag-value*, é o valor da tag da instância do Amazon EC2 que foi aplicado à sua instância do Amazon EC2 como parte de. Etapa 4: provisionar uma instância
- Se você estiver implantando em uma instância local, *on-premises-tag-key* é a chave de tag da instância local que foi aplicada à sua instância local como parte de. Etapa 4: provisionar uma instância
- Se você estiver implantando em uma instância local, *on-premises-tag-value* é o valor da tag da instância local que foi aplicado à sua instância local como parte de. Etapa 4: provisionar uma instância
- *service-role-arn*é o ARN da função de serviço para a função de serviço que você criou em Criar uma função de serviço para. CodeDeploy (Siga as instruções em Obter o ARN do perfil de serviço (CLI) para localizar o ARN da função de serviço.)

```
aws deploy create-deployment-group --application-name CodeDeployGitHubDemo-App
 --ec2-tag-filters Key=ec2-tag-key,Type=KEY_AND_VALUE,Value=ec2-tag-value --on-
premises-tag-filters Key=on-premises-tag-key,Type=KEY_AND_VALUE,Value=on-premises-
tag-value --deployment-group-name CodeDeployGitHubDemo-DepGrp --service-role-
arn service-role-arn
```

> **Note**
>
> O comando create-deployment-group fornece suporte para a criação de acionadores que resultam no envio de notificações do Amazon SNS aos assinantes do tópico sobre eventos específicos em implantações e instâncias. O comando também oferece suporte a opções para reverter automaticamente as implantações e configurar alarmes para interromper as implantações quando os limites de monitoramento nos alarmes da CloudWatch Amazon forem atingidos. Os comandos para essas ações não estão incluídos neste tutorial.

Etapa 6: implantar o aplicativo na instância

Nesta etapa, você usa o CodeDeploy console ou o AWS CLI para implantar a revisão de amostra do seu GitHub repositório na sua instância.

Para implantar a revisão (console)

1. Na página Deployment group details (Detalhes do grupo de implantação), selecione Create deployment (Criar implantação).
2. Em Deployment group (Grupo de implantação), selecione **`CodeDeployGitHubDemo-DepGrp`**.
3. Em Tipo de revisão, escolha GitHub.
4. Em Connect to GitHub, faça o seguinte:

   - Para criar uma conexão entre CodeDeploy aplicativos e uma GitHub conta, saia GitHub em uma guia separada do navegador. Em GitHub Conta, insira um nome para identificar essa conexão e escolha Conectar GitHub a. A página da web solicita que você autorize CodeDeploy a interação com o aplicativo GitHub chamado. `CodeDeployGitHubDemo-App` Continue na etapa 5.
   - Para usar uma conexão que você já criou, na GitHubconta, selecione seu nome e escolha Conectar GitHub a. Continue na etapa 7.
   - Para criar uma conexão com uma GitHub conta diferente, saia GitHub em uma guia separada do navegador. Escolha Conectar a uma GitHub conta diferente e, em seguida, escolha Conectar GitHub a. Continue na etapa 5.
5. Siga as instruções na página de login para fazer login com sua GitHub conta.
6. Na página para Autorizar aplicativo, escolha Autorizar aplicativo.
7. Na página CodeDeploy Criar implantação, em Nome do repositório, insira o nome de GitHub usuário que você usou para entrar, seguido por uma barra (`/`), seguida pelo nome do repositório em que você enviou a revisão do aplicativo (por exemplo,). ***`my-github-user-name`*****`/`**
   `CodeDeployGitHubDemo`

   Caso não tenha certeza do valor que deve ser digitado ou se quiser especificar um repositório diferente:

   a. Em uma guia separada do navegador, acesse seu GitHub painel.
   b. Em Seus repositórios, passe o ponteiro do mouse sobre o nome do repositório de destino. Uma dica de ferramenta aparece, exibindo o nome GitHub do usuário ou da organização,

seguido por uma barra (/), seguida pelo nome do repositório. Digite esse valor em Repository name (Nome do repositório).

> **Note**
>
> Se o nome do repositório de destino não for exibido em Seus repositórios, use a GitHub caixa Pesquisar para encontrar o repositório de destino e o nome GitHub do usuário ou da organização.

8. Na caixa ID da confirmação, insira a ID da confirmação associada ao envio da revisão do seu aplicativo para GitHub.

   Caso não tenha certeza do valor que deve ser digitado:

   a. Em uma guia separada do navegador, acesse seu GitHub painel.
   b. Em Your repositories (Seus repositórios), selecione CodeDeployGitHubDemo.
   c. Na lista de confirmações, localize e copie a ID da confirmação associada ao envio da revisão do seu aplicativo para GitHub. Em geral, esse ID tem 40 caracteres de comprimento e é formado por letras e números. (Não use a versão mais curta do ID de confirmação, que é normalmente os 10 primeiros caracteres da versão mais longa.)
   d. Cole o ID de confirmação na caixa ID de confirmação.
9. Escolha Implantar e continue até a próxima etapa.

Para implantar a revisão (CLI)

Antes de chamar qualquer AWS CLI comando que interaja com GitHub (como o create-deployment comando, que você chamará em seguida), você deve dar CodeDeploy permissão para usar sua conta de GitHub usuário GitHub para interagir com o `CodeDeployGitHubDemo-App` aplicativo. Atualmente, você deve usar o console do CodeDeploy para fazer isso.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.

3. Selecione CodeDeployGitHubDemo-App.

4. Na guia Implantações, selecione Criar implantação.

> **Note**
>
> Você não criará uma nova implantação. Atualmente, essa é a única maneira de dar CodeDeploy permissão para interagir GitHub em nome de sua conta de GitHub usuário.

5. Em Grupo de implantação, escolha CodeDeployGitHubDemo-DepGrp.

6. Em Tipo de revisão, escolha GitHub.

7. Em Connect to GitHub, faça o seguinte:

   - Para criar uma conexão entre CodeDeploy aplicativos e uma GitHub conta, saia GitHub em uma guia separada do navegador. Em GitHub Conta, digite um nome para identificar essa conexão e escolha Conectar GitHub a. A página da web solicita que você autorize CodeDeploy a interação com GitHub o aplicativo chamado. `CodeDeployGitHubDemo-App` Continue na etapa 8.
   - Para usar uma conexão que você já criou, na GitHubconta, selecione seu nome e escolha Conectar GitHub a. Continue na etapa 10.
   - Para criar uma conexão com uma GitHub conta diferente, saia GitHub em uma guia separada do navegador. Escolha Conectar a uma GitHub conta diferente e, em seguida, escolha Conectar GitHub a. Continue na etapa 8.

8. Siga as instruções na página de login para fazer login com seu nome de GitHub usuário ou e-mail e senha.

9. Na página para Autorizar aplicativo, escolha Autorizar aplicativo.

10. Na página CodeDeploy Criar implantação, escolha Cancelar.

11. Chame o create-deployment comando para implantar a revisão do seu GitHub repositório na instância, onde:

    - *`repository`*é o nome GitHub da sua conta, seguido por uma barra (`/`), seguida pelo nome do seu repositório (`CodeDeployGitHubDemo`), por exemplo,. `MyGitHubUserName/CodeDeployGitHubDemo`

Caso não tenha certeza do valor que deve ser usado ou se quiser especificar um repositório diferente:

1. Em uma guia separada do navegador, acesse seu GitHub painel.
2. Em Seus repositórios, passe o ponteiro do mouse sobre o nome do repositório de destino. Uma dica de ferramenta aparece, exibindo o nome GitHub do usuário ou da organização, seguido por uma barra (`/`), seguida pelo nome do repositório. Este é o valor a ser usado.

> **Note**
>
> Se o nome do repositório de destino não aparecer em Seus repositórios, use a GitHub caixa Pesquisar para encontrar o repositório de destino e o nome de GitHub usuário ou organização correspondente.

- *`commit-id`*é o commit associado à versão da revisão do aplicativo que você enviou para o seu repositório (por exemplo,`f835159a...528eb76f`).

  Caso não tenha certeza do valor que deve ser usado:

  1. Em uma guia separada do navegador, acesse seu GitHub painel.
  2. Em Your repositories (Seus repositórios), selecione CodeDeployGitHubDemo.
  3. Na lista de confirmações, encontre a ID da confirmação associada ao envio da revisão do seu aplicativo para GitHub. Em geral, esse ID tem 40 caracteres de comprimento e é formado por letras e números. (Não use a versão mais curta do ID de confirmação, que é normalmente os 10 primeiros caracteres da versão mais longa.) Use esse valor.

Se você estiver trabalhando em uma máquina Linux, macOS ou Unix local:

```
aws deploy create-deployment \
  --application-name CodeDeployGitHubDemo-App \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --deployment-group-name CodeDeployGitHubDemo-DepGrp \
  --description "My GitHub deployment demo" \
  --github-location repository=repository,commitId=commit-id
```

Se estiver trabalhando em uma máquina Windows local:

```
aws deploy create-deployment --application-name CodeDeployGitHubDemo-App --
deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name
 CodeDeployGitHubDemo-DepGrp --description "My GitHub deployment demo" --github-
location repository=repository,commitId=commit-id
```

Etapa 7: monitorar e verificar a implantação

Nesta etapa, você usará o CodeDeploy console ou o AWS CLI para verificar o sucesso da implantação. Você usará seu navegador da Web para visualizar a página da Web que foi implantada na instância que você criou ou configurou.

> **Note**
>
> Se estiver implantando em uma instância do Ubuntu Server, use sua própria estratégia de teste para determinar se a revisão implantada funciona conforme esperado na instância e depois continue na próxima etapa.

Para monitorar e verificar a implantação (console)

1. No painel de navegação, expanda a opção Implantar e escolha Implantações.

2. Na lista de implantações, procure a linha com um valor de Aplicativo de CodeDeployGitHubDemo-Appe um valor de Grupo de Implantação de CodeDeployGitHubDemo-DepGrp. Se Bem-sucedido ou Falha não aparecer na coluna Status, clique no botão Atualizar periodicamente.
3. Se Falha aparecer na coluna Status, siga as instruções em Exibir detalhes da instância (console) para solucionar problemas com a implantação.
4. Se Bem-sucedido aparecer na coluna Status, você poderá verificar a implantação no seu navegador da Web. Nossa revisão de amostra implanta uma única página da Web na instância. Se estiver implantando em uma instância do Amazon EC2, no seu navegador da Web, acesse `http://public-dns` para a instância (por exemplo, `http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Se conseguir ver a página da Web, parabéns! Agora que você usou com sucesso AWS CodeDeploy para implantar uma revisão do GitHub, você pode pular paraEtapa 8: Limpeza.

Para monitorar e verificar a implantação (CLI)

1. Chame o comando list-deployments para obter o ID de implantação do aplicativo chamado `CodeDeployGitHubDemo-App` e o grupo de implantação chamado `CodeDeployGitHubDemo-DepGrp`:

   ```
   aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --
   deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output
    text
   ```

2. Chame o comando get-deployment, fornecendo o ID de implantação na saída do comando list-deployments:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
   [status, creator]" --output text
   ```

3. Se Falha for retornado, siga as instruções em Exibir detalhes da instância (console) para solucionar a implantação.
4. Se Bem-sucedido for retornado, você poderá tentar verificar a implantação no seu navegador da Web. Nossa revisão de amostra é uma única página da Web implantada na instância. Se você estiver implantando em uma instância do Amazon EC2, poderá visualizar essa página no seu navegador da Web acessando `http://public-dns` para a instância do Amazon EC2 (por exemplo, `http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Se conseguir ver a página da Web, parabéns! Você usou com sucesso AWS CodeDeploy para implantar a partir do seu GitHub repositório.

Etapa 8: Limpeza

Para evitar cobranças adicionais pelos recursos usados durante esse tutorial, você deve encerrar a instância do Amazon EC2 e seus recursos associados. Opcionalmente, você pode excluir os registros de componentes de implantação do CodeDeploy associados a esse tutorial. Se você estava usando um GitHub repositório apenas para este tutorial, você também pode excluí-lo agora.

Para excluir um CloudFormation pilha (se você usou o CloudFormation modelo para criar uma instância do Amazon EC2)

1. Faça login no Console de gerenciamento da AWS e abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.

2. Na coluna Stacks (Pilhas) escolha a pilha que começa com `CodeDeploySampleStack`.
3. Escolha Excluir.
4. Quando solicitado, escolha Excluir pilha. A instância do Amazon EC2 e o perfil de serviço e o perfil de instância do IAM associados são excluídos.

Para cancelar o registro manualmente e limpar uma instância local (caso você tenha provisionado uma instância local)

1. Use o AWS CLI para chamar o comando de cancelamento de registro na instância local representada aqui por *`your-instance-name`* e na região associada por: *`your-region`*

```
aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --
region your-region
```

2. Na instância on-premises, chame o comando desinstalar:

```
aws deploy uninstall
```

Para encerrar manualmente uma instância do Amazon EC2 (caso você tenha iniciado manualmente uma instância do Amazon EC2)

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação, em Instâncias, escolha Instâncias.
3. Marque a caixa ao lado da instância do Amazon EC2 que você deseja encerrar. No menu Ações, selecione Estado da instância e escolha Encerrar.
4. Quando solicitado, escolha Sim, Encerrar.

Para excluir os registros do componente de CodeDeploy implantação

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Selecione CodeDeployGitHubDemo-App.
4. Selecione Excluir aplicativo.
5. Quando solicitado, digite **`Delete`** e escolha Excluir.

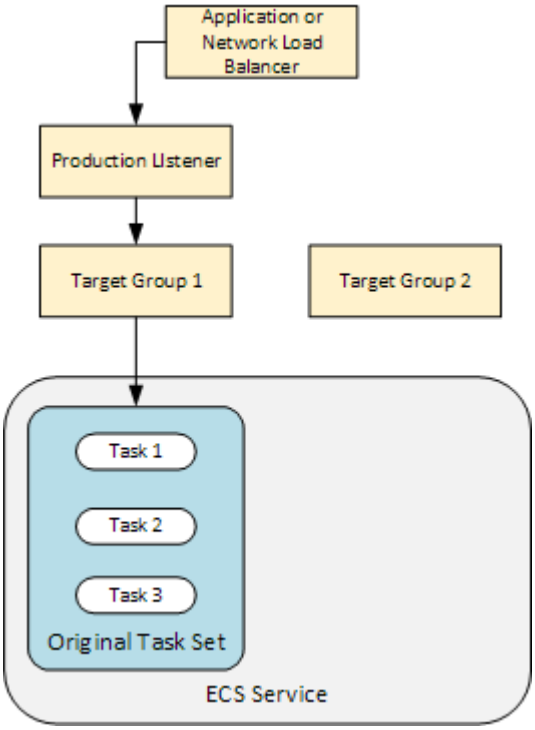
Para excluir seu GitHub repositório

Consulte Excluindo um repositório na GitHub ajuda.

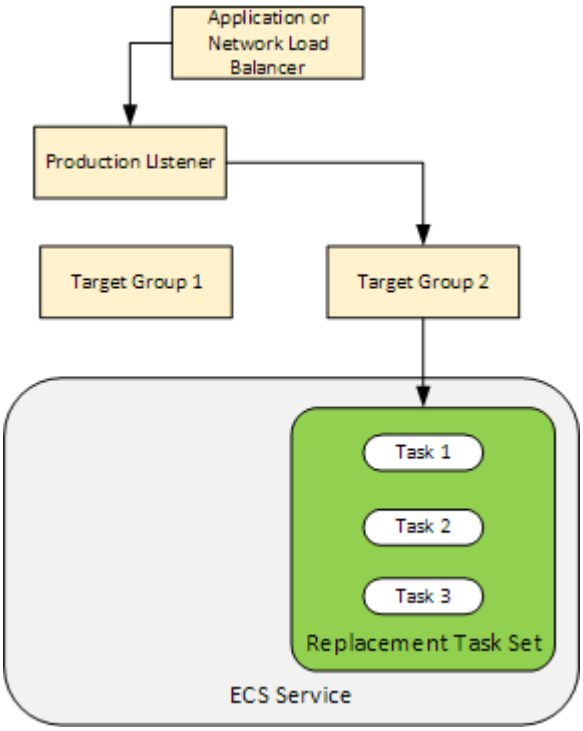
Tutorial: Implantar um aplicativo no Amazon ECS

Neste tutorial, você aprende como implantar um aplicativo no Amazon ECS usando CodeDeploy. Você começa com um aplicativo que já criou e implantou no Amazon ECS. A primeira etapa é atualizar o aplicativo modificando seu arquivo de definição de tarefa com uma nova tag. Em seguida, você usa CodeDeploy para implantar a atualização. Durante a implantação, CodeDeploy instala sua atualização em um novo conjunto de tarefas de substituição. Depois, ele muda o tráfego de produção da versão original do serviço do seu aplicativo do Amazon ECS, que está em seu conjunto de tarefas original, para a versão atualizada no conjunto de tarefas de substituição.

Durante uma implantação do Amazon ECS, CodeDeploy usa um balanceador de carga configurado com dois grupos-alvo e um ouvinte de tráfego de produção. O diagrama a seguir mostra como o balanceador de carga, o receptor de produção, os grupos de destino e o aplicativo do Amazon ECS são relacionados antes do início da implantação. Este tutorial usa um Application Load Balancer. Você também pode usar um Network Load Balancer.



Após uma implantação bem-sucedida, o listener do tráfego de produção atende ao tráfego do novo conjunto de tarefas de substituição e o conjunto de tarefas original é encerrado. O diagrama a seguir mostra como seus recursos estão relacionados após uma implantação bem-sucedida. Para obter mais informações, consulte O que acontece durante uma implantação do.



Para obter informações sobre como usar o AWS CLI para implantar um aplicativo no Amazon ECS, consulte Tutorial: Criando um serviço usando uma blue/green implantação. Para obter informações sobre como usar CodePipeline para detectar e implantar automaticamente alterações em um serviço do Amazon ECS com CodeDeploy, consulte Tutorial: Criar um pipeline com uma fonte e ECS-to-CodeDeploy implantação do Amazon ECR.

Depois de concluir este tutorial, você pode usar o CodeDeploy aplicativo e o grupo de implantação que você criou para adicionar um teste de validação de implantaçãoTutorial: Implantar um serviço do Amazon ECS com um teste de validação.

Tópicos

- Pré-requisitos
- Etapa 1: Atualizar o código do aplicativo do Amazon ECS
- Etapa 2: criar o AppSpec arquivo
- Etapa 3: usar o CodeDeploy console para implantar seu aplicativo
- Etapa 4: limpar

Pré-requisitos

Para concluir este tutorial, você deve primeiramente:

- Concluir as etapas 2 e 3 em Começando com CodeDeploy.
- Criar um Application Load Balancer configurado com dois grupos de destino e um receptor. Para obter informações sobre como criar um load balancer usando o console, consulte Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS. Para obter informações sobre como criar um balanceador de carga usando o AWS CLI, consulte Etapa 1: Criar um Application Load Balancer no Guia do usuário do Amazon Elastic Container Service. Ao criar o load balancer, anote o seguinte para este tutorial:
  - O nome do seu load balancer.
  - Os nomes dos seus grupos de destino.
  - A porta usada pelo listener do load balancer.
- Crie um cluster e serviço do Amazon ECS. Para obter mais informações, consulte as etapas 2, 3 e 4 no Tutorial: Criando um serviço usando uma blue/green implantação no Guia do usuário do Amazon Elastic Container Service. Anote o seguinte para este tutorial:
  - O nome do cluster do Amazon ECS.

- O ARN da definição de tarefa usada pelo seu serviço Amazon ECS.
- O nome do contêiner usado pelo seu serviço Amazon ECS.

- Crie um bucket do Amazon S3 para seu AppSpec arquivo.

Etapa 1: Atualizar o código do aplicativo do Amazon ECS

Nesta seção, você atualiza o aplicativo do Amazon ECS com uma nova revisão de sua definição de tarefa. A revisão atualizada adiciona um novo par de chave e tag. No Etapa 3: usar o CodeDeploy console para implantar seu aplicativo, você implanta a versão atualizada do aplicativo do Amazon ECS.

Para atualizar a definição da tarefa

1. Abra o console em https://console.aws.amazon.com/ecs/v2.
2. No painel de navegação, selecione Definições de tarefas.
3. Escolha a definição de tarefa usada pelo serviço do Amazon ECS.
4. Selecione a revisão da definição de tarefa e escolha Criar nova revisão, Criar nova revisão.
5. Para este tutorial, faça uma pequena atualização na definição de tarefa adicionando uma tag. Na parte inferior da página, em Tags, crie uma nova tag inserindo um novo par de chave e valor.
6. Escolha Criar.

   O número de revisão da definição da sua tarefa foi incrementado em um.
7. Selecione a guia JSON. Anote o seguinte, pois você precisará dessas informações na próxima etapa.

   - O valor de `taskDefinitionArn`. O formato é `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Esse é o ARN da sua definição de tarefa atualizada.
   - No elemento `containerDefinitions`, o valor de `name`. Esse é o nome do contêiner.
   - No elemento `portMappings`, o valor de `containerPort`. Essa é a porta do contêiner.

Etapa 2: criar o AppSpec arquivo

Nesta seção, você cria seu AppSpec arquivo e o carrega no bucket do Amazon S3 que você criou na Pré-requisitos seção. O AppSpec arquivo para uma implantação do Amazon ECS especifica

a definição da tarefa, o nome do contêiner e a porta do contêiner. Para obter mais informações, consulte AppSpec Exemplo de arquivo para uma implantação do Amazon ECS e AppSpec seção 'recursos' para implantações do Amazon ECS.

Para criar seu AppSpec arquivo

1. Se você quiser criar seu AppSpec arquivo usando YAML, crie um arquivo chamado`appspec.yml`. Se você quiser criar seu AppSpec arquivo usando JSON, crie um arquivo chamado`appspec.json`.
2. Escolha a guia apropriada, dependendo se você usa YAML ou JSON para seu AppSpec arquivo, e copie o conteúdo para o AppSpec arquivo que você acabou de criar. Para a propriedade `TaskDefinition`, use o ARN da definição de tarefa que você anotou na seção Etapa 1: Atualizar o código do aplicativo do Amazon ECS.

   JSON AppSpec

   ```
   {
     "version": 0.0,
     "Resources": [
       {
         "TargetService": {
           "Type": "AWS::ECS::Service",
           "Properties": {
             "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id:task-
   definition/ecs-demo-task-definition:revision-number",
             "LoadBalancerInfo": {
               "ContainerName": "your-container-name",
               "ContainerPort": your-container-port
             }
           }
         }
       }
     ]
   }
   ```

   YAML AppSpec

   ```
   version: 0.0
   Resources:
     - TargetService:
         Type: AWS::ECS::Service
   ```

```
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id:task-
definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "your-container-name"
          ContainerPort: your-container-port
```

> **Note**
>
> O conjunto de tarefas de substituição herda a sub-rede, o grupo de segurança, a versão da plataforma e os valores de IP públicos atribuídos de seu conjunto de tarefas original. Você pode substituir esses valores para seu conjunto de tarefas de substituição definindo suas propriedades opcionais em seu AppSpec arquivo. Para obter mais informações, consulte AppSpec seção 'recursos' para implantações do Amazon ECS e AppSpec Exemplo de arquivo para uma implantação do Amazon ECS.

3. Faça upload AppSpec do arquivo no bucket do S3 que você criou como pré-requisito para este tutorial.

Etapa 3: usar o CodeDeploy console para implantar seu aplicativo

Nesta seção, você cria um CodeDeploy aplicativo e um grupo de implantação para implantar seu aplicativo atualizado no Amazon ECS. Durante a implantação, CodeDeploy transfere o tráfego de produção do seu aplicativo para sua nova versão em um novo conjunto de tarefas de substituição. Para concluir esta etapa, você precisa dos seguintes itens:

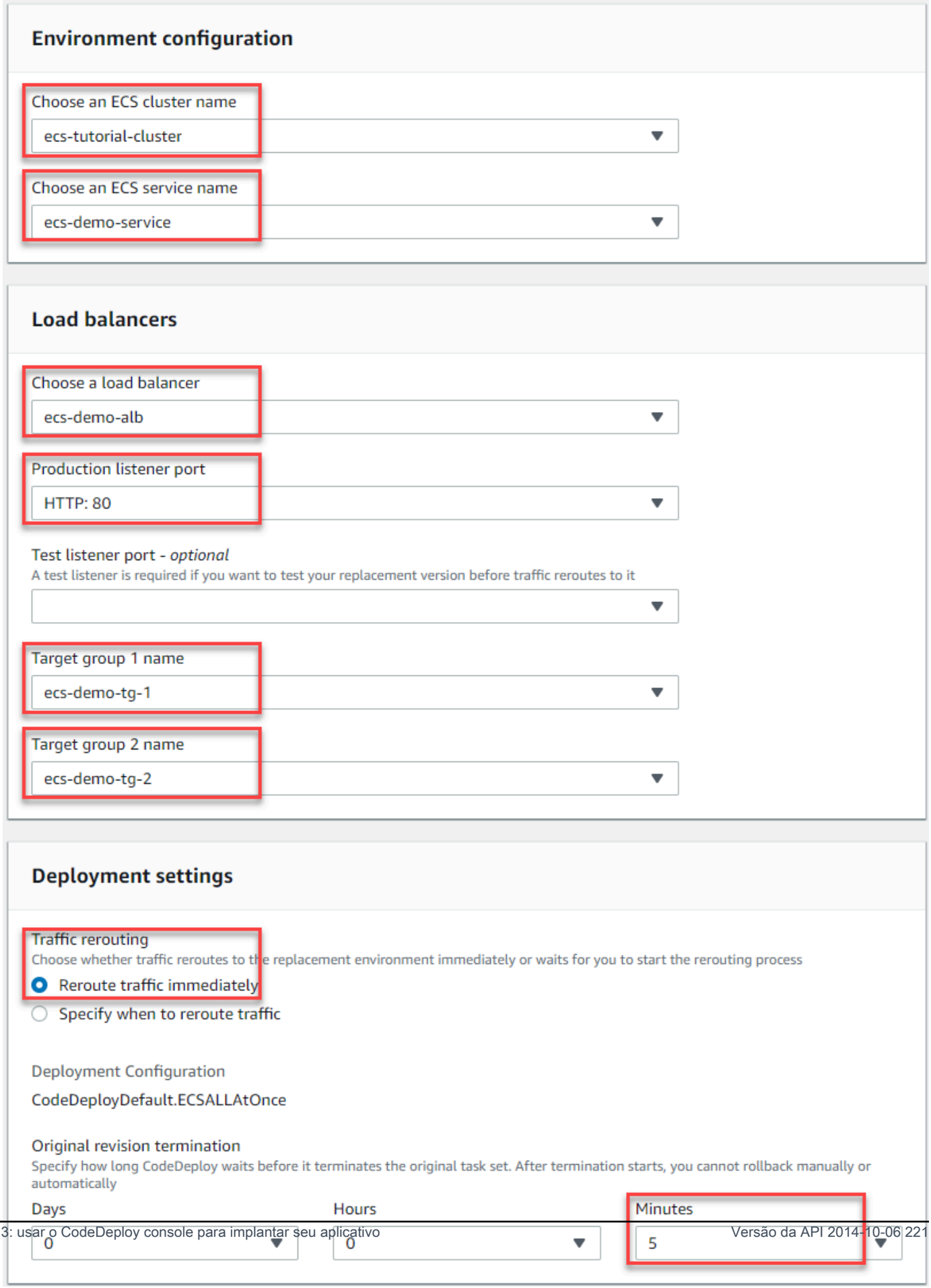
- O nome do cluster do Amazon ECS.
- O nome do serviço do Amazon ECS.
- O nome do Application Load Balancer.
- A porta do listener de produção.
- Os nomes dos grupos de destino.
- O nome do bucket do S3 que você criou.

Para criar um CodeDeploy aplicativo

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.
2. Escolha Criar aplicativo.
3. Em Nome do aplicativo, insira **`ecs-demo-codedeploy-app`**.
4. Em Compute platform (Plataforma de computação), selecione Amazon ECS.
5. Escolha Criar aplicativo.

Para criar um grupo CodeDeploy de implantação

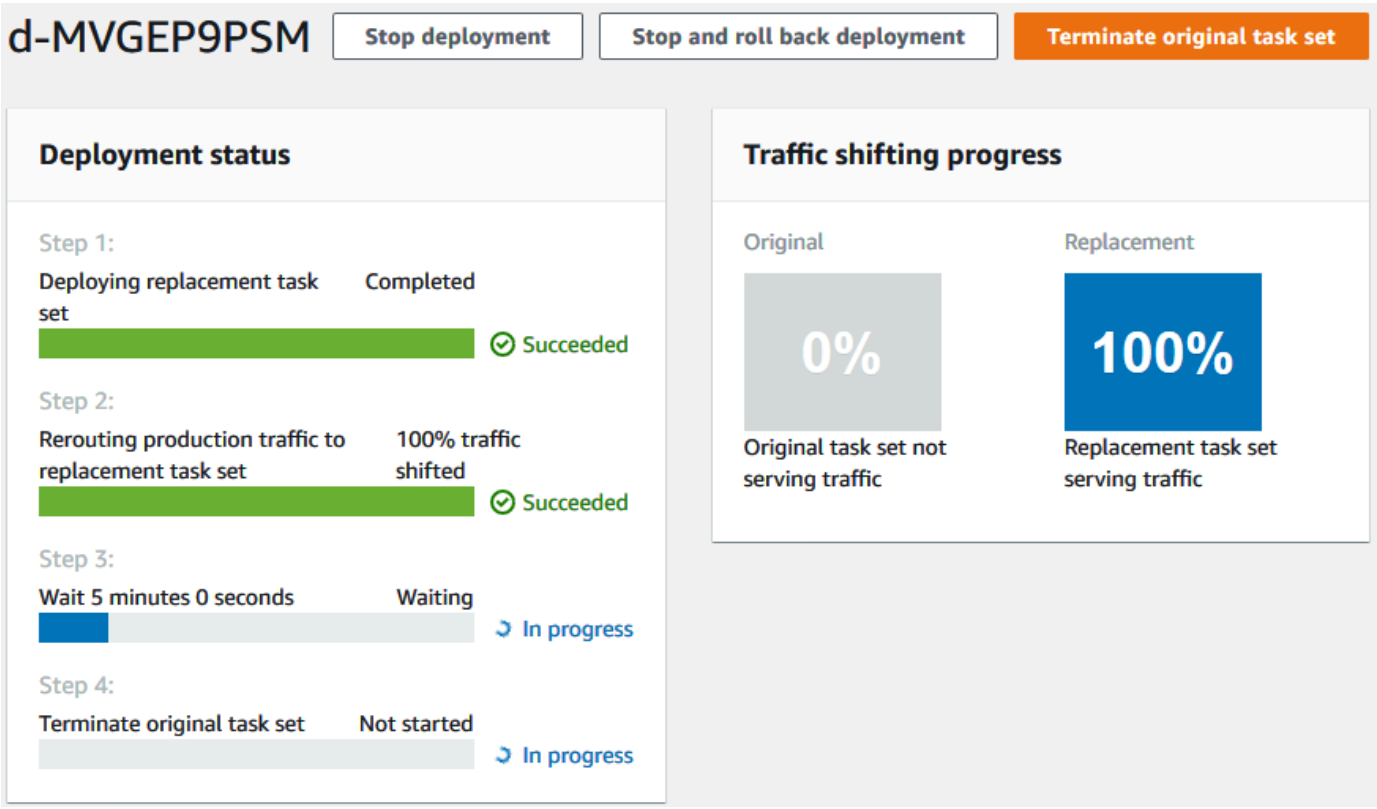
1. Na guia Deployment groups (Grupos de implantação) da página do aplicativo, escolha Create deployment group (Criar grupo de implantação).
2. Em Nome do grupo de implantação, insira **`ecs-demo-dg`**.
3. Em Função de serviço, escolha uma função de serviço que conceda CodeDeploy acesso ao Amazon ECS. Para obter mais informações, consulte Gerenciamento de identidade e acesso para AWS CodeDeploy.
4. Em Configuração de ambiente, selecione os nomes de cluster e serviço do Amazon ECS.
5. Em Balanceadores de carga, escolha o nome do balanceador de carga que distribui o tráfego para seu serviço do Amazon ECS.
6. Em Porta do ouvinte de produção, escolha a porta e o protocolo do ouvinte que fornece tráfego de produção para seu serviço Amazon ECS (por exemplo, HTTP: 80). Este tutorial não inclui um listener de teste opcional, portanto, não escolha uma porta em Test listener port (Porta do listener de teste).
7. Em Target group 1 name (Nome do grupo de destino 1) e Target group 2 name (Nome do grupo de destino 2), escolha dois grupos de destino diferentes para rotear o tráfego durante a implantação. Certifique-se de que esses são os grupos de destino criados para o load balancer. Não importa qual é usado para o grupo de destino 1 nem qual é usado para o grupo de destino 2.
8. Escolha Reroute traffic immediately (Redirecionar tráfego imediatamente).
9. Em Original revision termination (Encerramento da revisão original), escolha 0 dias, 0 horas e 5 minutos. Isso permite que você veja a implantação concluída mais rapidamente do que se você usar o padrão (1 hora).


Environment configuration
Choose an ECS cluster name
ecs-tutorial-cluster
Choose an ECS service name
ecs-demo-service
Load balancers
Choose a load balancer
ecs-demo-alb
Production listener port
HTTP: 80
Test listener port - optional
A test listener is required if you want to test your replacement version before traffic reroutes to it
Target group 1 name
ecs-demo-tg-1
Target group 2 name
ecs-demo-tg-2
Deployment settings
Traffic rerouting
Choose whether traffic reroutes to the replacement environment immediately or waits for you to start the rerouting process
Reroute traffic immediately
Specify when to reroute traffic
Deployment Configuration
CodeDeployDefault.ECSALLAtOnce
Original revision termination
Specify how long CodeDeploy waits before it terminates the original task set. After termination starts, you cannot rollback manually or automatically
Days
0
Hours
0
Minutes
5

10. Selecione Criar grupo de implantação.

Para implantar o aplicativo do Amazon ECS

1. Na página do console do grupo de implantação, escolha Criar implantação.
2. Em Grupo de implantação, escolha ecs-demo-dg.
3. Em Tipo de revisão, escolha Meu aplicativo está armazenado no Amazon S3. Em Revision location, (Local da revisão), insira o nome do bucket do S3.
4. Em Tipo de arquivo de revisão, escolha .json ou .yaml, conforme apropriado.
5. (Opcional) Em Descrição da implantação, digite uma descrição para a implantação.
6. Escolha Criar implantação.
7. Em Deployment status (Status da implantação), é possível monitorar a implantação. Depois que 100% do tráfego de produção for roteado para o conjunto de tarefas de substituição e antes do tempo de espera de cinco minutos expirar, você poderá escolher Encerrar conjunto de tarefas original para encerrar imediatamente o conjunto de tarefas original. Se você não escolher Terminate original task set (Encerrar conjunto de tarefas original), o conjunto de tarefas original será encerrado após o tempo de espera de cinco minutos especificado expirar.



Etapa 4: limpar

O próximo tutorial, Tutorial: Implantar um serviço do Amazon ECS com um teste de validação, se baseia neste tutorial e usa o aplicativo do CodeDeploy e o grupo de implantação que você criou. Se você quiser seguir as etapas desse tutorial, ignore esta etapa e não exclua os recursos que você criou.

> **Note**
>
> Sua AWS conta não incorre em cobranças pelos CodeDeploy recursos que você criou.

Os nomes dos recursos nessas etapas são os nomes sugeridos neste tutorial (por exemplo, **`ecs-demo-codedeploy-app`** para o nome do seu CodeDeploy aplicativo). Se você tiver usado nomes diferentes, certifique-se de usá-los durante a limpeza.

1. Use o comando delete-deployment-group para excluir o grupo de implantação. CodeDeploy

   ```
   aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --
   deployment-group-name ecs-demo-dg --region aws-region-id
   ```

2. Use o comando delete-application para excluir o aplicativo. CodeDeploy

   ```
   aws deploy delete-application --application-name ecs-demo-codedeploy-app --
   region aws-region-id
   ```

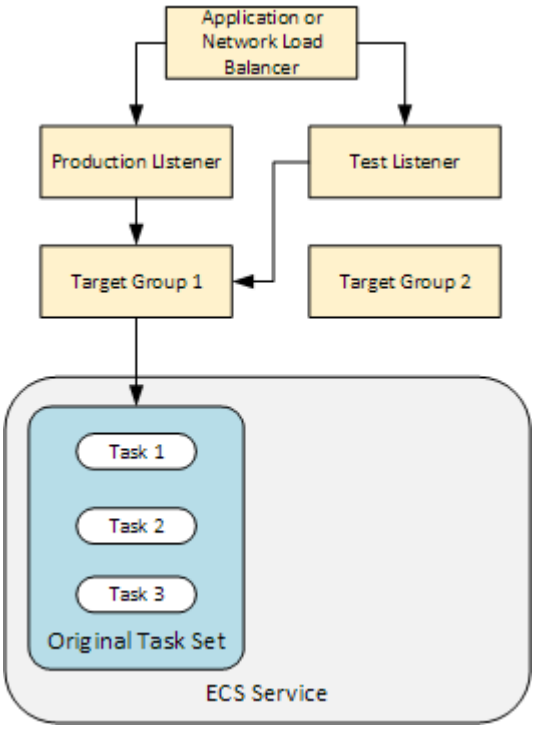
Tutorial: Implantar um serviço do Amazon ECS com um teste de validação

Neste tutorial, você aprenderá como usar uma função do Lambda para validar parte da implantação de um aplicativo do Amazon ECS atualizado. Este tutorial usa o CodeDeploy aplicativo, o grupo de CodeDeploy implantação e o aplicativo Amazon ECS em Tutorial: Implantar um aplicativo no Amazon ECS que você usou. Conclua esse tutorial antes de iniciar o seguinte.

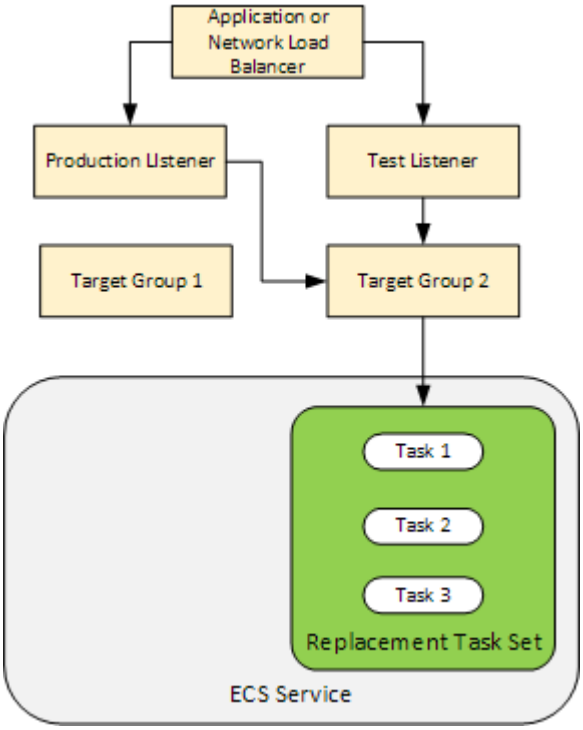
Para adicionar o teste de validação, primeiro implemente o teste em uma função do Lambda. Em seguida, em seu AppSpec arquivo de implantação, você especifica a função Lambda para o gancho de ciclo de vida que deseja testar. Se um teste de validação falhar, a implantação será interrompida,

revertida e marcada como com falha. Se o teste for bem-sucedido, a implantação continuará para o próximo evento do ciclo de vida da implantação ou gancho.

Durante uma implantação do Amazon ECS com testes de validação, CodeDeploy usa um balanceador de carga configurado com dois grupos-alvo: um ouvinte de tráfego de produção e um ouvinte de tráfego de teste. O diagrama a seguir mostra como o balanceador de carga, os receptores de produção e de teste, os grupos de destino e o aplicativo do Amazon ECS são relacionados antes do início da implantação. Este tutorial usa um Application Load Balancer. Você também pode usar um Network Load Balancer.



Durante uma implantação do Amazon ECS, há cinco ganchos do ciclo de vida para testes. Este tutorial implementa um teste durante o terceiro gancho de implantação de ciclo de vida, `AfterAllowTestTraffic`. Para obter mais informações, consulte Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS. Após uma implantação bem-sucedida, o listener do tráfego de produção atende ao tráfego do novo conjunto de tarefas de substituição e o conjunto de tarefas original é encerrado. O diagrama a seguir mostra como seus recursos estão relacionados após uma implantação bem-sucedida. Para obter mais informações, consulte O que acontece durante uma implantação do.



> **Note**
>
> A conclusão deste tutorial pode resultar em cobranças em sua AWS conta. Isso inclui possíveis cobranças de CodeDeploy AWS Lambda, CloudWatch e. Para obter mais informações, consulte AWS CodeDeploy preços, AWS Lambda preços e CloudWatch preços da Amazon.

Tópicos

- Pré-requisitos
- Etapa 1: Criar um receptor de teste
- Etapa 2: Atualizar o código do aplicativo do Amazon ECS
- Etapa 3: Criar uma função do Lambda de gancho do ciclo de vida
- Etapa 4: atualize seu AppSpec arquivo
- Etapa 5: Use o CodeDeploy console para implantar seu serviço Amazon ECS
- Etapa 6: Visualize a saída da função de gancho do Lambda em registros CloudWatch
- Etapa 7: limpar

Pré-requisitos

Para concluir este tutorial com êxito, você deve primeiro:

- Atenda os pré-requisitos em Pré-requisitos para Tutorial: Implantar um aplicativo no Amazon ECS.
- Siga as etapas em Tutorial: Implantar um aplicativo no Amazon ECS. Anote o seguinte:
  - O nome do seu load balancer.
  - Os nomes dos seus grupos de destino.
  - A porta usada pelo listener do load balancer.
  - O ARN do load balancer. Você o usará para criar um novo listener.
  - O ARN de um de seus grupos de destino. Você o usará para criar um novo listener.
  - O CodeDeploy aplicativo e o grupo de implantação que você cria.
  - O AppSpec arquivo que você cria e que é usado pela sua CodeDeploy implantação. Edite esse arquivo neste tutorial.

Etapa 1: Criar um receptor de teste

Uma implantação do Amazon ECS com testes de validação requer um segundo receptor. Esse receptor é usado para atender ao tráfego de teste para seu aplicativo do Amazon ECS atualizado em um conjunto de tarefas de substituição. Os testes de validação são executados no tráfego de teste.

O listener do tráfego de teste pode usar qualquer um de seus grupos de destino. Use o AWS CLI comando create-listener para criar um segundo ouvinte com uma regra padrão que encaminha o tráfego de teste para a porta 8080. Use o ARN do load balancer e o ARN de um de seus grupos de destino.

```
aws elbv2 create-listener --load-balancer-arn your-load-balancer-arn \
--protocol HTTP --port 8080 \
--default-actions Type=forward,TargetGroupArn=your-target-group-arn --region your-aws-region
```

Etapa 2: Atualizar o código do aplicativo do Amazon ECS

Nesta seção, você atualiza o aplicativo do Amazon ECS para usar uma nova revisão de sua definição de tarefa. Você cria a nova revisão e adiciona uma atualização secundária a ela adicionando uma tag.

Para atualizar a definição da tarefa

1. Abra o console clássico do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, selecione Definições de tarefas.
3. Marque a caixa de seleção da definição da tarefa usada pelo serviço do Amazon ECS.
4. Escolha Create new revisional (Criar nova revisão).
5. Faça uma pequena atualização na definição da tarefa adicionando uma tag. Na parte inferior da página, em Tags, crie uma nova tag inserindo um novo par de chave e valor.
6. Escolha Criar. Você deve ver que o número de revisão da definição da tarefa foi incrementado em um.
7. Selecione a guia JSON. Anote o valor de `taskDefinitionArn`. O formato é `arn:aws:ecs:`*`aws-region`*`:` *`account-id`*`:task-definition/`*`task-definition-family`*`:` *`task-definition-revision`*. Esse é o ARN da sua definição de tarefa atualizada.

Etapa 3: Criar uma função do Lambda de gancho do ciclo de vida

Nesta seção, você implementa uma função do Lambda para o hook do `AfterAllowTestTraffic` da implantação do Amazon ECS. A função do Lambda executa um teste de validação antes do aplicativo do Amazon ECS atualizado ser instalado. Para este tutorial, a função do Lambda retorna `Succeeded`. Durante uma implantação do mundo real, os testes de validação retornam `Succeeded` ou `Failed`, dependendo do resultado do teste de validação. Além disso, durante uma implantação do mundo real, você pode implementar uma função de teste do Lambda para um ou mais dos outros hooks de evento do ciclo de vida da implantação do Amazon ECS (`BeforeInstall`, `AfterInstall`, `BeforeAllowTraffic` e `AfterAllowTraffic`). Para obter mais informações, consulte Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS.

Um perfil do IAM é necessário para criar a função do Lambda. A função concede à função Lambda permissão para gravar em CloudWatch registros e definir o status de um gancho do ciclo de CodeDeploy vida.

Para criar um perfil do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções) e Create role (Criar função).
3. Crie uma função com as seguintes propriedades:

- Trusted entity (Entidade confiável): AWS Lambda.
- Permissões: AWSLambdaBasicExecutionRole. Isso concede à sua função Lambda permissão para CloudWatch gravar em registros.
- Nome da função: **`lambda-cli-hook-role`**.

Para obter mais informações, consulte Criar uma função AWS Lambda de execução.

4. Anexe a permissão `codedeploy:PutLifecycleEventHookExecutionStatus` à função que você criou. Isso concede às suas funções do Lambda permissão para definir o status de um gancho do CodeDeploy ciclo de vida durante uma implantação. Para obter mais informações, consulte Adicionar permissões de identidade do IAM no Guia AWS Identity and Access Management do usuário e PutLifecycleEventHookExecutionStatusna Referência da CodeDeploy API.

Como criar uma função do Lambda de hook do **`AfterAllowTestTraffic`**

1. Crie um arquivo denominado `AfterAllowTestTraffic.js` com os conteúdos a seguir.

```
'use strict';

 const AWS = require('aws-sdk');
 const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});

 exports.handler = (event, context, callback) => {

  console.log("Entering AfterAllowTestTraffic hook.");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event payload
  var deploymentId = event.DeploymentId;
  var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
  var validationTestResult = "Failed";

  // Perform AfterAllowTestTraffic validation tests here. Set the test result
  // to "Succeeded" for this tutorial.
  console.log("This is where AfterAllowTestTraffic validation tests happen.")
  validationTestResult = "Succeeded";

  // Complete the AfterAllowTestTraffic hook by sending CodeDeploy the validation
 status
```

```
  var params = {
   deploymentId: deploymentId,
   lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
   status: validationTestResult // status can be 'Succeeded' or 'Failed'
  };

  // Pass CodeDeploy the prepared validation test results.
  codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
   if (err) {
    // Validation failed.
    console.log('AfterAllowTestTraffic validation tests failed');
    console.log(err, err.stack);
    callback("CodeDeploy Status update failed");
   } else {
    // Validation succeeded.
    console.log("AfterAllowTestTraffic validation tests succeeded");
    callback(null, "AfterAllowTestTraffic validation tests succeeded");
   }
  });
 }
```

2. Crie um pacote de implantação do Lambda.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Use o comando `create-function` para criar uma função do Lambda para o hook do `AfterAllowTestTraffic`.

```
aws lambda create-function --function-name AfterAllowTestTraffic \
       --zip-file fileb://AfterAllowTestTraffic.zip \
       --handler AfterAllowTestTraffic.handler \
       --runtime nodejs10.x \
       --role arn:aws:iam::aws-account-id:role/lambda-cli-hook-role
```

4. Anote o ARN da função do Lambda na resposta do `create-function`. Você usa esse ARN ao atualizar o AppSpec arquivo de CodeDeploy implantação na próxima etapa.

Etapa 4: atualize seu AppSpec arquivo

Nesta seção, você atualiza seu AppSpec arquivo com uma `Hooks` seção. Na seção de `Hooks`, você especifica uma função do Lambda para o gancho do ciclo de vida do `AfterAllowTestTraffic`.

Para atualizar seu AppSpec arquivo

1. Abra o AppSpec arquivo que você criou no Etapa 2: criar o AppSpec arquivoTutorial: Implantar um aplicativo no Amazon ECS.
2. Atualize a propriedade `TaskDefinition` com o ARN da definição de tarefa anotado em Etapa 2: Atualizar o código do aplicativo do Amazon ECS.
3. Copie e cole a `Hooks` seção em seu AppSpec arquivo. Atualize o ARN depois do `AfterAllowTestTraffic` com o ARN da função do Lambda que você anotou no Etapa 3: Criar uma função do Lambda de gancho do ciclo de vida.

   JSON AppSpec

```
{
  "version": 0.0,
  "Resources": [
    {
      "TargetService": {
        "Type": "AWS::ECS::Service",
        "Properties": {
          "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id::task-
definition/ecs-demo-task-definition:revision-number",
          "LoadBalancerInfo": {
            "ContainerName": "sample-website",
            "ContainerPort": 80
          }
        }
      }
    }
  ],
  "Hooks": [
    {
      "AfterAllowTestTraffic": "arn:aws:lambda:aws-region-id:aws-account-
id:function:AfterAllowTestTraffic"
    }
  ]
}
```

YAML AppSpec

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id::task-
definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "sample-website"
          ContainerPort: 80
Hooks:
  - AfterAllowTestTraffic: "arn:aws:lambda:aws-region-id:aws-account-
id:function:AfterAllowTestTraffic"
```

4. Salve seu AppSpec arquivo e faça o upload para o bucket do S3.

Etapa 5: Use o CodeDeploy console para implantar seu serviço Amazon ECS

Nesta seção, você atualiza o grupo de implantação especificando a porta do listener de teste. Esse é o listener que você criou em Etapa 1: Criar um receptor de teste. Durante a implantação, o CodeDeploy executa o teste de validação durante o gancho de ciclo de vida da implantação `AfterAllowTestTraffic` usando o tráfego de teste fornecido a seu conjunto de tarefas de substituição usando o listener de teste. O teste de validação retorna o resultado `Succeeded`, portanto, a implantação prossegue com o próximo evento de ciclo de vida da implantação. Em um cenário do mundo real, sua função de teste retorna `Succeeded` ou `Failed`.

Para adicionar um listener de teste ao grupo de implantação

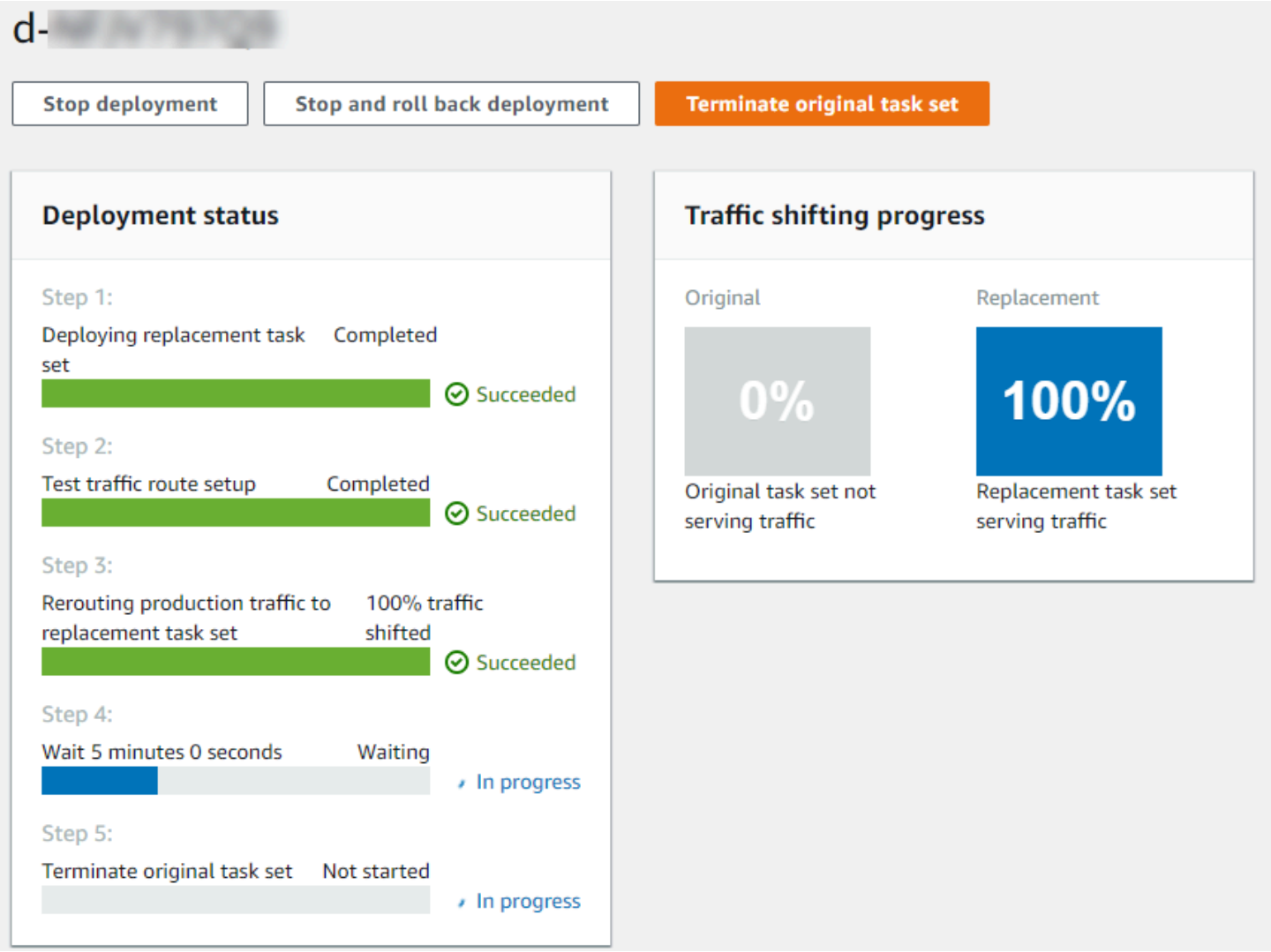
1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.
2. No painel de navegação, selecione Applications (Aplicativos).
3. Escolha o aplicativo que você criou em Tutorial: Implantar um aplicativo no Amazon ECS. Se você tiver usado o nome sugerido, ele será ecs-demo-codedeploy-app.

4. Em Deployment groups (Grupos de implantação), selecione o grupo de implantação que você criou em Tutorial: Implantar um aplicativo no Amazon ECS, Se você usou o nome sugerido, ele será ecs-demo-dg.
5. Escolha Editar.
6. Em Test listener port (Porta do listener de teste), escolha a porta e o protocolo para o listener de teste que você criou anteriormente neste tutorial. Ela deve ser HTTP: 8080.
7. Escolha Salvar alterações.

Para implantar o aplicativo do Amazon ECS

1. Na página do console do grupo de implantação, escolha Criar implantação.
2. Em Grupo de implantação, escolha ecs-demo-dg.
3. Em Tipo de revisão, escolha Meu aplicativo está armazenado no Amazon S3. Em Local da revisão, insira o nome do bucket e do AppSpec arquivo do S3 (por exemplo,**`s3://my-s3-bucket/appspec.json`**).
4. Em Tipo de arquivo de revisão, escolha .json ou .yaml conforme apropriado.
5. (Opcional) Em Descrição da implantação, digite uma descrição para a implantação.
6. Escolha Criar implantação.

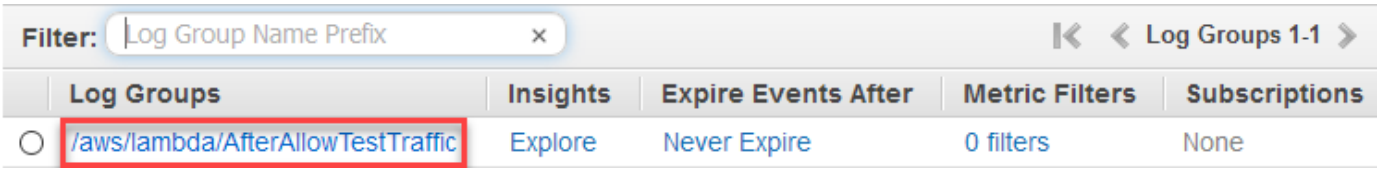
Você pode monitorar a implantação em Deployment status (Status da implantação). Depois que 100% do tráfego de produção for roteado para o conjunto de tarefas de substituição, você poderá escolher Encerrar conjunto de tarefas original para encerrar imediatamente o conjunto de tarefas original. Se você não escolher Terminate original task set (Encerrar conjunto de tarefas original), o conjunto de tarefas original será encerrado após o período especificado quando você criou o grupo de implantação.



Etapa 6: Visualize a saída da função de gancho do Lambda em registros CloudWatch

Se sua CodeDeploy implantação for bem-sucedida, os testes de validação em suas funções de gancho do Lambda também serão bem-sucedidos. Você pode confirmar isso examinando o registro da função de gancho em CloudWatch Logs.

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Na página de navegação, escolha Logs. Você deve ver um novo grupo de registros para a função de gancho do Lambda que você especificou em seu AppSpec arquivo.



3. Escolha o novo grupo de logs. Isso deve ser/aws/lambda/AfterAllowTestTrafficHook.
4. Escolha o fluxo de logs. Se você vir mais de um fluxo de logs, escolha aquele com a data e hora mais recentes em Last Event Time (Hora do último evento).
5. Expanda os eventos do fluxo de logs para confirmar se a função de hook do Lambda gravou mensagens de êxito no log. A tabela a seguir mostra que a função de hook do Lambda do `AfterAllowTraffic` foi bem-sucedida.



Etapa 7: limpar

Ao concluir este tutorial, limpe os recursos associados a ele para evitar cobranças por recursos que você não está usando. Os nomes dos recursos nesta etapa são os nomes sugeridos neste tutorial (por exemplo, **`ecs-demo-codedeploy-app`** para o nome do seu CodeDeploy aplicativo). Se você tiver usado nomes diferentes, certifique-se de usá-los em sua limpeza.

Para limpar os recursos do tutorial

1. Use o comando delete-deployment-group para excluir o grupo de implantação. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Use o comando delete-application para excluir o aplicativo. CodeDeploy

```
aws deploy delete-application --application-name ecs-demo-deployment-group --
region aws-region-id
```

3. Use o comando delete-function para excluir a função de hook do Lambda.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

4. Use o comando delete-log-group para excluir seu grupo de registros. CloudWatch

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutorial: Implemente uma função Lambda atualizada com CodeDeploy e o AWS Modelo de aplicativo sem servidores

AWS O SAM é uma estrutura de código aberto para criar aplicativos sem servidor. Ele transforma e expande a sintaxe YAML em um AWS SAM modelo em CloudFormation sintaxe para criar aplicativos sem servidor, como uma função Lambda. Para obter mais informações, consulte O que é o AWS Serverless Application Model?

Neste tutorial, você usa o AWS SAM para criar uma solução que faça o seguinte:

- Cria sua função do Lambda.
- Cria seu CodeDeploy aplicativo e seu grupo de implantação.
- Cria duas funções Lambda que executam testes de validação de implantação durante os ganchos do ciclo de CodeDeploy vida.
- Detecta quando a função do Lambda é atualizada. A atualização da função Lambda aciona uma implantação CodeDeploy que transfere incrementalmente o tráfego de produção da versão original da função Lambda para a versão atualizada.

> **Note**
>
> Este tutorial requer que você crie recursos que podem resultar em cobranças na sua conta da AWS . Isso inclui possíveis cobranças para CodeDeploy CloudWatch, Amazon AWS Lambda e. Para obter mais informações, consulte CodeDeploy preços, CloudWatch preços da Amazon e AWS Lambda preços.

Tópicos

- Pré-requisitos
- Etapa 1: Configurar sua infraestrutura
- Etapa 2: Atualizar a função do Lambda
- Etapa 3: Implantar a função do Lambda atualizada
- Etapa 4: Visualizar os resultados da implantação
- Etapa 5: limpar

Pré-requisitos

Para concluir este tutorial, você deve primeiramente:

- Siga as etapas em Começando com CodeDeploy.
- Instale a AWS Serverless Application Model CLI. Para obter informações, consulte Instalar a CLI do AWS SAM.
- Crie um bucket S3. AWS O SAM carrega os artefatos que são referenciados em seu modelo de AWS SAM nesse bucket.

Etapa 1: Configurar sua infraestrutura

Este tópico mostra como usar para criar arquivos AWS SAM para seu AWS SAM modelo e suas funções do Lambda. Em seguida, você usa os `deploy` comandos AWS SAM `package` e para gerar os componentes em sua infraestrutura. Quando sua infraestrutura estiver pronta, você terá um grupo de CodeDeploy aplicativos e implantação, a função Lambda para atualizar e implantar e duas funções Lambda que contêm testes de validação que são executados quando você implanta a função Lambda. Quando concluído, você pode usar CloudFormation para visualizar seus componentes no console Lambda ou AWS CLI para testar sua função Lambda.

Tópicos

- Criar seus arquivos
- Package the AWS Aplicação SAM
- Implemente o AWS Aplicação SAM
- (Opcional) Inspecionar e testar sua infraestrutura

Criar seus arquivos

Para criar sua infraestrutura, você deve criar os seguintes arquivos:

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Tópicos

- Crie seu AWS Modelo do SAM
- Criar um arquivo para sua função do Lambda
- Crie um arquivo para sua função BeforeAllowTraffic Lambda
- Crie um arquivo para sua função AfterAllowTraffic Lambda

Crie seu AWS Modelo do SAM

Crie um arquivo de modelo do AWS SAM que especifique os componentes em sua infraestrutura.

Para criar seu AWS Modelo do SAM

1. Crie um diretório denominado `SAM-Tutorial`.
2. No diretório `SAM-Tutorial`, crie um arquivo chamado `template.yml`.
3. Copie o código YAML a seguir no `template.yml`. Este é seu modelo do AWS SAM .

```
AWSTemplateFormatVersion : '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: A sample SAM template for deploying Lambda functions.

Resources:
# Details about the myDateTimeFunction Lambda function
  myDateTimeFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: myDateTimeFunction.handler
      Runtime: nodejs18.x
# Instructs your myDateTimeFunction is published to an alias named "live".
      AutoPublishAlias: live
```

```
# Grants this function permission to call lambda:InvokeFunction
      Policies:
        - Version: "2012-10-17"
          Statement:
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: '*'
      DeploymentPreference:
# Specifies the deployment configuration
          Type: Linear10PercentEvery1Minute
# Specifies Lambda functions for deployment lifecycle hooks
          Hooks:
            PreTraffic: !Ref beforeAllowTraffic
            PostTraffic: !Ref afterAllowTraffic

# Specifies the BeforeAllowTraffic lifecycle hook Lambda function
  beforeAllowTraffic:
    Type: AWS::Serverless::Function
    Properties:
      Handler: beforeAllowTraffic.handler
      Policies:
        - Version: "2012-10-17"
# Grants this function permission to call
 codedeploy:PutLifecycleEventHookExecutionStatus
          Statement:
          - Effect: "Allow"
            Action:
              - "codedeploy:PutLifecycleEventHookExecutionStatus"
            Resource:
              !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
        - Version: "2012-10-17"
# Grants this function permission to call lambda:InvokeFunction
          Statement:
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: !Ref myDateTimeFunction.Version
      Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
      FunctionName: 'CodeDeployHook_beforeAllowTraffic'
      DeploymentPreference:
        Enabled: false
```

```
      Timeout: 5
      Environment:
        Variables:
          NewVersion: !Ref myDateTimeFunction.Version

# Specifies the AfterAllowTraffic lifecycle hook Lambda function
  afterAllowTraffic:
    Type: AWS::Serverless::Function
    Properties:
      Handler: afterAllowTraffic.handler
      Policies:
        - Version: "2012-10-17"
          Statement:
# Grants this function permission to call
 codedeploy:PutLifecycleEventHookExecutionStatus
          - Effect: "Allow"
            Action:
              - "codedeploy:PutLifecycleEventHookExecutionStatus"
            Resource:
              !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
        - Version: "2012-10-17"
          Statement:
# Grants this function permission to call lambda:InvokeFunction
          - Effect: "Allow"
            Action:
              - "lambda:InvokeFunction"
            Resource: !Ref myDateTimeFunction.Version
      Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
      FunctionName: 'CodeDeployHook_afterAllowTraffic'
      DeploymentPreference:
        Enabled: false
      Timeout: 5
      Environment:
        Variables:
          NewVersion: !Ref myDateTimeFunction.Version
```

Esse modelo especifica o seguinte. Para obter mais informações, consulte Conceitos de modelo do AWS SAM.

Uma função do Lambda chamada **`myDateTimeFunction`**

Quando essa função do Lambda é publicada, a linha `AutoPublishAlias` no modelo a vincula a um alias chamado `live`. Posteriormente neste tutorial, uma atualização dessa função aciona uma implantação AWS CodeDeploy que transfere incrementalmente o tráfego de produção da versão original para a versão atualizada.

Duas funções de validação de implantação do Lambda

As seguintes funções do Lambda são executadas durante os ganchos do ciclo de CodeDeploy vida. As funções contêm código que validam a implantação da `myDateTimeFunction` atualizada. O resultado dos testes de validação é passado para o CodeDeploy usando seu método da API `PutLifecycleEventHookExecutionStatus`. Se houver falha em um teste de validação, haverá falha na implantação e ela será revertida.

- `CodeDeployHook_beforeAllowTraffic` é executado durante o gancho `BeforeAllowTraffic`.
- `CodeDeployHook_afterAllowTraffic` é executado durante o gancho `AfterAllowTraffic`.

O nome das duas funções começa com `CodeDeployHook_`. A função `CodeDeployRoleForLambda` permite chamadas para o método `invoke` do Lambda somente em funções do Lambda com nomes que começam com esse prefixo. Para obter mais informações, consulte AppSpec seção 'ganchos' para um AWS implantação do Lambda e PutLifecycleEventHookExecutionStatus na Referência da API do CodeDeploy .

Detecção automática de uma função do Lambda atualizada

O termo `AutoPublishAlias` informa à estrutura para detectar quando a função `myDateTimeFunction` muda e implanta-a usando o alias `live`.

Uma configuração de implantação

A configuração de implantação determina a taxa na qual seu CodeDeploy aplicativo transfere o tráfego da versão original da função Lambda para a nova versão. Esse modelo especifica a configuração de implantação predefinida `Linear10PercentEvery1Minute`.

> **Note**
>
> Você não pode especificar uma configuração de implantação personalizada em um modelo do AWS SAM. Para obter mais informações, consulte Create a Deployment Configuration.

Funções de gancho de ciclo de vida de implantação

A seção `Hooks` especifica as funções a serem executadas durante ganchos de evento de ciclo de vida. `PreTraffic` especifica a função que é executada durante o gancho `BeforeAllowTraffic`. `PostTraffic` especifica a função que é executada durante o gancho `AfterAllowTraffic`.

Permissões para o Lambda invocar outra função do Lambda

A `lambda:InvokeFunction` permissão especificada concede à função usada pelo aplicativo AWS SAM permissão para invocar uma função Lambda. Isso é necessário quando as funções `CodeDeployHook_beforeAllowTraffic` e `CodeDeployHook_afterAllowTraffic` invocam a função do Lambda implantada durante os testes de validação.

Criar um arquivo para sua função do Lambda

Crie o arquivo para a função que você atualiza e implanta posteriormente neste tutorial.

> **Note**
>
> Uma função do Lambda pode usar qualquer runtime compatível com o AWS Lambda. Para obter mais informações, consulte Tempos de execução do AWS Lambda.

Como criar sua função do Lambda

1. Crie um arquivo de texto e salve-o como `myDateTimeFunction.js` no diretório `SAM-Tutorial`.
2. Copie o Node.js código a seguir em`myDateTimeFunction.js`.

```
'use strict';
```

```
exports.handler = function(event, context, callback) {

  if (event.body) {
    event = JSON.parse(event.body);
  }

  var sc; // Status code
  var result = ""; // Response payload

  switch(event.option) {
    case "date":
      switch(event.period) {
        case "yesterday":
          result = setDateResult("yesterday");
          sc = 200;
          break;
        case "today":
          result = setDateResult();
          sc = 200;
          break;
        case "tomorrow":
          result = setDateResult("tomorrow");
          sc = 200;
          break;
        default:
          result = {
            "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
          };
          sc = 400;
          break;
      }
      break;

/*      Later in this tutorial, you update this function by uncommenting
        this section. The framework created by AWS SAM detects the update
        and triggers a deployment by CodeDeploy. The deployment shifts
        production traffic to the updated version of this function.

        case "time":
        var d = new Date();
        var h = d.getHours();
        var mi = d.getMinutes();
        var s = d.getSeconds();
```

```
            result = {
              "hour": h,
              "minute": mi,
              "second": s
            };
            sc = 200;
            break;
    */
          default:
            result = {
              "error": "Must specify 'date' or 'time'."
            };
            sc = 400;
          break;
      }

      const response = {
        statusCode: sc,
        headers: { "Content-type": "application/json" },
        body: JSON.stringify( result )
      };

      callback(null, response);

      function setDateResult(option) {

        var d = new Date(); // Today
        var mo; // Month
        var da; // Day
        var y; // Year

        switch(option) {
          case "yesterday":
            d.setDate(d.getDate() - 1);
            break;
          case "tomorrow":
            d.setDate(d.getDate() + 1);
          default:
           break;
        }

        mo = d.getMonth() + 1; // Months are zero offset (0-11)
        da = d.getDate();
        y = d.getFullYear();
```

```
        result = {
          "month": mo,
          "day": da,
          "year": y
        };

        return result;
      }
    };
```

A função do Lambda retorna o dia, mês e ano de ontem, hoje ou amanhã. Posteriormente neste tutorial, elimine o comentário do código que atualiza a função para retornar informações sobre o dia ou a hora que você especificar (por exemplo, o dia, mês e ano, ou a hora, o minuto e o segundo atuais). A estrutura criada por AWS SAM detecta e implanta a versão atualizada da função.

> **Note**
>
> Essa função Lambda também é usada em um AWS Cloud9 tutorial. AWS Cloud9 é um ambiente de desenvolvimento integrado baseado em nuvem. Para obter informações sobre como criar, executar, atualizar e depurar essa função em AWS Cloud9, consulte o AWS Lambda tutorial para AWS Cloud9.

Crie um arquivo para sua função BeforeAllowTraffic Lambda

Crie o arquivo para sua função do Lambda de hook `beforeAllowTraffic`.

1. Crie um arquivo de texto e salve-o como `beforeAllowTraffic.js` no diretório `SAM-Tutorial`.
2. Copie o Node.js código a seguir em`beforeAllowTraffic.js`. Essa função é executada durante o gancho `BeforeAllowTraffic` de sua implantação.

```
'use strict';

    const AWS = require('aws-sdk');
    const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
    var lambda = new AWS.Lambda();

    exports.handler = (event, context, callback) => {
```

```
    console.log("Entering PreTraffic Hook!");

    // Read the DeploymentId and LifecycleEventHookExecutionId from the event
payload
     var deploymentId = event.DeploymentId;
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    var functionToTest = process.env.NewVersion;
    console.log("BeforeAllowTraffic hook tests started");
    console.log("Testing new function version: " + functionToTest);

    // Create parameters to pass to the updated Lambda function that
    // include the newly added "time" option. If the function did not
    // update, then the "time" option is invalid and function returns
    // a statusCode of 400 indicating it failed.
    var lambdaParams = {
     FunctionName: functionToTest,
     Payload: "{\"option\": \"time\"}",
     InvocationType: "RequestResponse"
    };

    var lambdaResult = "Failed";
    // Invoke the updated Lambda function.
    lambda.invoke(lambdaParams, function(err, data) {
     if (err){ // an error occurred
      console.log(err, err.stack);
      lambdaResult = "Failed";
     }
     else{ // successful response
      var result = JSON.parse(data.Payload);
      console.log("Result: " +  JSON.stringify(result));
         console.log("statusCode: " + result.statusCode);

         // Check if the status code returned by the updated
         // function is 400. If it is, then it failed. If
         // is not, then it succeeded.
      if (result.statusCode != "400"){
           console.log("Validation succeeded");
       lambdaResult = "Succeeded";
         }
         else {
           console.log("Validation failed");
         }
```

```
      // Complete the PreTraffic Hook by sending CodeDeploy the validation status
      var params = {
       deploymentId: deploymentId,
       lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
       status: lambdaResult // status can be 'Succeeded' or 'Failed'
      };

      // Pass CodeDeploy the prepared validation test results.
      codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
       if (err) {
        // Validation failed.
        console.log("CodeDeploy Status update failed");
        console.log(err, err.stack);
        callback("CodeDeploy Status update failed");
       } else {
        // Validation succeeded.
        console.log("CodeDeploy status updated successfully");
        callback(null, "CodeDeploy status updated successfully");
       }
      });
     }
    });
   }
```

Crie um arquivo para sua função AfterAllowTraffic Lambda

Crie o arquivo para sua função do Lambda de hook `afterAllowTraffic`.

1. Crie um arquivo de texto e salve-o como `afterAllowTraffic.js` no diretório `SAM-Tutorial`.
2. Copie o Node.js código a seguir em`afterAllowTraffic.js`. Essa função é executada durante o gancho `AfterAllowTraffic` de sua implantação.

```
'use strict';

    const AWS = require('aws-sdk');
    const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
    var lambda = new AWS.Lambda();

    exports.handler = (event, context, callback) => {
```

```
    console.log("Entering PostTraffic Hook!");

    // Read the DeploymentId and LifecycleEventHookExecutionId from the event
payload
     var deploymentId = event.DeploymentId;
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    var functionToTest = process.env.NewVersion;
    console.log("AfterAllowTraffic hook tests started");
    console.log("Testing new function version: " + functionToTest);

    // Create parameters to pass to the updated Lambda function that
    // include the original "date" parameter. If the function did not
    // update as expected, then the "date" option might be invalid. If
    // the parameter is invalid, the function returns
    // a statusCode of 400 indicating it failed.
    var lambdaParams = {
     FunctionName: functionToTest,
     Payload: "{\"option\": \"date\", \"period\": \"today\"}",
     InvocationType: "RequestResponse"
    };

    var lambdaResult = "Failed";
    // Invoke the updated Lambda function.
    lambda.invoke(lambdaParams, function(err, data) {
     if (err){ // an error occurred
      console.log(err, err.stack);
      lambdaResult = "Failed";
     }
     else{ // successful response
      var result = JSON.parse(data.Payload);
      console.log("Result: " +  JSON.stringify(result));
         console.log("statusCode: " + result.statusCode);

         // Check if the status code returned by the updated
         // function is 400. If it is, then it failed. If
         // is not, then it succeeded.
      if (result.statusCode != "400"){
           console.log("Validation of time parameter succeeded");
       lambdaResult = "Succeeded";
         }
         else {
           console.log("Validation failed");
```

```
          }

       // Complete the PostTraffic Hook by sending CodeDeploy the validation status
       var params = {
        deploymentId: deploymentId,
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: lambdaResult // status can be 'Succeeded' or 'Failed'
       };

       // Pass CodeDeploy the prepared validation test results.
       codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
        if (err) {
         // Validation failed.
         console.log("CodeDeploy Status update failed");
         console.log(err, err.stack);
         callback("CodeDeploy Status update failed");
        } else {
         // Validation succeeded.
         console.log("CodeDeploy status updated successfully");
         callback(null, "CodeDeploy status updated successfully");
        }
       });
      }
     });
    }
```

Package the AWS Aplicação SAM

Agora você deve ter quatro arquivos em seu diretório `SAM-Tutorial`:

- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`
- `myDateTimeFunction.js`
- `template.yml`

Agora você está pronto para usar o sam package comando AWS SAM para criar e empacotar artefatos para suas funções CodeDeploy e aplicativos Lambda. Os artefatos são carregados em um bucket do S3. A saída do comando é um novo arquivo chamado `package.yml`. Esse arquivo é usado pelo sam deploy comando AWS SAM na próxima etapa.

> **Note**
>
> Para obter mais informações sobre o comando sam package, consulte a Referência do comando da CLI AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model .

No diretório `SAM-Tutorial`, execute o seguinte:

```
sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket amzn-s3-demo-bucket
```

Para o parâmetro `s3-bucket`, especifique o bucket do Amazon S3 que você criou como um pré-requisito para este tutorial. `output-template-file`Especifica o nome do novo arquivo usado pelo sam deploy comando AWS SAM.

Implemente o AWS Aplicação SAM

Use o sam deploy comando AWS SAM com o `package.yml` arquivo para criar suas funções Lambda e seu grupo de CodeDeploy aplicativos e implantação usando. CloudFormation

> **Note**
>
> Para obter mais informações sobre o comando sam deploy, consulte a Referência do comando da CLI AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model .

No diretório `SAM-Tutorial`, execute o comando a seguir.

```
sam deploy \
  --template-file package.yml \
  --stack-name my-date-time-app \
  --capabilities CAPABILITY_IAM
```

O parâmetro `--capabilities CAPABILITY_IAM` é necessário para autorizar o CloudFormation a criar perfis do IAM.

(Opcional) Inspecionar e testar sua infraestrutura

Este tópico mostra como visualizar seus componentes de infraestrutura e testar sua função do Lambda.

Para ver o resultado da sua pilha depois de executar **`sam deploy`**

1. Abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. No painel de navegação, escolha Pilhas. A pilha `my-date-time-app` é exibida na parte superior.
3. Escolha a guia Events (Eventos) para ver quais eventos foram concluídos. É possível visualizar os eventos enquanto a criação da pilha está em andamento. Quando a criação da pilha for concluída, você poderá ver todos os eventos de criação da pilha.
4. Com a pilha selecionada, escolha Resources (Recursos). Na coluna Tipo, é possível ver suas funções do Lambda, `myDateTimeFunction`, `CodeDeployHook_beforeAllowTraffic`, e `CodeDeployHook_afterAllowTraffic`. A coluna ID físico de cada uma de suas funções do Lambda contém um link para visualizar as funções no console do Lambda.

   > **Note**
   >
   > O nome da função `myDateTimeFunction` Lambda é prefixado com o nome da CloudFormation pilha e tem um identificador adicionado a ele, então parece. `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. Abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.
6. No painel de navegação, expanda a opção Deploy (Implantar) e escolha Applications (Aplicativos).
7. Você deve ver um novo CodeDeploy aplicativo criado por CloudFormation com um nome que começa com`my-date-time-app-ServerlessDeploymentApplication`. Escolha esse aplicativo.
8. Você deverá ver um grupo de implantação com um nome que começa com `my-date-time-app-myDateTimeFunctionDeploymentGroup`. Escolha esse grupo de implantação.

   Em Configuração de implantação, você deve ver CodeDeployDefault.LambdaLinear10PercentEvery1Minute.

(Opcional) Como testar sua função (console)

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. No painel de navegação, escolha sua função `my-date-time-app-myDateTimeFunction`. No console, o nome dela contém um identificador, portanto, ele se parece com `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Escolha Testar.
4. Em Event name (Nome do evento), insira um nome para o evento de teste.
5. Insira o seguinte para o evento de teste e escolha Create (Criar).

```
{
  "option": "date",
  "period": "today"
}
```

6. Escolha Testar. Você deverá ver apenas o evento de teste na lista de eventos de teste.

   Em Execution result (Resultado da execução), você deverá ver succeeded (bem-sucedido).
7. Em Execution result (Resultado da execução), expanda Details (Detalhes) para ver os resultados. Você deverá ver o mês, dia e ano atuais.

(Opcional) para testar sua função (AWS CLI)

1. Localize o ARN de sua função do Lambda. Ele será exibido na parte superior do console do Lambda quando você estiver visualizando a função.
2. Execute o comando a seguir. *`your-function-arn`*Substitua pela função ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload "{\"option\": \"date\", \"period\": \"today\"}" out.txt
```

3. Abra `out.txt` para confirmar que o resultado contém o mês, dia e ano atuais.

Etapa 2: Atualizar a função do Lambda

Neste tópico, você atualiza o arquivo `myDateTimeFunction.js`. Na próxima etapa, você usará o arquivo para implantar a função atualizada. Isso CodeDeploy aciona a implantação transferindo o tráfego de produção da versão atual da função Lambda para a versão atualizada.

Como atualizar sua função do Lambda

1. Abra o `myDateTimeFunction.js`.
2. Remova os dois marcadores de comentário ("`/*`" e "`*/`") e o texto explicativo no início e no final do `case` chamado `time` no bloco `switch`.

   O código sem comentário permite que você passe um novo parâmetro, `time`, para a função. Se você passar `time` para a função atualizada, ela retornará a `hour`, o `minute` e o `second` atuais.
3. Salvar `myDateTimeFunction.js`. Ele deve ter a seguinte aparência:

```
'use strict';

exports.handler = function(event, context, callback) {

  if (event.body) {
    event = JSON.parse(event.body);
  }

  var sc; // Status code
  var result = ""; // Response payload

  switch(event.option) {
    case "date":
      switch(event.period) {
        case "yesterday":
          result = setDateResult("yesterday");
          sc = 200;
          break;
        case "today":
          result = setDateResult();
          sc = 200;
          break;
        case "tomorrow":
          result = setDateResult("tomorrow");
          sc = 200;
```

```
          break;
        default:
          result = {
            "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
          };
          sc = 400;
          break;
      }
      break;
      case "time":
        var d = new Date();
        var h = d.getHours();
        var mi = d.getMinutes();
        var s = d.getSeconds();

        result = {
          "hour": h,
          "minute": mi,
          "second": s
        };
        sc = 200;
        break;

      default:
        result = {
          "error": "Must specify 'date' or 'time'."
        };
        sc = 400;
      break;
  }

  const response = {
    statusCode: sc,
    headers: { "Content-type": "application/json" },
    body: JSON.stringify( result )
  };

  callback(null, response);

  function setDateResult(option) {

    var d = new Date(); // Today
    var mo; // Month
    var da; // Day
```

```
    var y; // Year

    switch(option) {
      case "yesterday":
        d.setDate(d.getDate() - 1);
        break;
      case "tomorrow":
        d.setDate(d.getDate() + 1);
      default:
       break;
    }

    mo = d.getMonth() + 1; // Months are zero offset (0-11)
    da = d.getDate();
    y = d.getFullYear();

    result = {
      "month": mo,
      "day": da,
      "year": y
    };

    return result;
  }
};
```

Etapa 3: Implantar a função do Lambda atualizada

Nesta etapa, você usa a `myDateTimeFunction.js` atualizada para atualizar e iniciar a implantação da função do Lambda. Você pode monitorar o progresso da implantação no AWS Lambda console CodeDeploy ou.

A `AutoPublishAlias: live` linha em seu AWS SAM modelo faz com que sua infraestrutura detecte atualizações em funções que usam o `live` alias. Uma atualização em sua função aciona uma implantação CodeDeploy que transfere o tráfego de produção da versão original da função para a versão atualizada.

Os comandos sam package e sam deploy são usados para atualizar e acionar a implantação de sua função do Lambda. Você executou esses comandos em Package the AWS Aplicação SAM e Implemente o AWS Aplicação SAM.

Como implantar sua função do Lambda atualizada

1. No diretório `SAM-Tutorial`, execute o comando a seguir.

   ```
   sam package \
     --template-file template.yml \
     --output-template-file package.yml  \
     --s3-bucket amzn-s3-demo-bucket
   ```

   Isso cria um novo conjunto de artefatos que fazem referência à função do Lambda atualizada no bucket do S3.

2. No diretório `SAM-Tutorial`, execute o comando a seguir.

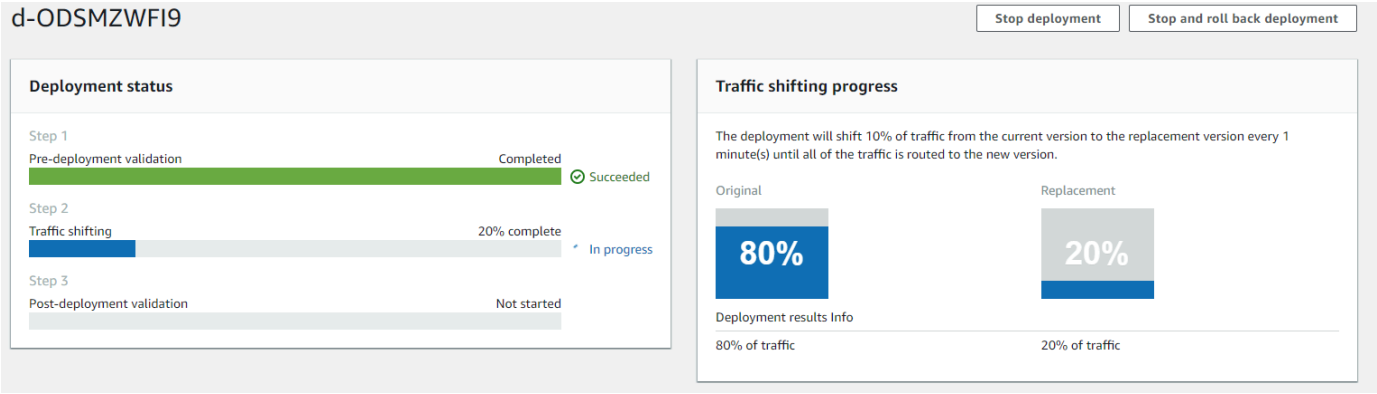
   ```
   sam deploy \
     --template-file package.yml \
     --stack-name my-date-time-app \
     --capabilities CAPABILITY_IAM
   ```

   Como o nome da pilha ainda é`my-date-time-app`, CloudFormation reconhece que essa é uma atualização da pilha. Para ver sua pilha atualizada, retorne ao CloudFormation console e, no painel de navegação, escolha Pilhas.

(Opcional) para visualizar o tráfego durante uma implantação (CodeDeploy console)

1. Abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.
2. No painel de navegação, expanda Aplicativos e escolha seu aplicativo my-date-time-app-. ServerlessDeploymentApplication
3. Em Deployment groups (Grupos de implantação), escolha o grupo de implantação do aplicativo. Seu status deve ser In progress (Em andamento).
4. Em Deployment group history (Histórico de grupos de implantação), escolha a implantação que está em andamento.

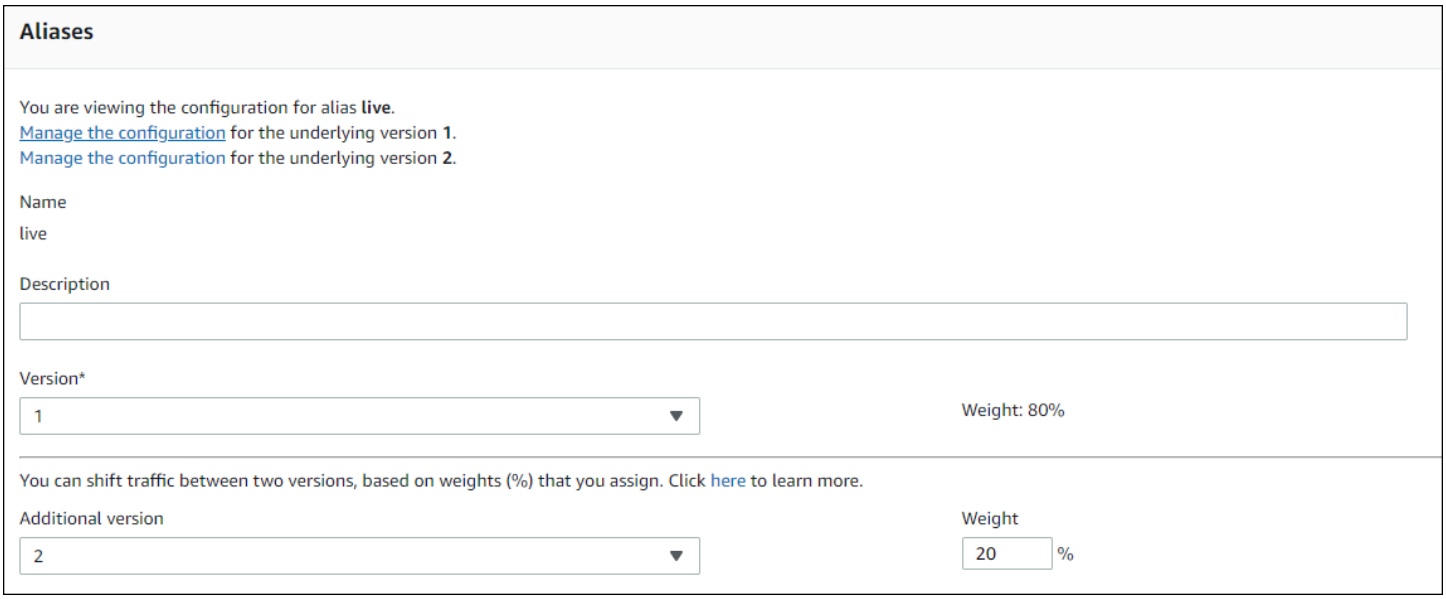
   A barra de progresso de Traffic shifting (Deslocamento de tráfego) e as porcentagens nas caixas Original e Replacement (Substituição) nesta página exibem o progresso.



(Opcional) Como visualizar o tráfego durante uma implantação (console do Lambda)

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. No painel de navegação, escolha sua função `my-date-time-app-myDateTimeFunction`. No console, o nome dela contém um identificador, portanto, ele se parece com `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Escolha Aliases e, em seguida, escolha ao vivo.

Os pesos ao lado da versão original da função (versão 1) e da versão atualizada da função (versão 2) mostram quanto tráfego é distribuído para cada versão no momento em que a página do console do AWS Lambda foi carregada. A página não atualiza os pesos ao longo do tempo. Se você atualizar a página uma vez por minuto, o peso da versão 1 diminuirá em 10% e o peso da versão 2 aumentará em 10% até que o peso da versão 2 seja 100.



Etapa 4: Visualizar os resultados da implantação

Nesta etapa, você visualiza os resultados da implantação. Se a implantação for bem-sucedida, você poderá confirmar que a função do Lambda atualizada recebe o tráfego de produção. Se sua implantação falhar, você poderá usar o CloudWatch Logs para visualizar a saída dos testes de validação na função Lambda que são executados durante os ganchos do ciclo de vida da implantação.

Tópicos

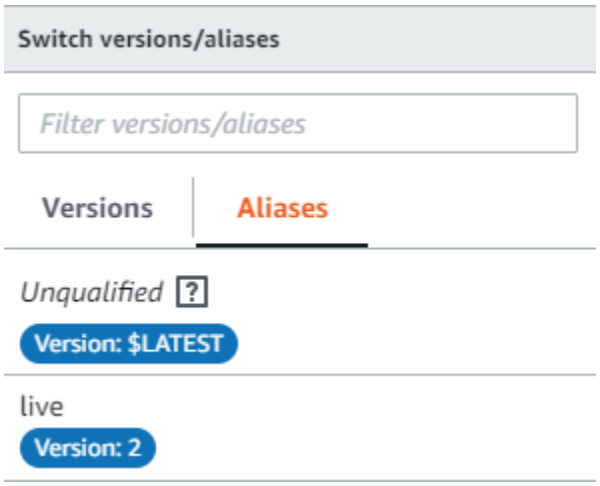
- Testar a função implantada
- Exibir eventos de gancho em CloudWatch Logs

Testar a função implantada

O comando sam deploy atualiza a função do Lambda do `my-date-time-app-myDateTimeFunction`. A versão da função é atualizada para 2 e adicionada ao alias `live`.

Como ver a atualização no console do Lambda

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. No painel de navegação, escolha a função `my-date-time-app-myDateTimeFunction`. No console, o nome dela contém um identificador, portanto, ele se parece com `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Escolha Qualifiers (Qualificadores) e Aliases. Depois que a implantação for concluída (aproximadamente 10 minutos), no alias `live`, você deverá ver Version: 2 (Versão: 2).



4. Em Function code (Código da função), visualize o código-fonte da função. Suas alterações deverão ser exibidas.
5. (Opcional) Você pode usar as instruções de teste em Etapa 2: Atualizar a função do Lambda para testar sua função atualizada. Crie um novo evento de teste com a carga a seguir e confirme se o resultado contém a hora, o minuto e o segundo atuais.

```
{
    "option": "time"
  }
```

Para usar o AWS CLI para testar sua função atualizada, execute o comando a seguir e abra `out.txt` para confirmar se o resultado contém a hora, o minuto e o segundo atuais.

```
aws lambda invoke --function your-function-arn --payload "{\"option\": \"time\"}"
 out.txt
```

> **Note**
>
> Se você usar o AWS CLI para testar sua função antes da conclusão da implantação, poderá receber resultados inesperados. Isso ocorre porque transfere CodeDeploy gradualmente 10% do tráfego para a versão atualizada a cada minuto. Durante a implantação, algum tráfego ainda aponta para a versão original, portanto, o `aws lambda invoke` pode usar a versão original. Após 10 minutos, a implantação é concluída e todo o tráfego aponta para a nova versão da função.

Exibir eventos de gancho em CloudWatch Logs

Durante o `BeforeAllowTraffic` hook, CodeDeploy executa sua função `CodeDeployHook_beforeAllowTraffic` Lambda. Durante o `AfterAllowTraffic` hook, CodeDeploy executa sua função `CodeDeployHook_afterAllowTraffic` Lambda. Cada função executa um teste de validação que invoca a versão atualizada de sua função usando o novo parâmetro `time`. Se a atualização da função do Lambda for bem-sucedida, a opção `time` não causará um erro e a validação será bem-sucedida. Se a função não tiver sido atualizada, o parâmetro não reconhecido causará um erro e haverá falha na validação. Esses testes de validação são apenas para fins de demonstração. Escreva seus próprios testes para validar sua implantação. Você pode usar o console de CloudWatch registros para ver seus testes de validação.

Para ver seus eventos de CodeDeploy gancho

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Na página de navegação, escolha Logs.
3. Na lista de grupos de registros, escolha/aws/lambda/CodeDeployHook_before AllowTraffic ou/aws/lambda/CodeDeployHookAllowTraffic_after.
4. Escolha o fluxo de logs. Você deverá ver somente um.
5. Expanda os eventos para ver seus detalhes.



Etapa 5: limpar

Para evitar cobranças adicionais pelos recursos que você usou durante este tutorial, exclua os recursos criados pelo seu AWS SAM modelo e CloudWatch os registros criados pelas suas funções de validação do Lambda.

Para excluir seu CloudFormation pilha

1. Faça login no Console de gerenciamento da AWS e abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.

2. Na coluna Stacks (Pilhas) escolha sua pilha `my-date-time-app` e escolha Delete (Excluir).
3. Quando solicitado, escolha Excluir pilha. As funções do Lambda, o grupo de CodeDeploy aplicativos e implantação e as funções do IAM criadas por AWS SAM são excluídas.

Para excluir seus registros em CloudWatch Registros

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Na página de navegação, escolha Logs.
3. Na lista de grupos de registros, escolha o botão ao lado de/aws/lambda/CodeDeployHook_before AllowTraffic.
4. Em Ações, escolha Excluir grupo de logs e Sim, Excluir.
5. Na lista de grupos de registros, escolha o botão ao lado de/aws/lambda/CodeDeployHook_after AllowTraffic.
6. Em Ações, escolha Excluir grupo de logs e Sim, Excluir.

Trabalhando com o CodeDeploy agente

O AWS CodeDeploy agente é um pacote de software que, quando instalado e configurado em uma instância, possibilita que essa instância seja usada em CodeDeploy implantações.

AWS suporta ativamente a versão secundária mais recente do CodeDeploy agente. Para obter a versão mais recente lançada, consulteHistórico de versões do CodeDeploy agente.

> **Note**
>
> O CodeDeploy agente é necessário somente se você implantar em uma plataforma EC2/On-Premises computacional. O agente não é necessário para implantações que usam a plataforma de computação do Amazon ECS ou do AWS Lambda.

Um arquivo de configuração é colocado na instância quando o agente é instalado. Este arquivo é usado para especificar como o agente funciona. Esse arquivo de configuração especifica caminhos de diretório e outras configurações AWS CodeDeploy a serem usadas ao interagir com a instância. É possível alterar algumas das opções de configuração no arquivo. Para obter informações sobre como trabalhar com o arquivo de configuração do CodeDeploy agente, consulteCodeDeploy referência de configuração do agente.

Para obter mais informações sobre como trabalhar com o CodeDeploy agente, como etapas para instalar, atualizar e verificar versões, consulteGerenciando as operações CodeDeploy do agente.

Tópicos

- Sistemas operacionais suportados pelo CodeDeploy agente
- Protocolo de comunicação e porta para o CodeDeploy agente
- Histórico de versões do CodeDeploy agente
- Gerenciando o CodeDeploy processo
- Limpeza de revisões de aplicativo e arquivos de log
- Arquivos instalados pelo CodeDeploy agente
- Gerenciando as operações CodeDeploy do agente

Sistemas operacionais suportados pelo CodeDeploy agente

Sistemas operacionais AMI do Amazon EC2 com suporte

O CodeDeploy agente foi testado nos seguintes sistemas operacionais AMI do Amazon EC2:

- Amazon Linux 2023 (x86_64, aarch64)
- Amazon Linux 2 (x86_64, aarch64)
- Red Hat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Servidor Ubuntu 16.04 LTS, 18,04 LTS, 20,04 LTS, 22,04 LTS, 24,04 LTS, 25,04, 26,04 LTS (x86_64, aarch64)
- Windows Server 2016, 2019, 2022, 2025 (x86_64)
- macOS 15, 26, 27 em instâncias EC2 Mac com Apple silicon (aarch64)

> **Note**
>
> As seguintes plataformas foram testadas somente na versão 2.0.x e posterior do CodeDeploy agente: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 e Ubuntu Server 24.04, 25.04, 26.04.
> O macOS é compatível somente com a versão 2.1.2 e posterior do CodeDeploy agente. O agente é compatível com macOS 15, 26, 27 com silicone Apple.

O CodeDeploy agente está disponível como código aberto para você se adaptar às suas necessidades. Ele pode ser usado com outros sistemas operacionais AMI do Amazon EC2. Para obter mais informações, acesse o repositório do CodeDeploy agente em GitHub.

Sistemas operacionais on-premises com suporte

O CodeDeploy agente foi testado nos seguintes sistemas operacionais locais:

- Windows Server 2016, 2019, 2022, 2025 (x86_64)
- Red Hat Enterprise Linux (RHEL) 8, 9, 10 (x86_64, aarch64)
- Oracle Linux 8, 9, 10 (x86_64, aarch64)
- Rocky Linux 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Servidor Ubuntu 16.04 LTS, 18,04 LTS, 20,04 LTS, 22,04 LTS, 24,04 LTS, 25,04, 26,04 LTS (x86_64, aarch64)
- macOS 15, 26, 27 com silicone Apple (aarch64)

> **Note**
>
> As seguintes plataformas foram testadas somente na versão 2.0.x e posterior do CodeDeploy agente: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13 e Ubuntu Server 24.04, 25.04, 26.04.
> O macOS é compatível somente com a versão 2.1.2 e posterior do CodeDeploy agente. O agente é compatível com macOS 15, 26, 27 com silicone Apple.

O CodeDeploy agente está disponível como código aberto para você se adaptar às suas necessidades. Ele pode ser usado com outros sistemas operacionais de instâncias locais. Para obter mais informações, acesse o repositório do CodeDeploy agente em GitHub.

Protocolo de comunicação e porta para o CodeDeploy agente

O CodeDeploy agente se comunica de saída usando HTTPS pela porta 443.

Quando o CodeDeploy agente é executado em uma instância do EC2, ele usa o endpoint de metadados do EC2 para recuperar informações relacionadas à instância. Saiba mais sobre como limitar e conceder acesso ao serviço de metadados da instância.

Histórico de versões do CodeDeploy agente

Suas instâncias devem estar executando uma versão compatível do CodeDeploy agente. A versão mínima atual suportada é 2.1.x.

> **Note**
>
> Recomendamos usar a versão mais recente do CodeDeploy agente. Se você estiver com problemas, atualize para a versão mais recente antes de entrar em contato com o AWS Support. Para obter informações sobre atualizações, consulte Atualize o CodeDeploy agente.

> **Status de implantação da versão 2.x**
>
> A versão 2.x do CodeDeploy agente está sendo lançada Regiões da AWS e talvez ainda não esteja disponível em todas as regiões. A versão 2.x é uma atualização opcional. Para atualizar para a versão 2.x a partir de qualquer revisão mais antiga, consulte. Instale o CodeDeploy agente Para descobrir qual versão você pode instalar, consulteDetermine a versão mais recente do agente disponível em um Região da AWS.

A tabela a seguir lista todas as versões do CodeDeploy agente e os recursos e aprimoramentos incluídos em cada versão.

Versão	Data de lançamento	Detalhes
2.1.2	13 de outubro de 2026	Adicionado: Support para macOS 15, 26, 27 com silicone Apple. Para instruções, consulte Instale o CodeDeploy agente para macOS. Adicionado: downloads em intervalos paralelos de revisões de aplicativos do Amazon S3, desativados por padrão. Para obter mais informações, consulte CodeDeploy referência de configuração do agente.

Versão	Data de lançamento	Detalhes
		Corrigido: a saída do script de eventos do ciclo de vida é gravada `codedeploy-agent-deployments.log` linha por linha, prefixada com o ID de implantação.
2.1.1	2 de outubro de 2026	Corrigido: o agente agora se conecta ao CodeDeploy endpoint correto na Nuvem Soberana AWS Europeia.
2.1.0	7 de setembro de 2026	Adicionado: suporte nativo para o modo de `RESTART` implantação, que reinicia o aplicativo nas instâncias de destino usando a revisão da última implantação bem-sucedida do grupo de implantação. O agente instala a partir da cópia da revisão que já está na instância. Se essa cópia estiver ausente ou incompleta, o agente baixa a revisão para que a implantação ainda seja concluída. Para o modo de implantação em si, consulte DeploymentMode na Referência da AWS CodeDeploy API. Alterado (segurança): o agente agora rejeita uma implantação cujo caminho de AppSpec arquivo é resolvido fora do diretório de revisão do aplicativo. Caminhos aninhados dentro da revisão, como `configs/appspec.yml` , ainda são suportados.
2.0.1	24 de agosto de 2026	Corrigido: no Windows, os caminhos de AppSpec arquivo que começam com uma barra invertida (\) ou com mais de um separador (por exemplo, `//my-folder`) agora são resolvidos dentro da revisão, não na raiz da unidade. Para obter exemplos de `source` valores compatíveis, consulteAppSpec seção 'arquivos' (somente EC2/On-Premises implantações).

Versão	Data de lançamento	Detalhes
2.0.0	15 de julho de 2026	Alterado: o CodeDeploy agente agora Rust-based é fornecido como um único binário nativo independente. A dependência do tempo de execução do Ruby foi eliminada. O Ruby não é mais necessário para instalar ou executar o agente. Por padrão, o agente é compatível com o comportamento da versão 1.8.x; os arquivos de configuração, o AppSpec manuseio, a semântica do ciclo de vida, o layout em disco e os resultados de implantação existentes permanecem inalterados, a menos que indicado abaixo. Adicionado: suporte expandido ao sistema operacional, incluindo Windows Server 2025, RHEL 10 e Ubuntu Server 26.04, bem como distribuições Linux adicionais e variantes ARM (aarch64). Para ver a lista completa das plataformas testadas com essa versão, consulteSistemas operacionais suportados pelo CodeDeploy agente. Adicionado: um `codedeploy-agent update` comando para atualização automática do agente sob demanda. Adicionado: uma porta de comando local. Essa é uma interface de gerenciamento TCP opcional que escuta somente o endereço de loopback (127.0.0.1) para consultar o status do agente e injetar comandos localmente. Ele está desativado por padrão e ativado com`enable_command_port` . Alterado: o comando de implantação local agora é`codedeploy-agent deploy-local` . O `codedeploy-local` comando permanece disponível para compatibilidade com versões anteriores. Todas as opções de comando são preservadas e ainda há suporte para caminhos locais, Amazon S3 (`s3://`) e fontes de GitHub pacotes.

Versão	Data de lançamento	Detalhes
		Alterado: No Linux, o agente é gerenciado por uma unidade systemd nativa (`codedeploy-agent.service`); o script SysV init.d não está mais instalado. Use `systemctl` para gerenciar o agente. Alterado (segurança): a verificação do certificado TLS está sempre ativada e os despejos principais são desativados por padrão. Adicionadas: configurações de proteção opcionais que você pode ativar para atender aos seus próprios requisitos de segurança. Cada configuração é desativada por padrão e você ativa as que deseja individualmente. Você pode rejeitar pacotes que contêm links simbólicos, percursos percorridos ou permissões inseguras; restringir o ambiente dos ganchos de eventos do ciclo de vida e restringir as permissões dos diretórios e arquivos de log do agente. Para ver a lista completa, consulte CodeDeploy referência de configuração do agente. Alterado: O script de instalação da linha de comando agora está escrito em Bash (anteriormente Ruby). Isso requer um `curl` ou `wget`. Alterado: o script de instalação da versão 2.0.x é publicado sob o `latestv2/` prefixo nos buckets regionais. `aws-codedeploy-`*`region-identifier`* O `latest/` prefixo continua servindo o script de instalação da versão 1.8.x. Alterado: AWS agora publica a versão 2.0.x como o pacote `AWSCodeDeployAgentV2` Systems Manager Distributor. O `AWSCodeDeployAgent` pacote continua servindo a versão 1.8.x. Removido: a definição `ssl_verify_peer` de configuração e as chaves de ProcessManager configuração do Ruby.

Versão	Data de lançamento	Detalhes
		Importante: atualmente não existe um caminho de atualização automática da versão 1.8.x para a 2.0.0. Para atualizar para a versão 2.0.0, instale-o com o pacote `AWSCodeDeployAgentV2` Systems Manager Distributor ou execute o `latestv2/install` script regional manualmente. Para instruções, consulte Instale o CodeDeploy agente.
1.8.1	3 de fevereiro de 2026	Corrigido: correção de bug do endpoint S3. Adicionado: certificado Sectigo CA à lista de CA do Windows.
1.8.0	31 de julho de 2025	Alterado: atualizou o Ruby incluído para 3.2 no CodeDeploy agente para Windows.
1.7.1	14 de novembro de 2024	Alteração: dependências foram atualizadas para patches de segurança.
1.7.0	6 de março de 2024	Adicionado: uma definição `:disable_imds_v1:` de configuração ao arquivo de configuração do CodeDeploy agente. Use essa configuração para desabilitar o fallback para o IMDSv1 quando ocorrerem erros no IMDSv2. O padrão é `false` (habilitar o fallback). Para obter mais informações, consulte a referência de configuração do CodeDeploy agente. Adição: suporte para o sistema operacional Red Hat Enterprise Linux 9 (RHEL 9). Adição: suporte para as versões 3.1 e 3.2 do Ruby no Ubuntu Server. Corrigido: o CodeDeploy agente agora gera um erro fácil de usar se o arquivo de configuração do CodeDeploy agente falhar ao carregar. Alterado: Atualizou o Ruby para 2.7.8-1 no agente para Windows. CodeDeploy

Versão	Data de lançamento	Detalhes
1.6.0	30 de março de 2023	Adição: suporte para Ruby 3.1, 3.2. Adição: suporte para o Amazon Linux 2023. Adição: suporte para o Windows Server 2022. Alteração: a configuração padrão de `verbose` agora é `false` para instâncias do Windows Server. Para continuar imprimindo mensagens de depuração em arquivos de log no Windows, você deve definir `verbose` como `true`. Remoção: suporte para Windows Server 2016 e Windows Server 2012 R2. Remoção: suporte para Amazon Linux 2018.03.x.

Versão	Data de lançamento	Detalhes
1.5.0	3 de março de 2023	Adição: suporte para Ruby 3. Adição: suporte para Ubuntu 22.04. Corrigido: um problema em que reiniciar o CodeDeploy agente logo após a inicialização fazia com que o agente parasse de funcionar. Alterado: o CodeDeploy agente agora falha na implantação do host na inicialização do agente se o serviço do agente for reiniciado inesperadamente durante a execução de um script de gancho. Essa correção permite que você evite esperar pelo período de tempo limite de 70 minutos antes de tentar novamente uma implantação. Aviso de descontinuação: o CodeDeploy agente 1.5.0 é a última versão compatível com o Windows Server 2016 e o Windows Server 2012 R2. Removido: Support para o CodeDeploy agente no Ubuntu 14.04 LTS, Windows Server 2008 R2 e Windows Server 2008 R2 de 32 bits.
1.4.1	6 de dezembro de 2022	Correção: vulnerabilidade de segurança relacionada ao registro em log. Melhoria: registro em log aprimorado ao pesquisar o comando host.

Versão	Data de lançamento	Detalhes
1.4.0	31 de agosto de 2022	Adição: suporte para Red Hat Enterprise Linux 8. Adicionado: Support para caminhos de arquivo longos no CodeDeploy agente para Windows. Para habilitar caminhos de arquivo longos, você precisará definir a chave de registro apropriada do Windows e reiniciar seu agente. Para obter mais informações, consulte Caminhos de arquivo longos causam erros do tipo “Nenhum arquivo ou diretório”. Correção: um problema com a operação de descompactação quando o disco estava cheio. O CodeDeploy agente agora detecta o código de saída 50 do unzip, indicando um disco cheio, remove arquivos parcialmente extraídos e gera uma exceção para publicar uma falha no servidor. CodeDeploy A mensagem de erro é visível como uma mensagem de erro de evento do ciclo de vida e a implantação no nível do host será interrompida sem travamentos nem limite de tempo. Correção: um problema que faria com que o agente falhasse. Correção: um problema em que os hooks expiravam durante uma condição de corrida extrema. Os hooks sem scripts agora continuarão e não causarão mais falhas ou tempos limite. Alterado: o `update` script do `bin` diretório do CodeDeploy agente foi removido porque não é mais usado. Alterado: o CodeDeploy agente para Windows Server agora inclui o Ruby 2.7. Alterado: novas variáveis de ambiente foram adicionadas para serem usadas por scripts de gancho, dependendo da origem do pacote de implantação (Amazon S3 GitHub ou).

Versão	Data de lançamento	Detalhes
		Para obter mais informações, consulte Disponibilidade variáveis de ambientes para hooks. **Important** Aviso de descontinuação: o CodeDeploy agente 1.4.0 é a última versão que incluirá instaladores para o Windows Server de 32 bits. Aviso de descontinuação: o CodeDeploy agente 1.4.0 é a última versão que oferecerá suporte ao Windows Server 2008 R2. Removido: Support para o CodeDeploy agente nas seguintes AMIs do Amazon EC2: Amazon Linux 2014.09, 2016.03, 2016.09 e 2017.03.

Versão	Data de lançamento	Detalhes
1.3.2	6 de maio de 2021	**Important** CodeDeploy endereços do agente 1.3.2 CVE-2018-1000201que afetam os hosts Windows que executam o agente. O CVE cita ruby-ffi, que é uma dependência do agente. CodeDeploy Se seu agente foi instalado com o Amazon EC2 Systems Manager (SSM) e está configurado para ser atualizado automaticamente, nenhuma ação é necessária. Caso contrário, é necessária uma ação para atualizar manualmente o agente. Para atualizar o agente, siga as instruções em Atualizar o CodeDeploy agente no Windows Server. Corrigido: um problema ao instalar o CodeDeploy agente no Ubuntu 20.04 e versões posteriores. Correção: um problema intermitente que ocorria ao extrair arquivos compactados porque os caminhos relativos não estavam sendo tratados corretamente. Adição: suporte para AWS PrivateLink e endpoints da VPC para instâncias do Windows. Adicionado: melhorias no AppSpec arquivo, conforme descrito abaixo. • Agora você pode especificar um nome de arquivo personalizado para o AppSpec arquivo ao criar uma implantação local. Para obter mais informações, consulte Criar uma implantação local. • O AppSpec arquivo agora pode ter uma extensão de `.yaml` arquivo.

Versão	Data de lançamento	Detalhes
		• Agora você pode sobrescrever os arquivos implantados usando uma nova `file_exists_behavior` configuração opcional no AppSpec arquivo. Para obter mais informações, consulte AppSpec seção 'arquivos' (somente EC2/On-Premises implantações). Atualizado: CodeDeploy agora usa o AWS SDK for Ruby 3.0.
1.3.1	22 de dezembro de 2020	Correção: problema da versão 1.3.0 que impedia o início de instâncias on-premises.
1.3.0	10 de novembro de 2020	**Important** Essa versão está obsoleta. Correção: foi removido um certificado expirado que não era mais usado. Corrigido: foi removida a mensagem de aviso do script de desinstalação do agente usado por AWS Systems Manager, facilitando o downgrade de um host ou frota para uma versão anterior do agente.

Versão	Data de lançamento	Detalhes
1.2.1	23 de setembro de 2020	Alteração: dependência do AWS SDK para Ruby atualizada de v2 para v3. Adição: suporte para IMDSv2. Inclui um fallback silencioso para o IMDSv1 caso as solicitações http do IMDSv2 falhem. Alteração: dependências atualizadas do Rake e do Rubyzip para patches de segurança. Correção: certifique-se de que um arquivo PID vazio retorne um status `No CodeDeploy Agent Running` e limpe o arquivo PID na inicialização do agente.

Versão	Data de lançamento	Detalhes
1.1.2	4 de agosto de 2020	Adição: suporte para Ubuntu Server 19.10 e 20.04. Nota: A versão 19.10 atingiu sua data de fim de vida útil e não é mais suportada pelo Ubuntu ou. CodeDeploy Adição: melhorias na eficiência de memória para Linux e Ubuntu para liberar a memória reservada mais rapidamente. Adição: compatibilidade com a “limpeza silenciosa” do Windows Server, que estava fazendo com que o agente parasse de responder em alguns casos. Adição: diretórios não vazios são ignorados durante a limpeza para evitar falhas na implantação. Adicionado: Support for AWS Local Zone em Los Angeles (LA). Adicionado: extraia AZ dos metadados da instância para fornecer compatibilidade com Zonas AWS Locais. Adição: os usuários agora podem fornecer seu arquivo em subdiretórios e não precisam armazená-lo no diretório raiz. Adição: detecção de um problema com o Rubyzip que poderia resultar em vazamentos de memória. Atualização do comando de descompactação para tentar primeiro usar um utilitário de descompactação instalado no sistema antes de usar o Rubyzip. Adição: `:enable_auth_policy:` como uma definição de configuração do agente. Alteração: os avisos de descompactação agora são ignorados para que as implantações continuem.

Versão	Data de lançamento	Detalhes
1.1.0	30 de junho de 2020	Alterado: o controle de versão do CodeDeploy agente agora segue a convenção de controle de versão padrão do Ruby. Adição: novo parâmetro para o comando de instalação e atualização a fim de permitir a instalação de uma versão específica do agente pela linha de comando. Removido: removido o CodeDeploy agente Auto Updater para Linux e Ubuntu. Para configurar as atualizações automáticas do CodeDeploy agente, consulte Instalar o CodeDeploy agente usando AWS Systems Manager.
1.0.1.1597	15 de novembro de 2018	Aprimoramento: CodeDeploy suporta o Ubuntu 18.04. Aprimoramento: CodeDeploy suporta Ruby 2.5. Aprimoramento: CodeDeploy suporta endpoints FIPS. Para obter mais informações sobre os endpoints FIPS, consulte Visão geral da publicação 140-2 da FIPS. Para endpoints que podem ser usados com CodeBuild, consulte CodeDeployRegiões e endpoints.
1.0.1.1518	12 de junho de 2018	Aprimoramento: corrigiu um problema que causava um erro quando o CodeDeploy agente era fechado enquanto aceitava solicitações de enquete. Aprimoramento: foi adicionado um recurso de rastreamento de implantação que impede que o CodeDeploy agente seja fechado quando uma implantação está em andamento. Melhoria: melhoria no desempenho ao excluir arquivos.

Versão	Data de lançamento	Detalhes
1.0.1.1458	6 de março de 2018	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Melhoria: validações de certificado aprimoradas para oferecer suporte a mais autoridades confiáveis. Aprimoramento: corrigiu um problema que fazia com que a CLI local falhasse durante uma implantação que BeforeInstall incluía um evento de ciclo de vida. Aprimoramento: corrigiu um problema que poderia fazer com que uma implantação ativa falhasse quando o CodeDeploy agente fosse atualizado.
1.0.1.1352	16 de novembro de 2017	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Funcionalidade: introduziu um novo recurso para testar e depurar uma EC2/On-Premises implantação em uma máquina ou instância local em que o CodeDeploy agente está instalado.
1.0.1.1106	16 de maio de 2017	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Recurso: introdução de um novo suporte para manipular o conteúdo em um local de destino que não faz parte da revisão de aplicativo da implantação bem-sucedida mais recente. As opções de implantação de conteúdo existente agora incluem manter o conteúdo, substituir o conteúdo ou marcar a implantação como falha. Aprimoramento: tornou o CodeDeploy agente compatível com a versão 2.9.2 do AWS SDK para Ruby (aws-sdk-core 2.9.2).

Versão	Data de lançamento	Detalhes
1.0.1.1095	29 de março de 2017	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Aprimoramento: foi introduzido o suporte para o CodeDeploy agente na região da China (Pequim). Melhoria: o Puppet pode ser executado em instâncias do Windows Server quando invocado por um hook de evento de ciclo de vida. Melhoria: a manipulação de operações `untar` foi aprimorada.
1.0.1.1067	6 de janeiro de 2017	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Melhoria: revisão de muitas mensagens de erro para incluir causas mais específicas de falhas de implantação. Aprimoramento: corrigiu um problema que impedia o CodeDeploy agente de identificar a revisão correta do aplicativo a ser implantada durante algumas implantações. Melhoria: reversão do uso de `pushd` e `popd` antes e depois da operação `untar`.
1.0.1.1045	21 de novembro de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Aprimoramento: tornou o CodeDeploy agente compatível com a versão 2.6.11 do AWS SDK para Ruby (aws-sdk-core 2.6.11).

Versão	Data de lançamento	Detalhes
1.0.1.1037	19 de outubro de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. O CodeDeploy agente para instâncias Amazon Linux, RHEL e Ubuntu Server foi atualizado com a seguinte alteração. Para instâncias do Windows Server, a versão mais recente continua a ser 1.0.1.998. Melhoria: o agente agora pode determinar qual versão do Ruby está instalada em uma instância, para que ele possa invocar o script `codedeploy-agent` usando essa versão.
1.0.1.1011.1	17 de agosto de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Melhoria: remoção das alterações introduzidas pela versão 1.0.1.1011 devido a problemas com suporte de shell. Essa versão do agente é funcionalmente equivalente à versão 1.0.1.998, lançada em 11 de julho de 2016.

Versão	Data de lançamento	Detalhes
1.0.1.1011	15 de agosto de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. O CodeDeploy agente para instâncias Amazon Linux, RHEL e Ubuntu Server foi atualizado com as seguintes alterações. Para instâncias do Windows Server, a versão mais recente continua a ser 1.0.1.998. Funcionalidade: Foi adicionado suporte para invocar o CodeDeploy agente usando o shell bash em sistemas operacionais em que o sistema systemd init está em uso. Aprimoramento: suporte ativado para todas as versões do Ruby 2.x no CodeDeploy agente e no atualizador do agente. CodeDeploy CodeDeploy Agentes atualizados não dependem mais apenas do Ruby 2.0. (O Ruby 2.0 ainda é necessário para as versões deb e rpm do instalador do CodeDeploy agente.)
1.0.1.998	11 de julho de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Aprimoramento: suporte fixo para executar o CodeDeploy agente com perfis de usuário diferentes do root. A variável denominada `USER` foi substituída por `CODEDEPLOY_USER` para evitar conflitos com variáveis ambientais.

Versão	Data de lançamento	Detalhes
1.0.1.966	16 de junho de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Funcionalidade: Foi introduzido o suporte para executar o CodeDeploy agente com perfis de usuário diferentes do root. Melhoria: suporte corrigido para especificar o número de revisões de aplicativo que você deseja que o agente do CodeDeploy arquive para um grupo de implantação. Aprimoramento: tornou o CodeDeploy agente compatível com a versão 2.3 do AWS SDK para Ruby (aws-sdk-core 2.3). Aprimoramento: foram corrigidos problemas com a UTF-8 codificação durante as implantações. Melhoria: melhor precisão ao identificar nomes de processos.
1.0.1.950	24 de março de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Recurso: suporte adicionado para proxies de instalação. Aprimoramento: Atualizou o script de instalação para não baixar o CodeDeploy agente se a versão mais recente já estiver instalada.
1.0.1.934	11 de fevereiro de 2016	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Recurso: suporte introduzido para especificar o número de revisões de aplicativo que você deseja que o agente do CodeDeploy arquive para um grupo de implantação.

Versão	Data de lançamento	Detalhes
1.0.1.880	11 de janeiro de 2016	Observação: esta versão não é mais compatível e pode causar falhas nas implantações. Aprimoramento: tornou o CodeDeploy agente compatível com a versão 2.2 do AWS SDK para Ruby (aws-sdk-core 2.2). Ainda há suporte para a versão 2.1.2.
1.0.1.854	17 de novembro de 2015	Observação: esta versão não é mais compatível. Se você usar esta versão, suas implantações poderão falhar. Funcionalidade: Foi introduzido o suporte para o algoritmo de SHA-256 hash. Recurso: suporte para controle de versão introduzido em arquivos `.version`. Recurso: o ID do grupo de implantação foi disponibilizado com o uso de uma variável de ambiente. Aprimoramento: Foi adicionado suporte para monitorar os registros CodeDeploy do agente usando o Amazon CloudWatch Logs.

Para obter informações relacionadas, consulte o seguinte:

- Determine a versão do CodeDeploy agente
- Instale o CodeDeploy agente

Para obter um histórico das versões do CodeDeploy agente, consulte o repositório Release em GitHub.

Gerenciando o CodeDeploy processo

O CodeDeploy agente usa o systemd para gerenciar o processo do agente em todas as distribuições Linux suportadas.

Para a versão 2.0.x e posterior, o agente envia somente uma unidade systemd nativa ()`codedeploy-agent.service`. Não há `/etc/init.d/codedeploy-agent` roteiro. Use `systemctl` para gerenciar o agente ou invocar o binário diretamente com os `stop` `status` subcomandos`start`,`restart`, ou.

Para a versão 1.8.x e anteriores, as distribuições rpm e deb são fornecidas com scripts de inicialização em. `/etc/init.d/codedeploy-agent` Em sistemas que suportam systemd, o uso do `service` comando pode executar inadvertidamente os scripts init.d em vez de permitir que o systemd gerencie o processo. Use `systemctl` para operações de agente para evitar esse problema.

Para gerenciar o CodeDeploy agente, use o `systemctl` utilitário:

```
sudo systemctl start codedeploy-agent
sudo systemctl stop codedeploy-agent
sudo systemctl restart codedeploy-agent
sudo systemctl status codedeploy-agent
```

No macOS, o agente é executado como o daemon launchd`com.amazon.aws.codedeploy-agent`, definido em`/Library/LaunchDaemons/com.amazon.aws.codedeploy-agent.plist`. launchd inicia o agente na inicialização e o reinicia se ele falhar. Para gerenciar o agente no macOS, use o `launchctl` utilitário:

```
sudo launchctl bootstrap system /Library/LaunchDaemons/com.amazon.aws.codedeploy-
agent.plist
sudo launchctl bootout system/com.amazon.aws.codedeploy-agent
sudo launchctl kickstart -k system/com.amazon.aws.codedeploy-agent
sudo launchctl print system/com.amazon.aws.codedeploy-agent
```

Esses comandos iniciam, param, reiniciam e mostram o status do agente, respectivamente.

Limpeza de revisões de aplicativo e arquivos de log

O CodeDeploy agente arquiva revisões e arquivos de log nas instâncias. O CodeDeploy agente limpa esses artefatos para economizar espaço em disco.

Registros de implantação da revisão do aplicativo: você pode usar a opção:max_revisions: no arquivo de configuração do agente para especificar o número de revisões do aplicativo a serem arquivadas inserindo qualquer número inteiro positivo. CodeDeploy também arquiva os arquivos

de log dessas revisões. Todos os outros são excluídos, com exceção do arquivo de log da última implantação bem-sucedida. Esse arquivo de log sempre é mantido, mesmo que o número de implantações com falha exceda o número de revisões mantidas. Se nenhum valor for especificado, CodeDeploy retém as cinco revisões mais recentes, além da revisão atualmente implantada.

CodeDeploy logs: para instâncias do Amazon Linux, Ubuntu Server, RHEL e macOS, CodeDeploy o agente gira os arquivos de log sob a pasta. `/var/log/aws/codedeploy-agent` O arquivo de log é rotacionado diariamente às 00:00:00 (horário da instância) ou quando o arquivo atinge 64 MB, o que ocorrer primeiro. Se uma segunda rotação ocorrer no mesmo dia, o agente acrescentará um sufixo numérico (por exemplo,). `codedeploy-agent.20260715.1.log` O agente exclui os arquivos de log após sete dias. O padrão de nomeação para arquivos de log em rodízio é `codedeploy-agent.`*`YYYYMMDD`*`.log`.

Arquivos instalados pelo CodeDeploy agente

O CodeDeploy agente armazena revisões, histórico de implantação e scripts de implantação em seu diretório raiz em uma instância. O nome e o local padrão desse diretório é:

`'/opt/codedeploy-agent/deployment-root'`para instâncias Amazon Linux, Ubuntu Server, RHEL e macOS.

`'C:\ProgramData\Amazon\CodeDeploy'` para instâncias do Windows Server.

Você pode usar a configuração root_dir no arquivo de configuração do CodeDeploy agente para configurar o nome e a localização do diretório. Para obter mais informações, consulte CodeDeploy referência de configuração do agente.

Veja a seguir um exemplo do arquivo e da estrutura do diretório no diretório raiz. A estrutura pressupõe que há N números de grupos de implantação, e que cada grupo de implantação contém N números de implantações.

```
|--deployment-root/
|-- deployment group 1 ID
|    |-- deployment 1 ID
|    |    |-- Contents and logs of the deployment's revision
|    |-- deployment 2 ID
|    |    |-- Contents and logs of the deployment's revision
|    |-- deployment N ID
|    |    |-- Contents and logs of the deployment's revision
|-- deployment group 2 ID
|    |-- deployment 1 ID
```

```
|    |    |-- bundle.tar
|    |    |-- deployment-archive
|    |    |    | -- contents of the deployment's revision
|    |    |-- logs
|    |    |    | -- scripts.log
|    |-- deployment 2 ID
|    |    |-- bundle.tar
|    |    |-- deployment-archive
|    |    |    | -- contents of the deployment's revision
|    |    |-- logs
|    |    |    | -- scripts.log
|    |-- deployment N ID
|    |    |-- bundle.tar
|    |    |-- deployment-archive
|    |    |    | -- contents of the deployment's revision
|    |    |-- logs
|    |    |    | -- scripts.log
|-- deployment group N ID
|    |-- deployment 1 ID
|    |    |-- Contents and logs of the deployment's revision
|    |-- deployment 2 ID
|    |    |-- Contents and logs of the deployment's revision
|    |-- deployment N ID
|    |    |-- Contents and logs of the deployment's revision
|-- deployment-instructions
|    |-- [deployment group 1 ID]_cleanup
|    |-- [deployment group 2 ID]_cleanup
|    |-- [deployment group N ID]_cleanup
|    |-- [deployment group 1 ID]_install.json
|    |-- [deployment group 2 ID]_install.json
|    |-- [deployment group N ID]_install.json
|    |-- [deployment group 1 ID]_last_successful_install
|    |-- [deployment group 2 ID]_last_successful_install
|    |-- [deployment group N ID]_last_successful_install
|    |-- [deployment group 1 ID]_most_recent_install
|    |-- [deployment group 2 ID]_most_recent_install
|    |-- [deployment group N ID]_most_recent_install
|-- deployment-logs
|    |-- codedeploy-agent-deployments.log
```

- As pastas do ID do grupo de implantação representam cada um de seus grupos de implantação. Um nome de diretório do grupo de implantação é o seu ID (por exemplo, `acde1916-9099-7caf-`

`fd21-012345abcdef`). Cada diretório do grupo de implantação contém um subdiretório para cada tentativa de implantação nesse grupo de implantação.

Você pode usar o comando batch-get-deployments para localizar um ID de grupo de implantação.

- As pastas do ID de implantação representam cada implantação em um grupo de implantação. Cada nome de diretório de implantação é o seu ID. Cada pasta contém:
  - bundle.tar, um arquivo compactado com o conteúdo da revisão da implantação. Use um utilitário de descompactação zip caso você queira ver a revisão.
  - deployment-archive, um diretório com o conteúdo da revisão da implantação.
  - logs, um diretório que contém um arquivo `scripts.log`. Esse arquivo lista a saída de todos os scripts especificados no AppSpec arquivo da implantação.

  Se você quiser encontrar a pasta para uma implantação, mas não souber sua ID de implantação ou ID do grupo de implantação, poderá usar o AWS CodeDeploy console ou o AWS CLI para encontrá-los. Para obter mais informações, consulte Exibir detalhes CodeDeploy da implantação.

  O número máximo padrão de implantações que podem ser arquivadas em um grupo de implantação é cinco. Quando esse número é atingido, as implementações futuras são arquivadas, e o arquivo mais antigo é excluído. Você pode usar a configuração max_revisions no arquivo de configuração do CodeDeploy agente para alterar o padrão. Para obter mais informações, consulte CodeDeploy referência de configuração do agente.

  > **Note**
  >
  > Se você deseja recuperar o espaço em disco usado por implantações arquivadas, atualize a configuração max_revisions para um número baixo, como um ou dois. A próxima implantação exclui implantações arquivadas para que o número seja igual ao número que você especificou.

- deployment-instructions contém quatro arquivos de texto para cada grupo de implantação:
  - [ID do grupo de implantação] -cleanup, um arquivo de texto com uma versão de desfazer de cada comando executado durante uma implantação. Um nome de arquivo de exemplo é `acde1916-9099-7caf-fd21-012345abcdef-cleanup`.
  - [ID do grupo de implantação]-install.json, um arquivo JSON criado durante a implantação mais recente. Ele contém os comandos executados durante a implantação. Um nome de arquivo de exemplo é `acde1916-9099-7caf-fd21-012345abcdef-install.json`.

  - [ID do grupo de implantação] _last_successful_install, um arquivo de texto que lista o diretório de arquivamento da última implantação bem-sucedida. Esse arquivo é criado quando o CodeDeploy agente copia todos os arquivos do aplicativo de implantação para a instância. Ele é usado pelo CodeDeploy agente durante a próxima implantação para determinar quais `BeforeInstall` scripts `ApplicationStop` devem ser executados. Um nome de arquivo de exemplo é `acde1916-9099-7caf-fd21-012345abcdef_last_successful_install`.
  - [ID do grupo de implantação]_most_recent_install, um arquivo de texto que lista o nome do diretório do arquivo da implantação mais recente. Esse arquivo é criado quando os arquivos na implantação são baixados com sucesso. O arquivo [ID do grupo de implantação] _last_successful_install é criado após esse arquivo, quando os arquivos baixados são copiados para o destino final. Um nome de arquivo de exemplo é `acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.
- deployment-logs contém os seguintes arquivos de log:
  - Os arquivos codedeploy-agent.yyyymmdd.log são criados para cada dia em que há uma implantação. Cada arquivo de log contém informações sobre as implantações do dia. Esses arquivos de log podem ser úteis para a depuração de problemas, como um problema de permissões. O arquivo de log é inicialmente nomeado `codedeploy-agent.log`. No dia seguinte, a data de suas implantações é inserida no nome do arquivo. Por exemplo, se hoje é 3 de janeiro de 2018, você pode ver informações de todas as implantações de hoje em `codedeploy-agent.log`. Amanhã, 4 de janeiro de 2018, o arquivo de log será renomeado `codedeploy-agent.20180103.log`.
  - O codedeploy-agent-deployments.log compila o conteúdo dos arquivos `scripts.log` para cada implantação. Os arquivos `scripts.log` estão localizados na subpasta `logs` em cada pasta `Deployment ID`. As entradas nesse arquivo tem um ID de implantação no prefixo. Por exemplo, "`[d-ABCDEF123]LifecycleEvent - BeforeInstall`" pode ser gravado durante uma implantação com um ID de `d-ABCDEF123`. Quando `codedeploy-agent-deployments.log` atinge seu tamanho máximo, o CodeDeploy agente continua gravando nele enquanto exclui o conteúdo antigo.

Gerenciando as operações CodeDeploy do agente

As instruções nesta seção mostram como instalar, desinstalar, reinstalar ou atualizar o CodeDeploy agente e como verificar se o CodeDeploy agente está em execução.

> **Tip**
>
> Algumas instruções nesta página variam de acordo com a versão do agente. Se você não tiver certeza de qual versão está instalada, execute `/opt/codedeploy-agent/bin/codedeploy-agent --version` (Linux ou macOS) `& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version` ou (Windows). Se esse comando não for reconhecido, você está executando a versão 1.8.x ou anterior. Para obter mais informações, consulte Determine a versão do CodeDeploy agente.

Tópicos

- Verifique se o CodeDeploy agente está em execução
- Determine a versão do CodeDeploy agente
- Instale o CodeDeploy agente
- Atualize o CodeDeploy agente
- Desinstale o CodeDeploy agente
- Enviar registros CodeDeploy do agente para CloudWatch

Verifique se o CodeDeploy agente está em execução

Esta seção descreve os comandos a serem executados se você suspeitar que o CodeDeploy agente parou de ser executado em uma instância.

Tópicos

- Verifique se o CodeDeploy agente para Amazon Linux ou RHEL está em execução
- Verifique se o CodeDeploy agente do Ubuntu Server está em execução
- Verifique se o CodeDeploy agente para Windows Server está em execução
- Verifique se o CodeDeploy agente para macOS está em execução

Verifique se o CodeDeploy agente para Amazon Linux ou RHEL está em execução

Para ver se o CodeDeploy agente está instalado e em execução, faça login na instância e execute o seguinte comando:

```
systemctl status codedeploy-agent
```

Se o comando retornar um erro, o CodeDeploy agente não está instalado. Instale-o conforme descrito em Instale o CodeDeploy agente para Amazon Linux ou RHEL.

Se o CodeDeploy agente estiver instalado e em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.` (código de saída 0).

Se você ver uma mensagem como `The AWS CodeDeploy agent is not running.` (código de saída 3), inicie o serviço e execute os dois comandos a seguir, um por vez:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Para a versão 2.0.x e posterior, o `codedeploy-agent status` comando retorna os códigos de LSB-compatible saída: `0` (em execução), `3` (não em execução), `4` (desconhecido). Ao usar`systemctl status`, o estado da unidade é exibido (por exemplo,`Active: active (running)`).

Verifique se o CodeDeploy agente do Ubuntu Server está em execução

Para ver se o CodeDeploy agente está instalado e em execução, faça login na instância e execute o seguinte comando:

```
systemctl status codedeploy-agent
```

Se o comando retornar um erro, o CodeDeploy agente não está instalado. Instale-o conforme descrito em Instale o CodeDeploy agente para o Ubuntu Server.

Se o CodeDeploy agente estiver instalado e em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.` (código de saída 0).

Se você ver uma mensagem como `The AWS CodeDeploy agent is not running.` (código de saída 3), inicie o serviço e execute os dois comandos a seguir, um por vez:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Verifique se o CodeDeploy agente para Windows Server está em execução

Para ver se o CodeDeploy agente está instalado e em execução, faça login na instância e execute o seguinte comando:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Você deve ver uma saída semelhante a:

```
Status   Name               DisplayName
------   ----               -----------
Running codedeployagent    CodeDeploy Host Agent Service
```

Se o comando retornar um erro, o CodeDeploy agente não está instalado. Instale-o conforme descrito em Instale o CodeDeploy agente para Windows Server.

Se `Status` mostrar algo diferente de `Running`, inicie o serviço com o seguinte comando:

```
powershell.exe -Command Start-Service -Name codedeployagent
```

É possível reiniciar o serviço com o seguinte comando:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

É possível interromper o serviço com o seguinte comando:

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Verifique se o CodeDeploy agente para macOS está em execução

Para ver se o CodeDeploy agente está instalado e em execução, faça login na instância e execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Se o comando retornar um erro porque o arquivo não foi encontrado, o CodeDeploy agente não está instalado. Instale-o conforme descrito em Instale o CodeDeploy agente para macOS.

Se o CodeDeploy agente estiver instalado e em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.` (código de saída 0).

Se você ver uma mensagem como `The AWS CodeDeploy agent is not running.` (código de saída 3), inicie o agente com o seguinte comando:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

Se esse comando informar que o serviço não pôde ser encontrado, o daemon lançado não será carregado. Carregue e inicie com o seguinte comando:

```
sudo launchctl bootstrap system /Library/LaunchDaemons/com.amazon.aws.codedeploy-
agent.plist
```

Para ver o estado do daemon lançado, execute:

```
sudo launchctl print system/com.amazon.aws.codedeploy-agent
```

Determine a versão do CodeDeploy agente

Você pode determinar a versão do CodeDeploy agente em execução na sua instância de duas maneiras.

Primeiro, para a versão 2.0.x e posterior, use o `codedeploy-agent --version` comando em todas as plataformas.

Em segundo lugar, você pode ver a versão em um `.version` arquivo na instância. No Amazon Linux, RHEL, Ubuntu Server e macOS, o arquivo está localizado `/opt/codedeploy-agent/.version` em, e no Windows Server em. `C:\ProgramData\Amazon\CodeDeploy\.version` O conteúdo do arquivo varia de acordo com a versão:

- Para a versão 2.0.x e posterior, o arquivo contém o número da versão simples, no formato. `agent_version: 2.0.0` Esse valor corresponde à versão que o agente reporta ao CodeDeploy serviço.
- Para a versão 1.8.x e anteriores, a string da versão usa o formato `OFFICIAL_1.0.1.854_rpm` (ou`_deb`,`_msi`).

Tópicos

- Determinar a versão no Amazon Linux ou no RHEL
- Determinar a versão no Ubuntu Server
- Determinar a versão no Windows Server
- Determine a versão no macOS
- Determine a versão mais recente do agente disponível em um Região da AWS

Determinar a versão no Amazon Linux ou no RHEL

Entre na instância e execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Para versões anteriores à 2.0.0, você também pode usar:

```
sudo yum info codedeploy-agent
```

Determinar a versão no Ubuntu Server

Entre na instância e execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Para versões anteriores à 2.0.0, você também pode usar:

```
sudo dpkg -s codedeploy-agent
```

Determinar a versão no Windows Server

Entre na instância e execute o seguinte comando:

```
& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version
```

Determine a versão no macOS

Entre na instância e execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Você também pode ver a versão do pacote instalado:

```
pkgutil --pkg-info com.amazon.aws.codedeploy-agent
```

Determine a versão mais recente do agente disponível em um Região da AWS

Para descobrir qual versão é a mais recente disponível na qual Região da AWS você implanta, consulte o bucket do kit de CodeDeploy recursos dessa região. Cada bucket regional contém um `latestv2/LATEST_VERSION` arquivo que nomeia os pacotes de agentes atualmente lançados nessa região. O arquivo pode ser lido publicamente, então você não precisa de credenciais para recuperá-lo.

```
curl https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/LATEST_VERSION
```

Substitua *`bucket-name`* e *`region-identifier`* com o bucket do kit de recursos e o identificador de região da região em que você implanta. Para ver a lista, consulte Nomes de bucket do kit de recursos por região.

Esse comando retorna uma saída semelhante à seguinte:

```
{
    "rpm_x86_64": "releases/codedeploy-agent-2.1.0-9745926189.x86_64.rpm",
    "rpm_aarch64": "releases/codedeploy-agent-2.1.0-9745926189.aarch64.rpm",
    "deb_x86_64": "releases/codedeploy-agent_2.1.0-9745926189_amd64.deb",
    "deb_aarch64": "releases/codedeploy-agent_2.1.0-9745926189_arm64.deb",
    "msi": "releases/codedeploy-agent-2.1.0-9745926189.msi"
}
```

Cada chave nomeia um tipo de pacote e, para RPM e DEB, a arquitetura da instância também. Cada valor é a chave de objeto do Amazon S3 desse pacote, e a versão do agente e o número da compilação fazem parte do nome do arquivo, no formato. *`agent-version`*-*`build-number`* Na saída anterior, a versão mais recente disponível na região é a 2.1.0.

Para a versão 2.1.2 e posterior, o arquivo também tem uma `pkg_aarch64` chave, que nomeia o pacote do instalador do macOS.

Para instalar ou atualizar para a versão que esse arquivo nomeia, consulte Instale o CodeDeploy agente Atualize o CodeDeploy agente e. Para ver os recursos incluídos em cada versão, consulteHistórico de versões do CodeDeploy agente.

Instale o CodeDeploy agente

Para usar CodeDeploy em instâncias do EC2 ou servidores locais, o CodeDeploy agente deve ser instalado primeiro. Você pode instalar o CodeDeploy agente com AWS Systems Manager ou com a linha de comando. Para obter mais informações sobre o Systems Manager, consulte O que é o AWS Systems Manager.

Para obter a versão recomendada para instalação, consulte Histórico de versões do CodeDeploy agente.

Tópicos

- Instale o CodeDeploy agente usando AWS Systems Manager
- Instale o CodeDeploy agente usando a linha de comando

Instale o CodeDeploy agente usando AWS Systems Manager

Você pode usar o Console de gerenciamento da AWS ou o AWS CLI para instalar o CodeDeploy agente em seu Amazon EC2 ou em instâncias locais usando. AWS Systems Manager Você pode optar por instalar uma versão específica ou sempre instalar a versão mais recente do agente. Para obter mais informações sobre AWS Systems Manager, consulte O que é AWS Systems Manager.

Você também pode instalar o CodeDeploy agente a partir de um bucket do Amazon S3. Para obter informações sobre o uso de um link para download do Amazon S3, consulte Instale o CodeDeploy agente usando a linha de comando.

Tópicos

- Pré-requisitos
- Instale o CodeDeploy agente

Pré-requisitos

Siga as etapas em Começando com CodeDeploy para configurar as permissões do IAM e a AWS CLI.

Se estiver instalando o CodeDeploy agente em um servidor local com o Systems Manager, você deverá registrar seu servidor local no Amazon EC2 Systems Manager. Para obter mais informações, consulte Configurar o Systems Manager em ambientes híbridos no Manual do usuário do AWS Systems Manager .

Instale o CodeDeploy agente

Antes de usar o Systems Manager para instalar o CodeDeploy agente, você deve se certificar de que a instância esteja configurada corretamente para o Systems Manager.

Instalar ou atualizar o agente SSM

Em uma instância do Amazon EC2, o CodeDeploy agente exige que a instância esteja executando a versão 2.3.274.0 ou posterior. Antes de instalar o CodeDeploy agente, atualize ou instale o agente SSM na instância, caso ainda não tenha feito isso.

O agente SSM vem pré-instalado em algumas AMIs do Amazon EC2 fornecidas pela. AWS Para obter mais informações, consulte Imagens de máquina da Amazon (AMI) com o agente SSM pré-instalado.

> **Note**
>
> Certifique-se de que o sistema operacional da instância também seja suportado pelo CodeDeploy agente. Para obter mais informações, consulte Sistemas operacionais suportados pelo CodeDeploy agente.

Para obter informações sobre como instalar ou atualizar o agente SSM em uma instância que execute o Linux, consulte Instalar e configurar o agente SSM em instâncias do Linux no Manual do usuário do AWS Systems Manager .

Para obter informações sobre como instalar ou atualizar o agente SSM em uma instância que executa o Windows Server, consulte Instalar e configurar o agente SSM em instâncias do Windows no Manual do usuário do AWS Systems Manager .

Para obter informações sobre como instalar ou atualizar o agente SSM em uma instância do EC2 Mac, consulte Instalação e desinstalação manual do agente SSM em instâncias do EC2 para macOS no Guia do usuário.AWS Systems Manager

(Opcional) Verificar os pré-requisitos do Systems Manager

Antes de usar o Systems Manager Run Command para instalar o CodeDeploy agente, verifique se suas instâncias atendem aos requisitos mínimos do Systems Manager. Para obter mais informações, consulte Configurar o AWS Systems Manager no Manual do usuário do AWS Systems Manager .

Instale o CodeDeploy agente

Com o SSM, você pode instalar CodeDeploy uma única vez ou configurar um cronograma para instalar novas versões.

Para instalar o CodeDeploy agente, escolha o pacote Systems Manager Distributor para a versão do agente desejada e siga as etapas em Instalar ou atualizar pacotes com o AWS Systems Manager distribuidor. O pacote difere de acordo com a versão do agente:

- Para a versão 2.0.x e posterior, escolha o `AWSCodeDeployAgentV2` pacote. Esse é o único pacote compatível com macOS, começando com a versão 2.1.2.
- Para a versão 1.8.x e anteriores, escolha o `AWSCodeDeployAgent` pacote.

Instale o CodeDeploy agente usando a linha de comando

Use os tópicos a seguir para instalar e executar o CodeDeploy agente usando a linha de comando.

Tópicos

- Instale o CodeDeploy agente para Amazon Linux ou RHEL
- Instale o CodeDeploy agente para o Ubuntu Server
- Instale o CodeDeploy agente para Windows Server
- Instale o CodeDeploy agente para macOS

Instale o CodeDeploy agente para Amazon Linux ou RHEL

Entre na instância e execute os seguintes comandos, um por vez. Executar primeiro o comando `sudo yum update` é considerado uma prática recomendada ao usar `yum` para instalar pacotes, mas você poderá ignorá-lo se não desejar atualizar todos os pacotes.

```
sudo yum update
```

Para a versão 2.0.x e posterior:

```
sudo yum install wget
```

Para a versão 1.8.x e anteriores:

```
sudo yum install ruby wget
```

(Opcional) Para limpar a AMI de qualquer informação anterior de armazenamento em cache do agente, execute o seguinte script:

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Mude para o diretório inicial.

```
cd /home/ec2-user
```

> **Note**
>
> No comando anterior, `/home/ec2-user` representa o nome de usuário padrão para uma instância do Amazon EC2 para Amazon Linux ou RHEL. Se a sua instância foi criada usando um AMI personalizado, o proprietário do AMI pode ter especificado um nome de usuário padrão diferente.

Baixe o instalador do CodeDeploy agente. O `latestv2/` prefixo serve para o instalador da versão 2.0.x e o `latest/` prefixo serve para o instalador da versão 1.8.x.

Para a versão 2.0.x e posterior:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Para a versão 1.8.x e anteriores:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*`bucket-name`*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit da sua região e *`region-identifier`* é o identificador da sua região.

Por exemplo, para a versão 2.0.x e posterior:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Para a versão 1.8.x e anteriores:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Para obter uma lista dos nomes de bucket e identificadores de região, consulte Nomes de bucket do kit de recursos por região.

Defina as permissões de execução no arquivo `install`:

```
chmod +x ./install
```

Para instalar a versão mais recente do CodeDeploy agente:

-
```
sudo ./install auto
```

Para instalar uma versão específica do CodeDeploy agente:

- Liste as versões disponíveis na sua região:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
 | grep '\.rpm$'
```

- Instale uma das versões:

  Para a versão 2.0.x e posteriores, os pacotes são específicos da arquitetura:

```
sudo ./install auto -v releases/codedeploy-agent-version.arch.rpm
```

  Onde *arch* está `x86_64` ou `aarch64` depende da arquitetura da sua instância.

  Para a versão 1.8.x e anteriores, os pacotes usam: `noarch`

```
sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
```

  Para obter a versão mais recente lançada, consulteHistórico de versões do CodeDeploy agente.

Para verificar se o serviço está em execução, execute o seguinte comando:

```
systemctl status codedeploy-agent
```

Para a versão 2.0.x e posterior, se o agente estiver em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.`

Para a versão 1.8.x e anteriores, se o agente estiver em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.`

Se o agente não estiver em execução, inicie o serviço e verifique o status:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Instale o CodeDeploy agente para o Ubuntu Server

Para instalar o CodeDeploy agente no Ubuntu Server

1. Faça login na instância.
2. Insira o seguinte comando, um após o outro:

   ```
   sudo apt update
   ```

   Para a versão 2.0.x e posterior:

   ```
   sudo apt install wget
   ```

   Para a versão 1.8.x e anteriores:

   ```
   sudo apt install ruby-full wget
   ```

3. Digite o comando:

   ```
   cd /home/ubuntu
   ```

*/home/ubuntu*representa o nome de usuário padrão para uma instância do Ubuntu Server. Se a sua instância foi criada usando um AMI personalizado, o proprietário do AMI pode ter especificado um nome de usuário padrão diferente.

4. Insira o comando da a seguir. O `latestv2/` prefixo serve para o instalador da versão 2.0.x e o `latest/` prefixo serve para o instalador da versão 1.8.x.

   Para a versão 2.0.x e posterior:

   ```
   wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
   ```

   Para a versão 1.8.x e anteriores:

   ```
   wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
   ```

   *bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit da sua região e *region-identifier* é o identificador da sua região.

   Por exemplo, para a versão 2.0.x e posterior:

   `https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

   Para a versão 1.8.x e anteriores:

   `https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

   Para obter uma lista dos nomes de bucket e identificadores de região, consulte Nomes de bucket do kit de recursos por região.

5. Digite o comando:

   ```
   chmod +x ./install
   ```

6. Execute um destes procedimentos:

   - Para instalar a versão mais recente do CodeDeploy agente:

     ```
     sudo ./install auto
     ```

- Para instalar uma versão específica do CodeDeploy agente:
  - Liste as versões disponíveis na sua região:

    ```
    aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.deb$'
    ```

  - Instale uma das versões:

    Para a versão 2.0.x e posteriores, os pacotes são específicos da arquitetura:

    ```
    sudo ./install auto -v releases/codedeploy-agent_version_arch.deb
    ```

    Onde *arch* está `amd64` ou `arm64` depende da arquitetura da sua instância.

    Para a versão 1.8.x e anteriores, os pacotes usam uma convenção de nomenclatura diferente:

    ```
    sudo ./install auto -v releases/codedeploy-agent_version_all.deb
    ```

    Para obter a versão mais recente lançada, consulteHistórico de versões do CodeDeploy agente.

Para verificar se o serviço está em execução

1. Digite o comando:

   ```
   systemctl status codedeploy-agent
   ```

   Se o CodeDeploy agente estiver instalado e em execução:

   Para a versão 2.0.x e posterior, você deve ver: `The AWS CodeDeploy agent is running.`

   Para a versão 1.8.x e anteriores, você deve ver: `The AWS CodeDeploy agent is running.`
2. Se o agente não estiver em execução, inicie o serviço e verifique o status:

   ```
   systemctl start codedeploy-agent
   ```

```
systemctl status codedeploy-agent
```

Instale o CodeDeploy agente para Windows Server

Nas instâncias do Windows Server, você pode usar um desses métodos para baixar e instalar o CodeDeploy agente:

- Uso AWS Systems Manager (recomendado)
- Execute uma série de PowerShell comandos do Windows.
- Escolha um link de download direto.
- Execute um comando de cópia do Amazon S3.

> **Note**
>
> A pasta na qual o CodeDeploy agente está instalado é`C:\ProgramData\Amazon\CodeDeploy`. Não coloque junções de diretórios ou links simbólicos nesse caminho. Para a versão 2.0.x e posterior, o agente impõe esse requisito e se recusa a iniciar se o binário do serviço não estiver inativo. `C:\ProgramData\Amazon\CodeDeploy\`

Tópicos

- Usar o Systems Manager
- Use o Windows PowerShell
- Usar um link direto
- Usar um comando de cópia do Amazon S3

Usar o Systems Manager

Siga as instruções Instale o CodeDeploy agente usando AWS Systems Manager para instalar o CodeDeploy agente.

Use o Windows PowerShell

Faça login na instância e execute os seguintes comandos no Windows PowerShell:

1. Exija que todos os scripts e arquivos de configuração baixados da Internet sejam assinados por um editor confiável. Se você precisar alterar a política de execução, digite "**Y**".

```
Set-ExecutionPolicy RemoteSigned
```

2. Carregue AWS Tools for Windows PowerShell o.

```
Import-Module AWSPowerShell
```

3. Crie um diretório no qual o arquivo de instalação do CodeDeploy agente é baixado.

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

4. Configure AWS as credenciais usando os `Initialize-AWSDefaultConfiguration` comandos `Set-AWSCredential` e. Para obter mais informações, consulte Usando AWS credenciais nas AWS ferramentas do Guia PowerShell do Usuário.

5. Faça o download do arquivo de instalação do CodeDeploy agente.

> **Note**
>
> Para obter a versão mais recente lançada, consulteHistórico de versões do CodeDeploy agente.

Para instalar a versão mais recente do CodeDeploy agente. O `latestv2/` prefixo serve para o instalador da versão 2.x (2.0.x e posterior), e o `latest/` prefixo serve para o instalador da versão 1.8.x.

- Para a versão 2.0.x e posterior:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latestv2/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Para a versão 1.8.x e anteriores:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Para instalar uma versão específica do CodeDeploy agente:

- ```
  powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/
  codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
  ```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit para sua região. Por exemplo, para a região Leste dos EUA (Ohio), *bucket-name* substitua poraws-codedeploy-us-east-2. Para obter uma lista de nomes de buckets, consulte Nomes de bucket do kit de recursos por região.

6. Execute o arquivo de instalação do CodeDeploy agente.

   ```
   c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
   ```

Para verificar se o serviço está em execução, execute o seguinte comando:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Se o CodeDeploy agente acabou de ser instalado e não foi iniciado, depois de executar o Get-Service comando, em Status, você deverá ver**`Start...`**:

```
Status     Name               DisplayName
------     ----               -----------
Start...   codedeployagent    CodeDeploy Host Agent Service
```

Se o CodeDeploy agente já estiver em execução, depois de executar o Get-Service comando, em Status, você deverá ver**`Running`**:

```
Status     Name               DisplayName
------     ----               -----------
Running    codedeployagent    CodeDeploy Host Agent Service
```
```

Usar um link direto

Se as configurações de segurança do navegador na instância do Windows Server fornecerem as permissões (por exemplo, para`https://s3.*.amazonaws.com`), você poderá usar um link direto para sua região para baixar o CodeDeploy agente e executar o instalador manualmente.

O `latestv2/` prefixo serve para o instalador da versão 2.x (2.0.x e posterior), e o `latest/` prefixo serve para o instalador da versão 1.8.x.

Para a versão 2.0.x e posterior, o link é:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latestv2/codedeploy-agent.msi
```

Para a versão 1.8.x e anteriores:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... onde *`region`* está a AWS região em que você está implantando seu aplicativo.

Por exemplo, para a versão 2.0.x e posterior:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi
```

> **Important**
>
> Obtenha o `.msi` arquivo da mesma região do seu CodeDeploy aplicativo. Escolher uma região diferente pode causar falhas de `inconsistent region` no arquivo `codedeploy-agent-log` quando você executar o arquivo `.msi`.

Usar um comando de cópia do Amazon S3

Se o AWS CLI estiver instalado na instância, você poderá usar o comando cp do Amazon S3 para baixar o CodeDeploy agente e, em seguida, executar o instalador manualmente. Para obter informações, consulte Instalar o AWS Command Line Interface no Microsoft Windows.

O `latestv2/` prefixo serve para o instalador da versão 2.x (2.0.x e posterior), e o `latest/` prefixo serve para o instalador da versão 1.8.x.

Para a versão 2.0.x e posterior, o comando do Amazon S3 é:

```
aws s3 cp s3://aws-codedeploy-region/latestv2/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

Para a versão 1.8.x e anteriores:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

... onde *region* está a AWS região em que você está implantando seu aplicativo.

Por exemplo, para a versão 2.0.x e posterior:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi codedeploy-
agent.msi --region af-south-1
```

Instale o CodeDeploy agente para macOS

Entre na instância e execute os seguintes comandos, um por vez.

Mude para o diretório inicial.

```
cd ~
```

Baixe o pacote do instalador do CodeDeploy agente:

```
curl -O https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/codedeploy-
agent.aarch64.pkg
```

*bucket-name*é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy Resource Kit da sua região e *region-identifier* é o identificador da sua região.

Por exemplo:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg`

Para obter uma lista dos nomes de bucket e identificadores de região, consulte Nomes de bucket do kit de recursos por região.

Para instalar a versão mais recente do CodeDeploy agente:

- ```
  sudo installer -pkg codedeploy-agent.aarch64.pkg -target /
  ```

Para instalar uma versão específica do CodeDeploy agente:

- Liste as versões disponíveis na sua região:

  ```
  aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
   | grep '\.pkg$'
  ```

- Baixe e instale uma das versões:

  ```
  aws s3 cp s3://aws-codedeploy-region-identifier/releases/codedeploy-
  agent-version.aarch64.pkg codedeploy-agent.aarch64.pkg --region region-identifier
  ```

  ```
  sudo installer -pkg codedeploy-agent.aarch64.pkg -target /
  ```

  Para obter a versão mais recente lançada, consulteHistórico de versões do CodeDeploy agente.

Para verificar se o serviço está em execução, execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Se o agente estiver em execução, você deverá ver uma mensagem como `The AWS CodeDeploy agent is running.`

Se o agente não estiver em execução, inicie o serviço e verifique o status:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Atualize o CodeDeploy agente

Você pode configurar atualizações automáticas e programadas do CodeDeploy agente em todos os sistemas operacionais compatíveis usando AWS Systems Manager. Você também pode forçar atualizações em todos os sistemas operacionais com suporte, executando um comando em uma instância.
```

Tópicos

- Atualize o CodeDeploy agente no Amazon Linux ou no RHEL
- Atualize o CodeDeploy agente no Ubuntu Server
- Atualize o CodeDeploy agente no Windows Server
- Atualize o CodeDeploy agente no macOS

Atualize o CodeDeploy agente no Amazon Linux ou no RHEL

Para configurar atualizações automáticas e programadas do CodeDeploy agente usando AWS Systems Manager, siga as etapas em Instalar o CodeDeploy agente com AWS Systems Manager.

Para a versão 2.0.x e posterior, para forçar uma atualização, execute:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Como alternativa, baixe novamente e execute o `latestv2/install` script regional. Para obter instruções de instalação, consulte Instale o CodeDeploy agente para Amazon Linux ou RHEL.

Para a versão 1.8.x e anteriores, para forçar uma atualização, execute:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Atualmente, não existe nenhum caminho de atualização automática da versão 1.8.x para a 2.0.0. Para atualizar para a versão 2.0.0, instale-o com o pacote `AWSCodeDeployAgentV2` Systems Manager Distributor ou execute o `latestv2/install` script regional manualmente. Para instruções, consulte Instale o CodeDeploy agente.

Atualize o CodeDeploy agente no Ubuntu Server

Para configurar atualizações automáticas e programadas do CodeDeploy agente usando AWS Systems Manager, siga as etapas em Instalar o CodeDeploy agente com AWS Systems Manager.

Para a versão 2.0.x e posterior, para forçar uma atualização, execute:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Como alternativa, baixe novamente e execute o `latestv2/install` script regional. Para obter instruções de instalação, consulte Instale o CodeDeploy agente para o Ubuntu Server.

Para a versão 1.8.x e anteriores, para forçar uma atualização, execute:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Atualmente, não existe nenhum caminho de atualização automática da versão 1.8.x para a 2.0.0. Para atualizar para a versão 2.0.0, instale-o com o pacote `AWSCodeDeployAgentV2` Systems Manager Distributor ou execute o `latestv2/install` script regional manualmente. Para instruções, consulte Instale o CodeDeploy agente.

Atualize o CodeDeploy agente no Windows Server

Você pode ativar as atualizações automáticas do CodeDeploy agente com AWS Systems Manager. Com o Systems Manager, você pode definir uma programação para atualizar suas instâncias do Amazon EC2 ou on-premises criando uma associação com o gerenciador de estados do Systems Manager. Você também pode atualizar manualmente o CodeDeploy agente desinstalando a versão atual e instalando uma mais nova.

Tópicos

- Configure a atualização automática do CodeDeploy agente com AWS Systems Manager
- Atualize o CodeDeploy agente manualmente

Configure a atualização automática do CodeDeploy agente com AWS Systems Manager

Para configurar o Systems Manager e ativar as atualizações automáticas do CodeDeploy agente, siga as instruções em Instalar o CodeDeploy agente usando AWS Systems Manager.

Atualize o CodeDeploy agente manualmente

Para atualizar o CodeDeploy agente manualmente, você pode instalar a versão mais recente a partir da CLI ou usando o Systems Manager. Siga as instruções em Instalar o agente do CodeDeploy. É recomendável desinstalar as versões mais antigas do CodeDeploy agente seguindo as instruções em Desinstalar o CodeDeploy agente.

Atualize o CodeDeploy agente no macOS

Para configurar atualizações automáticas e programadas do CodeDeploy agente usando AWS Systems Manager, siga as etapas Instale o CodeDeploy agente usando AWS Systems Manager e escolha o `AWSCodeDeployAgentV2` pacote.

Para forçar uma atualização, execute:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Como alternativa, baixe o `.pkg` arquivo mais recente e instale-o sobre o agente existente. Para obter instruções de instalação, consulte Instale o CodeDeploy agente para macOS.

Desinstale o CodeDeploy agente

Você pode remover o CodeDeploy agente das instâncias quando ele não for mais necessário ou quando quiser realizar uma nova instalação.

Desinstale o CodeDeploy agente do Amazon Linux ou do RHEL

Para desinstalar o CodeDeploy agente, faça login na instância e execute o seguinte comando:

```
sudo yum erase codedeploy-agent
```

Desinstale o CodeDeploy agente do Ubuntu Server

Para desinstalar o CodeDeploy agente, faça login na instância e execute o seguinte comando:

```
sudo dpkg --purge codedeploy-agent
```

Desinstale o CodeDeploy agente do Windows Server

Para desinstalar o CodeDeploy agente, faça login na instância e execute os três comandos a seguir, um por vez:

```
wmic
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
exit
```

> **Note**
>
> No Windows Server 2025 e versões posteriores, `wmic` está obsoleto. Em vez disso, use PowerShell o seguinte comando:
>
> ```
> Get-ItemProperty 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*' |
>   Where-Object DisplayName -eq 'CodeDeploy Host Agent' |
>   ForEach-Object { Start-Process msiexec.exe -ArgumentList "/x
>  $($_.PSChildName) /quiet" -Wait }
> ```

Você também pode entrar na instância e, no Painel de Controle, abrir Programas e Recursos, escolher CodeDeploy Host Agent e, em seguida, escolher Desinstalar.

Desinstale o CodeDeploy agente do macOS

Para desinstalar o CodeDeploy agente, faça login na instância e execute o seguinte comando:

```
sudo /opt/codedeploy-agent/bin/uninstall.sh
```

Enviar registros CodeDeploy do agente para CloudWatch

Você pode enviar dados métricos e de registro do CodeDeploy agente para CloudWatch usar o CloudWatch agente unificado ou, mais simplesmente, o CloudWatch agente.

Use as instruções a seguir para instalar o CloudWatch agente e configurá-lo para uso com CodeDeploy agentes.

Pré-requisitos

Antes de começar, conclua as seguintes tarefas:

- Instale o CodeDeploy agente e verifique se ele está em execução. Para obter mais informações, consulte Instale o CodeDeploy agente e Verifique se o CodeDeploy agente está em execução.
- Instale o CloudWatch agente. Para obter mais informações, consulte Instalação do CloudWatch agente.
- Adicione as seguintes permissões ao perfil da instância CodeDeploy do IAM:
  - CloudWatchLogsFullAccess

- CloudWatchAgentServerPolicy

Para obter mais informações sobre o perfil da CodeDeploy instância, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2 ofComeçando com CodeDeploy.

Configurar o CloudWatch agente para coletar CodeDeploy registros

Você pode configurar o CloudWatch agente percorrendo um assistente ou criando ou editando manualmente um arquivo de configuração.

Para configurar o CloudWatch agente usando o assistente (Linux)

1. Execute o assistente, conforme descrito em Executar o assistente de configuração do CloudWatch agente.
2. No atendente, quando perguntado `Do you want to monitor any log files?`, digite **1**.
3. Especifique o arquivo de log do CodeDeploy agente, da seguinte forma:
    1. Para `Log file path` inserir o caminho para o arquivo de CodeDeploy log, por exemplo:**/var/log/aws/codedeploy-agent/codedeploy-agent.log**.
    2. Em `Log group name`, insira um nome de grupo de logs, por exemplo: **codedeploy-agent-log**.
    3. Em `Log stream name`, insira um nome de fluxo de logs, por exemplo: **{instance_id}-codedeploy-agent-log**.
4. Quando perguntado `Do you want to specify any additional log files?`, insira **1**.
5. Especifique os registros de implantação do CodeDeploy agente, da seguinte forma:
    1. Para `Log file path` inserir o caminho para o arquivo CodeDeploy de log de implantação, por exemplo:**/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log**.
    2. Em `Log group name`, insira um nome de grupo de logs, por exemplo: **codedeploy-agent-deployment-log**.
    3. Em `Log stream name`, insira um nome de fluxo de logs, por exemplo: **{instance_id}-codedeploy-agent-deployment-log**.
6. Quando perguntado `Do you want to specify any additional log files?`, insira **1**.
7. Especifique os registros do atualizador do CodeDeploy agente, da seguinte forma:

1. Para `Log file path`, insira o caminho para o arquivo de log do CodeDeploy atualizador. O caminho depende da versão do seu agente:
   - Para a versão 2.0.x e posterior, use. **`/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log`**
   - Para a versão 1.8.x e anteriores, use. **`/tmp/codedeploy-agent.update.log`**
2. Em `Log group name`, insira um nome de grupo de logs, por exemplo: **`codedeploy-agent-updater-log`**.
3. Em `Log stream name`, insira um nome de fluxo de logs, por exemplo: **`{instance_id}-codedeploy-agent-updater-log`**.

Para configurar o CloudWatch agente usando o assistente (Windows)

1. Execute o assistente, conforme descrito em Executar o assistente de configuração do CloudWatch agente.
2. No atendente, quando perguntado `Do you want to monitor any customized log files?`, digite **`1`**.
3. Especifique o arquivo de CodeDeploy log, da seguinte forma:
   1. Para `Log file path`, insira o caminho para o arquivo de log do CodeDeploy agente. O caminho depende da versão do seu agente:
      - Para a versão 2.0.x e posterior, use. **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log`**
      - Para a versão 1.8.x e anteriores, use. **`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt`**
   2. Em `Log group name`, insira um nome de grupo de logs, por exemplo: **`codedeploy-agent-log`**.
   3. Em `Log stream name`, insira um nome de fluxo de logs, por exemplo: **`{instance_id}-codedeploy-agent-log`**.
4. Quando perguntado `Do you want to specify any additional log files?`, insira **`1`**.
5. Especifique os registros de implantação do CodeDeploy agente, da seguinte forma:
   1. Para `Log file path`, insira o caminho para o arquivo CodeDeploy de log de implantação:**`C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log`**.

2. Em `Log group name`, insira um nome de grupo de logs, por exemplo: **`codedeploy-agent-deployment-log`**.
3. Em `Log stream name`, insira um nome de fluxo de logs, por exemplo: **`{instance_id}-codedeploy-agent-deployment-log`**.

Para configurar o CloudWatch agente criando ou editando manualmente um arquivo de configuração (Linux)

1. Crie ou edite o arquivo de configuração do CloudWatch agente conforme descrito em Criar ou editar manualmente o arquivo de configuração do CloudWatch agente.
2. Verifique se o arquivo se chama `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` e se ele contém o seguinte código:

   O caminho do registro do atualizador difere de acordo com a versão. Para a versão 2.0.x e posterior, use. `/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log` Para a versão 1.8.x e anteriores, use. `/tmp/codedeploy-agent.update.log`

```
...
"logs": {
    "logs_collected": {
        "files": {
            "collect_list": [
                {
                    "file_path": "/var/log/aws/codedeploy-agent/codedeploy-
agent.log",
                    "log_group_name": "codedeploy-agent-log",
                    "log_stream_name": "{instance_id}-agent-log"
                },
                {
                    "file_path": "/opt/codedeploy-agent/deployment-root/deployment-
logs/codedeploy-agent-deployments.log",
                    "log_group_name": "codedeploy-agent-deployment-log",
                    "log_stream_name": "{instance_id}-codedeploy-agent-deployment-
log"
                },
                {
                    "file_path": "/var/log/aws/codedeploy-agent/codedeploy-agent-
updater.log",
                    "log_group_name": "codedeploy-agent-updater-log",
                    "log_stream_name": "{instance_id}-codedeploy-agent-updater-log"
```

```
                }
            ]
        }
    }
}
...
```

Para configurar o CloudWatch agente criando ou editando manualmente um arquivo de configuração (Windows)

1. Crie ou edite o arquivo de configuração do CloudWatch agente conforme descrito em Criar ou editar manualmente o arquivo de configuração do CloudWatch agente.
2. Verifique se o arquivo se chama `C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` e se ele contém o seguinte código:

   Para a versão 2.0.x e posterior, o nome do arquivo de log do agente é. `codedeploy-agent.log` Para a versão 1.8.x e anteriores, o nome do arquivo de log do agente é. `codedeploy-agent-log.txt` O caminho do log de implantação é o mesmo para as duas versões.

```
...
"logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\\log\
\codedeploy-agent.log",
                        "log_group_name": "codedeploy-agent-log",
                        "log_stream_name": "{instance_id}-codedeploy-agent-log"
                    },
                    {
                        "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\
\deployment-logs\\codedeploy-agent-deployments.log",
                        "log_group_name": "codedeploy-agent-deployment-log",
                        "log_stream_name": "{instance_id}-codedeploy-agent-
deployment-log"
                    }
                ]
            },
            ...
```

```
        }
    },
...
```

Reinicie o CloudWatch agente

Depois de fazer suas alterações, reinicie o CloudWatch agente conforme descrito em Iniciar o CloudWatch agente.

Trabalhando com instâncias para CodeDeploy

CodeDeploy oferece suporte a implantações em instâncias que executam Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) e Windows Server.

Você pode usar CodeDeploy para implantar tanto em instâncias do Amazon EC2 quanto em instâncias locais. Uma instância local é qualquer dispositivo físico que não seja uma instância do Amazon EC2 que possa executar CodeDeploy o agente e se conectar a endpoints de serviço AWS público. Você pode usar CodeDeploy para implantar simultaneamente um aplicativo em instâncias do Amazon EC2 na nuvem e em PCs desktop em seu escritório ou servidores em seu próprio data center.

Comparação de instâncias do Amazon EC2 com instâncias on-premises

A tabela a seguir compara instâncias do Amazon EC2 e instâncias on-premises:

Sujeito	Instâncias do Amazon EC2	On-premises Instâncias
Exige que você instale e execute uma versão do CodeDeploy agente compatível com o sistema operacional em execução na instância.	Sim	Sim
Requer que a instância seja capaz de se conectar ao serviço do CodeDeploy.	Sim	Sim
Requer que um perfil de instância do IAM seja anexado à instância. O perfil da instância do IAM precisa ter permissões para participar das CodeDeploy implantaç	Sim	Não

Sujeito	Instâncias do Amazon EC2	On-premises Instâncias
ões. Para mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.		
Requer que você siga um destes procedimentos para autenticar e registrar instâncias: • Criar um perfil do IAM que possa ser assumido por um usuário do IAM em cada instância para recuperar credenciais temporárias periodicamente atualizadas que são geradas pelo AWS Security Token Service. • Criar um usuário do IAM para cada instância e armazenar as credenciais de conta desse usuário em texto sem formatação na instância.	Não	Sim
Exige que você registre cada instância CodeDeploy antes de poder implantá-la.	Não	Sim
Exige que você marque cada instância antes de CodeDeploy poder implantá-la.	Sim	Sim

Sujeito	Instâncias do Amazon EC2	On-premises Instâncias
Pode participar dos cenários Amazon EC2 Auto Scaling e Elastic Load Balancing como parte das implantações. CodeDeploy	Sim	Não
Pode ser implantado a partir de GitHub buckets e repositór ios do Amazon S3.	Sim	Sim
Pode dar suporte a gatilhos que solicitam o envio de notificações por SMS ou e-mail quando eventos específic os ocorrerem em implement ações ou instâncias.	Sim	Sim
Está sujeito a cobrança por implantações associadas.	Não	Sim

Tarefas de instância para CodeDeploy

Para iniciar ou configurar instâncias para uso em implantações, escolha uma das seguintes instruções:

Quero iniciar uma nova instância do Amazon EC2 para Amazon Linux ou Windows Server.	Para iniciar a instância do Amazon EC2 com o mínimo de esforço, consulte Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo). Para iniciar a instância do Amazon EC2 principalmente por sua conta, consulte Crie uma instância do Amazon EC2 para

	(CodeDeploy AWS CLI ou console Amazon EC2).
Quero iniciar uma nova instância do Amazon EC2 para Ubuntu Server ou RHEL.	Consulte Crie uma instância do Amazon EC2 para (CodeDeploy AWS CLI ou console Amazon EC2).
Quero configurar uma instância do Amazon EC2 para Amazon Linux, Windows Server, Ubuntu Server ou RHEL.	Consulte Configurar uma instância do Amazon EC2 para trabalhar com CodeDeploy.
Quero configurar uma instância on-premises para Windows Server, Ubuntu Server ou RHEL (dispositivos físicos que não são instâncias do Amazon EC2).	Consulte Working with On-Premises Instances.
Quero CodeDeploy provisionar uma frota substituta de instâncias durante uma blue/green implantação.	Consulte Trabalhando com implantações em CodeDeploy.

Para preparar instâncias do Amazon EC2 em grupos do Amazon EC2 Auto Scaling, você deve seguir algumas etapas adicionais. Para obter mais informações, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

Tópicos

- Tagging Instances for Deployments
- Working with Amazon EC2 Instances
- Working with On-Premises Instances
- View Instance Details
- Instance Health

Marcar instâncias para grupos de implantação em CodeDeploy

Para ajudar a gerenciar as instâncias do Amazon EC2 e instâncias on-premises, você pode usar tags para atribuir seus próprios metadados a cada recurso. As tags permitem categorizar suas instâncias de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando você

tem várias instâncias. Você pode identificar rapidamente uma instância ou grupo de instâncias com base nas tags que você tiver atribuído a elas. Cada tag consiste de uma chave e um valor opcional, que podem ser definidos. Para obter mais informações, consulte Marcar recursos do Amazon EC2.

Para especificar quais instâncias estão incluídas em um grupo de CodeDeploy implantação, você especifica tags em um ou mais grupos de tags. As instâncias que atendem aos critérios de tag são aquelas nas quais a revisão de aplicativo mais recente esteja instalada quando uma implantação para o grupo de implementação for criado

> **Note**
>
> Você também pode incluir grupos do Amazon EC2 Auto Scaling em grupos de implantação, mas eles são identificados por seus nomes em vez de tags aplicadas a instâncias. Para mais informações, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

Os critérios para instâncias em um grupo de implementação podem ser tão simples quanto uma única tag em um único grupo de tags. Pode ser tão complexo como 10 tags, cada uma em um máximo de três grupos de tags.

Se você usar um grupo de tag única, qualquer instância identificada por pelo menos uma tag no grupo é incluída no grupo de implementação. Se você usar grupos de várias tags, apenas as instâncias que são identificadas por pelo menos uma tag em cada um dos grupos de tags serão incluídas.

Os exemplos a seguir ilustram como tags e grupos de tags podem ser usados para selecionar as instâncias de um grupo de implantação.

Tópicos

- Exemplo 1: Grupo de tag única, tag única
- Exemplo 2: Grupo de tag única, várias tags
- Exemplo 3: Grupos de várias tags, tags únicas
- Exemplo 4: Grupos de várias tags, várias tags

Exemplo 1: Grupo de tag única, tag única

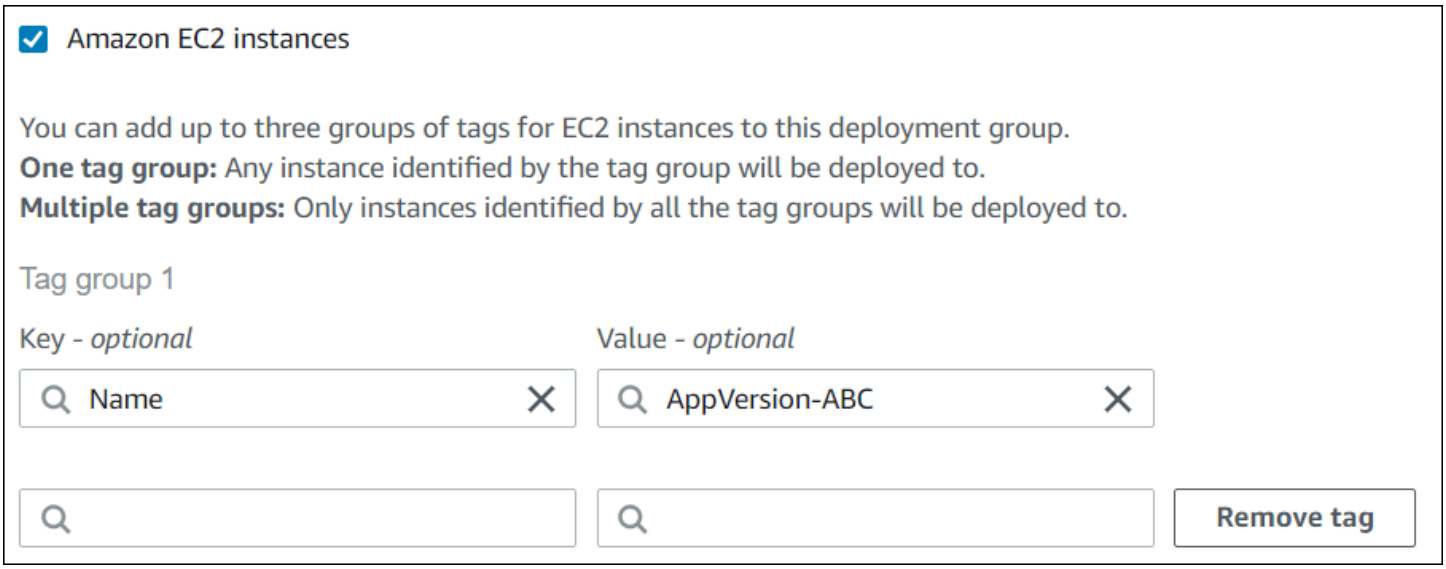
Você pode especificar uma tag única em um grupo de tag única:

Grupo de tags 1

Chave	Valor
Nome	AppVersion-ABC

Cada instância marcada com `Name=AppVersion-ABC` faz parte do grupo de implantação, mesmo se tiver outras tags aplicadas.

CodeDeploy visualização da configuração do console:



Estrutura do JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Name",
                  "Value": "AppVersion-ABC"
               }
            ]
         ]
      },
```

Exemplo 2: Grupo de tag única, várias tags

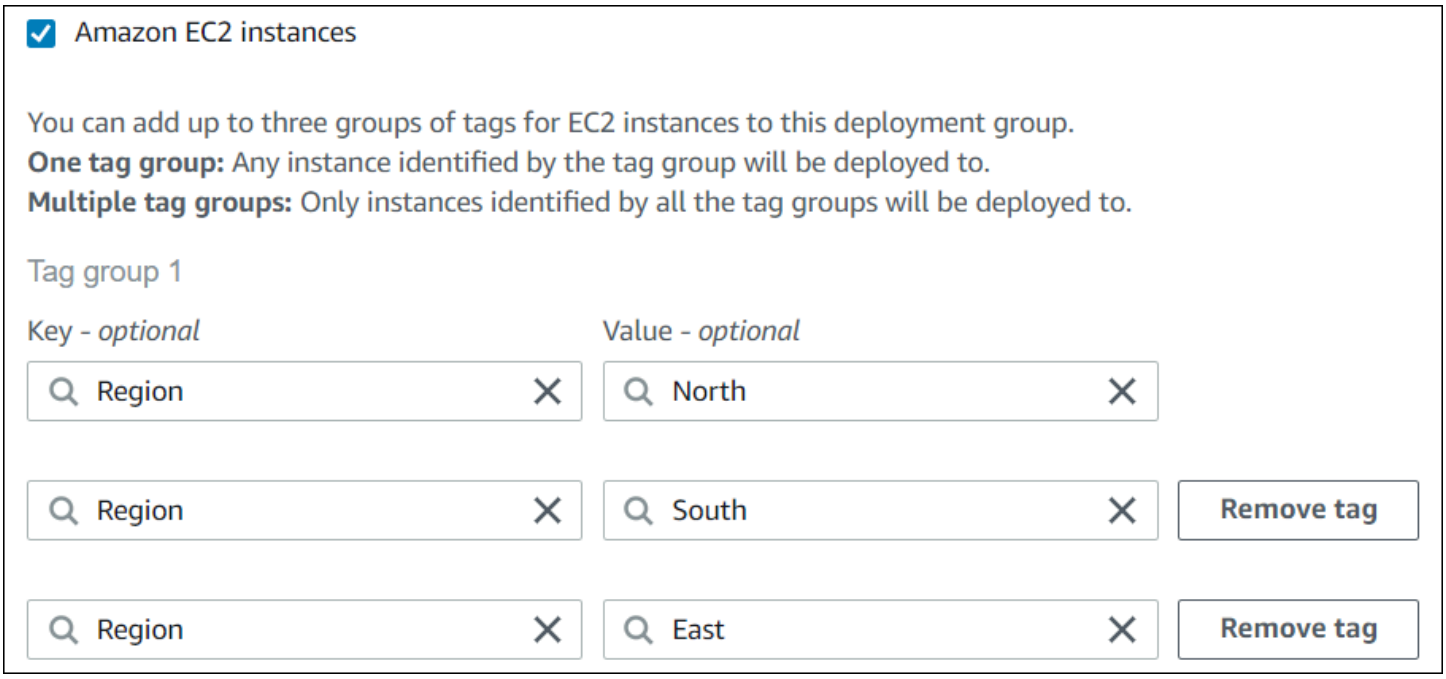
Você também pode especificar várias tags em um grupo de tag única:

Grupo de tags 1

Chave	Valor
Região	Norte
Região	Sul
Região	Leste

Uma instância marcada com qualquer dessas três tags faz parte do grupo de implantação, mesmo se tiver outras tags aplicadas. Se, por exemplo, você tiver outras instâncias com `Region=West`, elas não seriam incluídas no grupo de implantação.

CodeDeploy visualização da configuração do console:



Estrutura do JSON:

```
"ec2TagSet": {
        "ec2TagSetList": [
```

```
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               },
                           {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "South"
               },
                           {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "East"
               }
            ]
         ]
      },
```

Exemplo 3: Grupos de várias tags, tags únicas

Você também pode usar vários conjuntos de grupos de tags com um par de chave/valor único em cada para especificar os critérios para instâncias em um grupo de implantação. Quando você usa grupos de várias tags em um grupo de implantação, apenas as instâncias que são identificadas por todos os grupos de tags são incluídas no grupo de implantação.

Grupo de tags 1

Chave	Valor
Nome	AppVersion-ABC

Grupo de tags 2

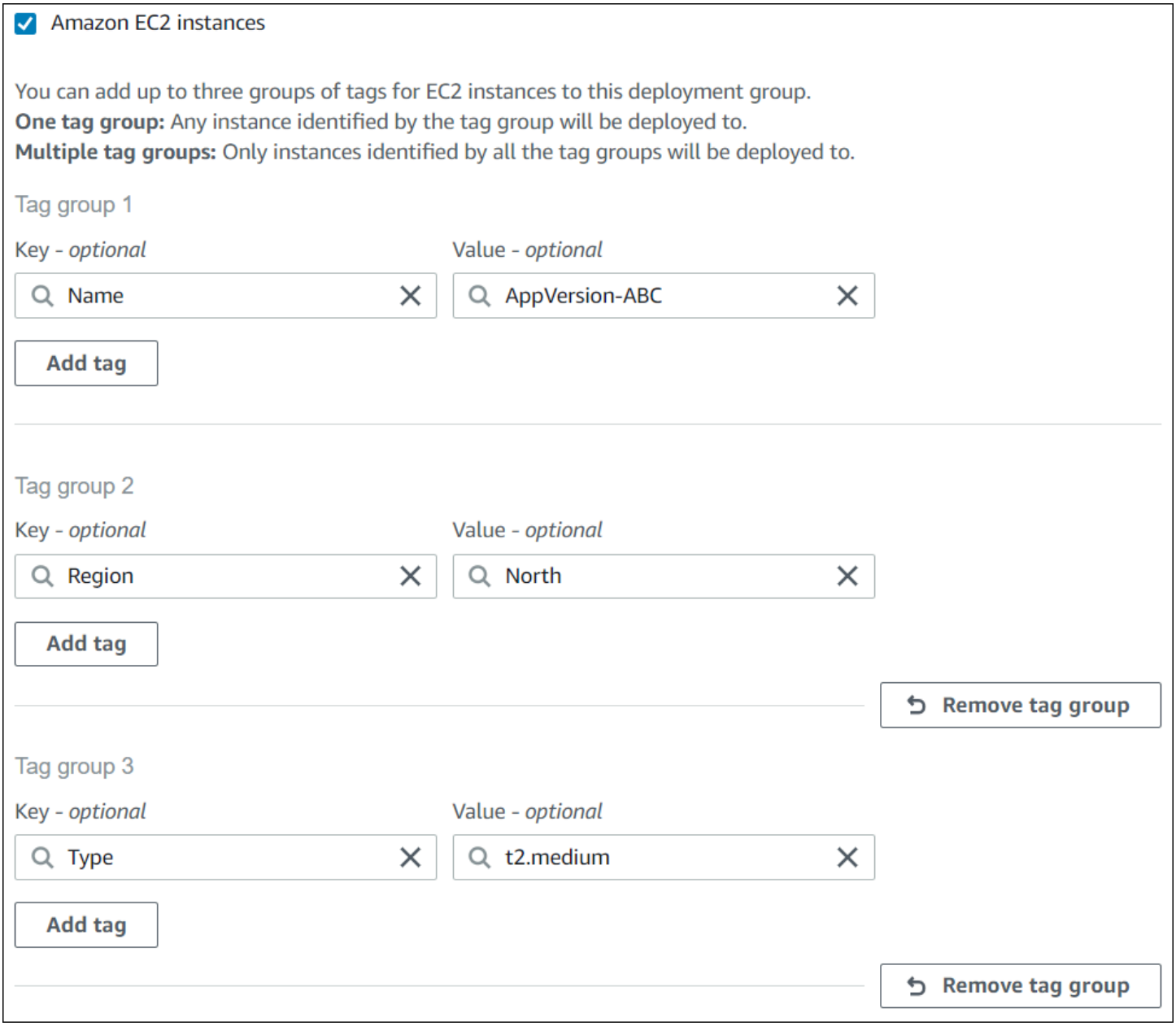
Chave	Valor
Região	Norte

Grupo de tags 3

Chave	Valor
Tipo	t2.medium

Você pode ter instâncias em várias regiões e de vários tipos de instância marcadas com`Name=AppVersion-ABC`. Neste exemplo, somente as instâncias também marcadas com `Region=North` e `Type=t2.medium` fazem parte do grupo de implantação.

CodeDeploy visualização da configuração do console:



Estrutura do JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Name",
                  "Value": "AppVersion-ABC"
               }
            ],
```

```
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.medium"
               }
            ],
         ]
      },
```

Exemplo 4: Grupos de várias tags, várias tags

Quando você usa vários grupos de tags com várias tags em um ou mais grupos, uma instância deve corresponder a pelo menos uma das tags em cada um dos grupos.

Grupo de tags 1

Chave	Valor
Environment	Beta
Environment	Preparação

Grupo de tags 2

Chave	Valor
Região	Norte
Região	Sul
Região	Leste

Grupo de tags 3

Chave	Valor
Tipo	t2.medium
Tipo	t2.large

Neste exemplo, uma instância a ser incluída no grupo de implantação deve ser marcada com (1) `Environment=Beta` ou `Environment=Staging`, com (2) `Region=North`, `Region=Southou` `Region=East` e com (3) `Type=t2.medium` ou `Type=t2.large`.

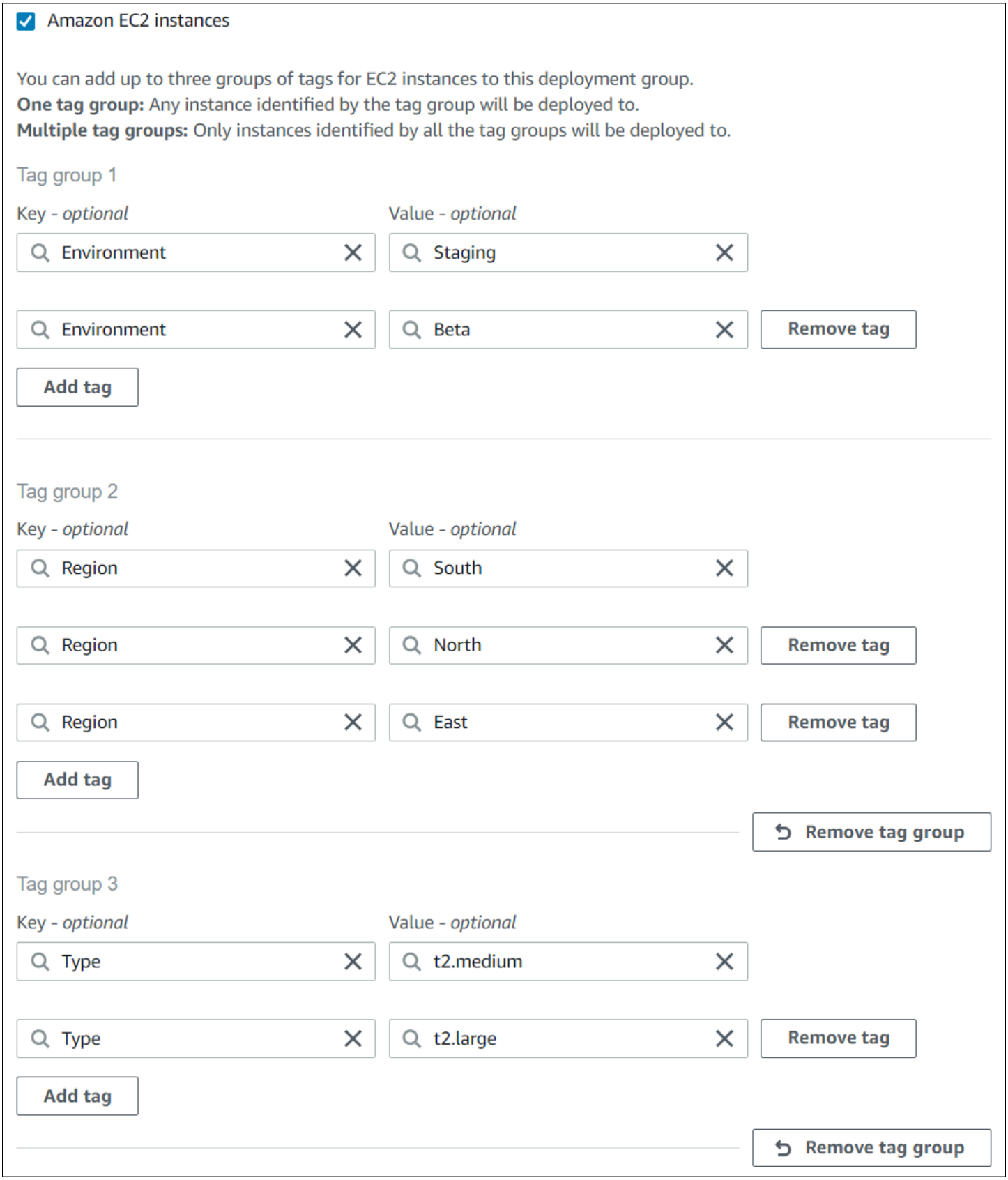
Para ilustrar, instâncias com os seguintes grupos de tags estariam entre as inclusas no grupo de implantação:

- `Environment=Beta`, `Region=North`,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,`Type=t2.large`
- `Environment=Staging`,`Region=South`,`Type=t2.large`

Instâncias com os seguintes grupos de tags não estariam inclusas no grupo de implantação: Os valores de chave destacados fazem com que as instâncias sejam excluídas:

- `Environment=Beta`, Region=West,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,Type=t2.micro
- Environment=Production,`Region=South`,`Type=t2.large`

CodeDeploy visualização da configuração do console:


Amazon EC2 instances
You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.
Tag group 1
Key - optional
Value - optional
Environment
Staging
Environment
Beta
Remove tag
Add tag
Tag group 2
Key - optional
Value - optional
Region
South
Region
North
Remove tag
Region
East
Remove tag
Add tag
Remove tag group
Tag group 3
Key - optional
Value - optional
Type
t2.medium
Type
t2.large
Remove tag
Add tag
Remove tag group

Estrutura do JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Environment",
                  "Value": "Beta"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Environment",
                  "Value": "Staging"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "North"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "South"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Region",
                  "Value": "East"
               }
            ],
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.medium"
               },
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Type",
                  "Value": "t2.large"
               }
```

```
            ],
        ]
    },
```

Trabalhando com instâncias do Amazon EC2 para CodeDeploy

Uma instância do Amazon EC2 é um ambiente de computação virtual que você cria e configura usando o Amazon Elastic Compute Cloud. O Amazon EC2 fornece capacidade de computação escalável na nuvem. AWS Você pode usar o Amazon EC2 para lançar quantos servidores virtuais forem necessários para suas CodeDeploy implantações.

Para obter mais informações sobre o Amazon EC2, consulte o Guia de conceitos básicos do Amazon EC2.

As instruções nesta seção mostram como criar e configurar instâncias do Amazon EC2 para uso em suas CodeDeploy implantações.

Tópicos

- Crie uma instância do Amazon EC2 para (CodeDeploy AWS CLI ou console Amazon EC2)
- Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo)
- Configurar uma instância do Amazon EC2 para trabalhar com CodeDeploy

Crie uma instância do Amazon EC2 para (CodeDeploy AWS CLI ou console Amazon EC2)

Essas instruções mostram como iniciar uma nova instância do Amazon EC2 que está configurada para uso em CodeDeploy implantações.

Você pode usar nosso CloudFormation modelo para iniciar uma instância do Amazon EC2 executando o Amazon Linux ou o Windows Server que já esteja configurada para uso em CodeDeploy implantações. Não fornecemos um CloudFormation modelo para instâncias do Amazon EC2 executando o Ubuntu Server ou o Red Hat Enterprise Linux (RHEL). Para alternativas ao uso do template, consulte Trabalhando com instâncias para CodeDeploy.

Você pode usar o console do Amazon EC2 ou as APIs do Amazon EC2 para iniciar uma instância do Amazon EC2. AWS CLI

Iniciar uma instância do Amazon EC2 (console)

Pré-requisitos

Se você ainda não tiver feito isso, siga as instruções Começando com CodeDeploy para configurar AWS CLI e criar um perfil de instância do IAM.

Iniciar uma instância do Amazon EC2

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação, escolha Instances (Instâncias) e Launch Instance (Iniciar instância).
3. Na página Step 1: Choose an Amazon Machine Image (AMI) (Etapa 1: escolher uma imagem de máquina da Amazon (AMI)), na guia Quick Start (Início rápido), localize o sistema operacional e a versão que você deseja usar, e escolha Select (Selecionar). Você deve escolher um sistema operacional Amazon EC2 AMI suportado pela. CodeDeploy Para obter mais informações, consulte Sistemas operacionais suportados pelo CodeDeploy agente.
4. Na página Etapa 2: Escolher um tipo de instância, escolha qualquer tipo de instância do Amazon EC2 disponível e, em seguida, Próximo: Configurar detalhes da instância.
5. Na página Etapa 3: Configurar detalhes da instância, na lista Perfil do IAM, escolha o perfil da instância do IAM criada em Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2. Se você usou o nome de função sugerido, escolha **`CodeDeployDemo-EC2-Instance-Profile`**. Caso tenha criado seu próprio nome de função, selecione-o.

   > **Note**
   >
   > Se uma nuvem privada virtual (VPC) padrão não for exibida na lista Rede, você deverá escolher ou criar uma Amazon VPC e uma sub-rede. Selecione Create new VPC (Criar nova VPC) ou Create new subnet (Criar nova sub-rede) ou ambos. Para obter mais informações, consulte Sua VPC e sub-redes.

6. Escolha Next: Add Storage (Próximo: adicionar armazenamento).
7. Deixe a página Step 4: Add Storage (Etapa 4: adicionar armazenamento) inalterada e escolha Next: Add Tags (Próximo: adicionar tags).
8. Na página Step 5: Add Tags (Etapa 5: Adicionar tags), escolha Add Tag (Adicionar tag).
9. Na caixa Key (Chave), digite **`Name`**. Na caixa Value (Valor), digite **`CodeDeployDemo`**.

> **Important**
>
> O conteúdo das caixas Key (Chave) e Value (Valor) faz distinção entre maiúsculas e minúsculas.

10. Escolha Próximo: configurar grupo de segurança.
11. Na página Step 6: Configure Security Group (Etapa 6: configurar o grupo de segurança), deixe a opção Create a new security group (Criar novo grupo de segurança) selecionada.

    Uma função do SSH padrão é configurada para instâncias do Amazon EC2 que executem o Amazon Linux, o Ubuntu Server ou o RHEL. Uma função do RDP padrão é configurada para instâncias do Amazon EC2 que executem o Windows Server.
12. Para abrir a porta HTTP, escolha o botão Add Rule (Adicionar regra) e, na lista suspensa Type (Tipo), escolha **HTTP**. Aceite o valor de origem padrão de Custom 0.0.0. 0/0e, em seguida, escolha Revisar e iniciar.

    > **Note**
    >
    > Em um ambiente de produção, recomendamos restringir o acesso às portas SSH, RDP e HTTP, em vez de especificar Anywhere 0.0.0. 0/0. CodeDeploy não exige acesso irrestrito à porta e não requer acesso HTTP. Para obter mais informações, consulte Dicas para proteger sua instância do Amazon EC2.

    Se uma caixa de diálogo Boot from General Purpose (SSD) (Inicialização de uso geral (SSD)) for exibida, siga as instruções e escolha Next (Próximo).
13. Deixe a página Step 7: Review Instance Launch (Etapa 7: revisar o lançamento da instância) inalterada e escolha Launch (Iniciar).
14. Na caixa de diálogo Select an existing key pair or create a new key pair (Selecionar um par de chaves existente ou criar um novo par de chaves), escolha Choose an existing key pair (Selecionar um par de chaves existente) ou Create a new key pair (Criar um novo par de chaves). Se já tiver configurado um par de chaves de instância do Amazon EC2, você pode escolhê-lo aqui.

    Caso você ainda não tenha um par de chaves da instância do Amazon EC2, escolha Create a new key pair (Criar um novo par de chaves) e atribua a ele um nome reconhecível. Escolha

Download do par de chaves para fazer download do par de chaves da instância do Amazon EC2 no computador.

> **Important**
>
> Você deve ter um par de chaves se deseja acessar sua instância do Amazon EC2 com SSH ou RDP.

15. Selecione Launch Instances.
16. Escolha o ID da sua instância do Amazon EC2. Não continue até que a instância tenha sido iniciada e aprovada em todas as verificações.

Instale o CodeDeploy agente

O CodeDeploy agente deve ser instalado na sua instância do Amazon EC2 antes de ser usado em CodeDeploy implantações. Para obter mais informações, consulte Instale o CodeDeploy agente.

> **Note**
>
> Você pode configurar a instalação e as atualizações automáticas do CodeDeploy agente ao criar seu grupo de implantação no console.

Iniciar uma instância do Amazon EC2 (CLI)

Pré-requisitos

Se você ainda não tiver feito isso, siga as instruções Começando com CodeDeploy para configurar AWS CLI e criar um perfil de instância do IAM.

Iniciar uma instância do Amazon EC2

1. Apenas para Windows Server Caso você esteja criando uma instância do Amazon EC2 executando o Windows Server, chame os comandos create-security-group e authorize-security-group-ingress para criar um grupo de segurança que permita acesso RDP (não permitido por padrão) e, como alternativa, acesso HTTP. Por exemplo, para criar um grupo de segurança chamado CodeDeployDemo-Windows-Security-Group, execute os seguintes comandos, um de cada vez:

```
aws ec2 create-security-group --group-name CodeDeployDemo-Windows-Security-Group --
description "For launching Windows Server images for use with CodeDeploy"
```

```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-
Security-Group --to-port 3389 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port
 3389
```

```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-
Security-Group --to-port 80 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port 80
```

> **Note**
>
> Para fins de demonstração, esses comandos criam um grupo de segurança que permite acesso irrestrito para o RDP por meio da porta 3389 e, alternativamente, para o HTTP por meio da porta 80. Como melhor prática, recomendamos restringir o acesso às portas RDP e HTTP. O CodeDeploy não exige acesso irrestrito a portas nem acesso HTTP. Para obter mais informações, consulte Dicas para proteger sua instância do Amazon EC2.

2. Chame o comando run-instances para criar e iniciar a instância do Amazon EC2.

   Antes de chamar esse comando, você precisará coletar o seguinte:

   - O ID de uma Amazon Machine Image (AMI) (*`ami-id`*) que você usa para a instância. Para obter o ID, consulte Encontrar uma AMI adequada.
   - O nome do tipo de instância do Amazon EC2 (*`instance-type`*) que você cria, como. `t1.micro` Para ver uma lista, consulte Tipos de instâncias do Amazon EC2.
   - O nome de um perfil de instância do IAM com permissão para acessar o bucket do Amazon S3 onde os arquivos de instalação do CodeDeploy agente para sua região estão armazenados.

     Para obter informações sobre como criar um perfil de instância do IAM, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.
   - O nome de um par de chaves de instância do Amazon EC2 (*`key-name`*) para permitir o acesso SSH a uma instância do Amazon EC2 executando Amazon Linux, Ubuntu Server ou acesso RHEL ou RDP a uma instância do Amazon EC2 executando o Windows Server.

> **Important**
>
> Digite apenas o nome do par de chaves, e não sua extensão de arquivo. Por exemplo, my-keypair, e não my-keypair.pem.

Para encontrar um nome de par de chaves, abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2 No painel de navegação, em Rede e segurança, escolha Pares de chaves e anote o nome do par de chaves na lista.

Para gerar um par de chaves, consulte Criando seu par de chaves com o Amazon EC2. Certifique-se de criar o par de chaves em uma das regiões listadas em Região e nos endpoints em Referência geral da AWS. Caso contrário, você não poderá usar o par de chaves de instância do Amazon EC2 com. CodeDeploy

Para Amazon Linux, RHEL e Ubuntu Server

Para chamar o comando run-instances para iniciar uma instância do Amazon EC2 que executa o Amazon Linux, Ubuntu Server ou RHEL e anexar o perfil de instância do IAM criado em Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2. Por exemplo:

```
aws ec2 run-instances \
  --image-id ami-id \
  --key-name key-name \
  --count 1 \
  --instance-type instance-type \
  --iam-instance-profile Name=iam-instance-profile
```

> **Note**
>
> Esse comando cria um grupo de segurança padrão para a instância do Amazon EC2 que permite acesso a várias portas, incluindo acesso irrestrito para SSH por meio da porta 22 e, alternativamente, acesso para HTTP por meio da porta 80. Como prática recomendada, recomendamos restringir o acesso somente às portas SSH e HTTP. CodeDeploy não exige acesso irrestrito à porta e não exige acesso à porta HTTP. Para obter mais informações, consulte Dicas para proteger sua instância do Amazon EC2.

Para Windows Server

Para chamar o comando run-instances a fim de iniciar uma instância do Amazon EC2 que execute o Windows Server e anexar o perfil de instância do IAM criado em Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2 e especificar o nome do grupo de segurança que você criou na Etapa 1. Por exemplo:

```
aws ec2 run-instances --image-id ami-id --key-name key-name --count 1 --instance-
type instance-type --iam-instance-profile Name=iam-instance-profile --security-
groups CodeDeploy-Windows-Security-Group
```

Esses comandos iniciam uma única instância do Amazon EC2 com a AMI, o par de chaves e o tipo de instância especificados, com o perfil de instância do IAM especificado, e executam o script especificado durante a inicialização.

3. Observe o valor de `InstanceID` na saída. Se você esquecer esse valor, poderá obtê-lo mais tarde chamando o comando describe-instances para o par de chaves de instância do Amazon EC2.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query
 "Reservations[*].Instances[*].[InstanceId]" --output text
```

Use o ID da instância para chamar o create-tags comando, que marca a instância do Amazon EC2 para que CodeDeploy você possa encontrá-la posteriormente durante uma implantação. No exemplo a seguir, a tag se chama **`CodeDeployDemo`**, mas você pode especificar qualquer tag de instância do Amazon EC2 que desejar.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

Você pode aplicar várias tags a uma instância ao mesmo tempo. Por exemplo:

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance
 Key=Region,Value=West Key=Environment,Value=Beta
```

Para verificar se a instância do Amazon EC2 foi iniciada e aprovada em todas as verificações, use o ID de instância para chamar o comando describe-instance-status.

```
aws ec2 describe-instance-status --instance-ids instance-id --query
 "InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Se a instância tiver sido executada e aprovada em todas as verificações, ok será exibido na saída.

Instale o CodeDeploy agente

O CodeDeploy agente deve ser instalado na sua instância do Amazon EC2 antes de ser usado em CodeDeploy implantações. Para obter mais informações, consulte Instale o CodeDeploy agente.

> **Note**
>
> Você pode configurar a instalação e as atualizações automáticas do CodeDeploy agente ao criar seu grupo de implantação no console.

Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo)

Você pode usar nosso CloudFormation modelo para iniciar rapidamente uma instância do Amazon EC2 executando o Amazon Linux ou o Windows Server. Você pode usar o AWS CLI, o CodeDeploy console ou as AWS APIs para iniciar a instância com o modelo. Além de iniciar a instância, o template faz o seguinte:

- Instrui CloudFormation a conceder permissão à instância para participar das CodeDeploy implantações.
- Marca a instância para que CodeDeploy possa encontrá-la durante uma implantação.
- Instala e executa o CodeDeploy agente na instância.

Você não precisa usar o nosso CloudFormation para configurar uma instância do Amazon EC2. Para conhecer alternativas, consulte Trabalhando com instâncias para CodeDeploy.

Não fornecemos um CloudFormation modelo para instâncias do Amazon EC2 executando o Ubuntu Server ou o Red Hat Enterprise Linux (RHEL).

Tópicos

- Antes de começar

- Execute uma instância do Amazon EC2 com o CloudFormation modelo (console)
- Execute uma instância do Amazon EC2 com o CloudFormation modelo (AWS CLI)

Antes de começar

Antes de usar o CloudFormation modelo para iniciar instâncias do Amazon EC2, certifique-se de concluir as etapas a seguir.

1. Certifique-se de ter criado um usuário administrativo, conforme descrito emEtapa 1: configuração. Double-check que o usuário tenha as seguintes permissões mínimas e adicione as que não estão presentes:

   - cloudformation:*
   - codedeploy:*
   - ec2:*
   - objetivo: AddRoleToInstanceProfile
   - objetivo: CreateInstanceProfile
   - objetivo: CreateRole
   - objetivo: DeleteInstanceProfile
   - objetivo: DeleteRole
   - objetivo: DeleteRolePolicy
   - objetivo: GetRole
   - objetivo: DeleteRolePolicy
   - objetivo: PutRolePolicy
   - objetivo: RemoveRoleFromInstanceProfile
2. Certifique-se de ter um par de chaves de instância para permitir o acesso SSH à instância do Amazon EC2 executando o Amazon Linux ou acesso RDP à instância que executa o Windows Server.

   Para encontrar um nome de par de chaves, abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2 No painel de navegação, em Rede e segurança, escolha Pares de chaves e anote o nome do par de chaves na lista.

   Para gerar um novo par de chaves, consulte Criando seu par de chaves com o Amazon EC2. Certifique-se de que o par de chaves tenha sido criado em uma das regiões listadas em Região

e nos endpoints em Referência geral da AWS. Caso contrário, você não poderá usar o par de chaves de instância com o CodeDeploy.

Execute uma instância do Amazon EC2 com o CloudFormation modelo (console)

1. Faça login no Console de gerenciamento da AWS e abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.

   > **Important**
   >
   > Faça login no Console de gerenciamento da AWS com a mesma conta que você usouComeçando com CodeDeploy. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e os endpoints em. Referência geral da AWS CodeDeploy suporta somente essas regiões.

2. Selecione Create Stack (Criar pilha).
3. Em Escolher um modelo, selecione Especificar um URL do modelo do Amazon S3. Na caixa, digite a localização do CloudFormation modelo para sua região e escolha Avançar.

Região	Localização do CloudFormation modelo
Região Leste dos EUA (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Leste dos EUA (Norte da Virgínia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Oeste dos EUA (Norte da Califórnia)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região Oeste dos EUA (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Canadá (Central)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Londres)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região de Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Hong Kong)	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Tóquio)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Seul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região Ásia-Pacífico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região América do Sul (São Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

4. Na caixa Stack name (Nome da pilha), digite um nome para a pilha (por exemplo, **`CodeDeployDemoStack`**).
5. Em Parâmetros, digite o seguinte e escolha Avançar.

   - Para InstanceCount, digite o número de instâncias que você deseja iniciar. (Recomendamos deixar o padrão de 1.)
   - Para InstanceType, digite o tipo de instância que você deseja iniciar (ou deixe o padrão de t1.micro).
   - Para KeyPairName, digite o nome do par de chaves da instância. Digite apenas o nome do par de chaves, e não sua extensão de arquivo.
   - Em OperatingSystembox, digite **`Windows`** para iniciar instâncias executando o Windows Server (ou deixe o padrão do Linux).
   - Em SSHLocation, digite o intervalo de endereços IP a ser usado para se conectar à instância com SSH ou RDP (ou deixe o padrão de 0.0.0). 0/0).

> **Important**
>
> O padrão de **`0.0.0.0/0`** é fornecido apenas para fins de demonstração. CodeDeploy não exige que as instâncias do Amazon EC2 tenham acesso irrestrito às portas. Como prática recomendada, recomendamos restringir o acesso às portas SSH (e HTTP). Para obter mais informações, consulte Dicas para proteger sua instância do Amazon EC2.

- Para TagKey, digite a tag de instância que a chave CodeDeploy usará para identificar as instâncias durante a implantação (ou deixe o padrão de Nome).
- Para TagValue, digite o valor da tag de instância CodeDeploy que será usado para identificar as instâncias durante a implantação (ou deixe o padrão de CodeDeployDemo).

6. Na página Opções, deixe as caixas de opções em branco e escolha Avançar.

> **Important**
>
> CloudFormation as tags são diferentes das CodeDeploy tags. CloudFormation usa tags para simplificar a administração da sua infraestrutura. CodeDeploy usa tags para identificar instâncias do Amazon EC2. Você especificou tags do CodeDeploy na página Specify Parameters (Especificar parâmetros).

7. Na página de revisão, em Capacidades, selecione a caixa Eu reconheço que CloudFormation podem criar recursos do IAM e, em seguida, escolha Criar.

   Depois de criar CloudFormation a pilha e iniciar as instâncias do Amazon EC2, no CloudFormation console, CREATE_COMPLETE será exibido na coluna Status. Esse processo pode levar alguns minutos.

Para verificar se o CodeDeploy agente está sendo executado nas instâncias do Amazon EC2Gerenciando as operações CodeDeploy do agente, consulte e prossiga para. Crie um aplicativo com CodeDeploy

Execute uma instância do Amazon EC2 com o CloudFormation modelo (AWS CLI)

1. Use nosso CloudFormation modelo em uma chamada para o create-stack comando. Essa pilha iniciará uma nova instância do Amazon EC2 com CodeDeploy o agente instalado.

Para iniciar uma instância do Amazon EC2 executando o Amazon Linux:

```
aws cloudformation create-stack \
  --stack-name CodeDeployDemoStack \
  --template-url templateURL \
  --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro \
    ParameterKey=KeyPairName,ParameterValue=keyName
 ParameterKey=OperatingSystem,ParameterValue=Linux \
    ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name \
    ParameterKey=TagValue,ParameterValue=CodeDeployDemo \
  --capabilities CAPABILITY_IAM
```

Para iniciar uma instância do Amazon EC2 executando o Windows Server:

```
aws cloudformation create-stack --stack-name CodeDeployDemoStack --template-
url template-url --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro
 ParameterKey=KeyPairName,ParameterValue=keyName
 ParameterKey=OperatingSystem,ParameterValue=Windows
 ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name
 ParameterKey=TagValue,ParameterValue=CodeDeployDemo --capabilities CAPABILITY_IAM
```

*keyName*é o nome do par de chaves da instância. Digite apenas o nome do par de chaves, e não sua extensão de arquivo.

*template-url*é a localização do CloudFormation modelo para sua região:

Região	Localização do CloudFormation modelo
Região Leste dos EUA (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região Leste dos EUA (Norte da Virgínia)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Oeste dos EUA (Norte da Califórnia)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Oeste dos EUA (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Canadá (Central)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Irlanda)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Londres)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Europa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região Europa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região de Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Hong Kong)	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Tóquio)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Seul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Região	Localização do CloudFormation modelo
Região Ásia-Pacífico (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Região Ásia-Pacífico (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Região América do Sul (São Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

Esse comando cria uma CloudFormation pilha chamada**`CodeDeployDemoStack`**, usando o CloudFormation modelo no bucket do Amazon S3 especificado. A instância do Amazon EC2 se baseia no tipo de instância t1.micro, mas você pode usar qualquer tipo. Ela é marcada com o valor **`CodeDeployDemo`**, mas você pode marcá-la com qualquer valor. Ela tem o par de chaves de instância especificado aplicado.

2. Chame o describe-stacks comando para verificar se a CloudFormation pilha chamada **`CodeDeployDemoStack`** foi criada com sucesso:

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query
 "Stacks[0].StackStatus" --output text
```

Não prossiga até que o valor `CREATE_COMPLETE` seja retornado.

Para verificar se o CodeDeploy agente está sendo executado na instância do Amazon EC2Gerenciando as operações CodeDeploy do agente, consulte e prossiga para. Crie um aplicativo com CodeDeploy

Configurar uma instância do Amazon EC2 para trabalhar com CodeDeploy

Essas instruções mostram como configurar uma instância do Amazon EC2 executando Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) ou Windows Server para uso em implantações. CodeDeploy

> **Note**
>
> Se você não tiver uma instância do Amazon EC2, poderá usar o CloudFormation modelo para iniciar uma executando o Amazon Linux ou o Windows Server. Não fornecemos um modelo para o Ubuntu Server ou o RHEL.

Etapa 1: Verificar se um perfil de instância do IAM está anexado à sua instância do EC2

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação, em Instâncias, escolha Instâncias.
3. Navegue e escolha sua instância do Amazon EC2 na lista.
4. No painel de detalhes, na guia Descrição, anote o valor contido no campo Perfil do IAM e, então, prossiga para a próxima seção.

   Se o campo estiver vazio, você poderá anexar um perfil de instância do IAM à instância. Para obter informações, consulte Anexar um perfil do IAM a uma instância.

Etapa 2: Verificar se o perfil da instância do IAM tem as permissões de acesso corretas

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação, escolha Perfis.
3. Navegue e escolha o nome do perfil do IAM anotado na etapa 4 da seção anterior.

> **Note**
>
> Se você quiser usar a função de serviço gerada pelo CloudFormation modelo em vez da que você criou seguindo as instruções emEtapa 2: criar uma função de serviço para CodeDeploy, observe o seguinte:
> Em algumas versões do nosso CloudFormation modelo, o nome de exibição do perfil de instância do IAM gerado e anexado às instâncias do Amazon EC2 não é o mesmo que o nome de exibição no console do IAM. Por exemplo, o perfil de instância do IAM pode ter um nome para exibição de `CodeDeploySampleStack-expnyi6-InstanceRoleInstanceProfile-IK8J8A9123EX`, enquanto o perfil de instância do IAM no console do IAM pode ter um nome para exibição de `CodeDeploySampleStack-expnyi6-InstanceRole-C5P33V1L64EX`.
> Para ajudá-lo a identificar o perfil de instância no console do IAM, você verá que o prefixo de `CodeDeploySampleStack-expnyi6-InstanceRole` é o mesmo para ambos. Para obter informações sobre por que esses nomes para exibição podem ser diferentes, consulte Perfis de instância.

4. Selecione a guia Relacionamentos de confiança. Se não houver uma entrada em Entidades de confiança em que esteja escrito Provedor(es) de identidade ec2.amazonaws.com, não será possível usar essa instância do Amazon EC2. Pare e crie uma instância do Amazon EC2 usando as informações em Trabalhando com instâncias para CodeDeploy.

   Se houver uma entrada que diz O (s) provedor (es) de identidade ec2.amazonaws.com e você estiver armazenando seus aplicativos somente em GitHub repositórios, vá para. Etapa 3: Marcar a instância do Amazon EC2

   Se houver uma entrada indicando Provedor(es) de identidade ec2.amazonaws.com e você estiver armazenando suas aplicações somente em buckets do Amazon S3, escolha a guia Permissões.
5. Se houver uma política na área Permissions policies (Políticas de permissões), expanda a política e selecione Edit policy (Editar política).
6. Selecione a guia JSON. Se você estiver armazenando suas aplicações em buckets do Amazon S3, certifique-se de que `"s3:Get*"` e `"s3:List*"` estejam na lista de ações especificadas.

Essa lista pode ser semelhante a:

```
{"Statement":[{"Resource":"*","Action":[
  ... Some actions may already be listed here ...
  "s3:Get*","s3:List*"
  ... Some more actions may already be listed here ...
  ],"Effect":"Allow"}]}
```

Ou ser algo neste estilo:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Se `"s3:Get*"` e `"s3:List*"` não estiverem na lista de ações especificadas, escolha Editar para adicioná-las e escolha Salvar. (Se nem `"s3:Get*"` nem `"s3:List*"` for a última ação na lista, certifique-se de adicionar uma vírgula após a ação, para que o documento de política seja validado.)

> **Note**
>
> Recomendamos que você restrinja essa política somente aos buckets do Amazon S3 que suas instâncias do Amazon EC2 precisam acessar. Certifique-se de dar acesso aos buckets do Amazon S3 que contêm o agente. CodeDeploy Caso contrário, poderá ocorrer um erro quando o CodeDeploy agente for instalado ou atualizado nas instâncias. Para conceder ao perfil da instância do IAM acesso somente a alguns buckets do kit de

CodeDeploy recursos no Amazon S3, use a política a seguir, mas remova as linhas dos buckets aos quais você deseja impedir o acesso:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
        "arn:aws:s3:::aws-codedeploy-me-central-1/*",
        "arn:aws:s3:::aws-codedeploy-me-south-1/*",
        "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
      ]
    }
```

```
  ]
}
```

Etapa 3: Marcar a instância do Amazon EC2

Para obter instruções sobre como marcar a instância do Amazon EC2 para que ela CodeDeploy possa ser encontrada durante uma implantação, consulte Como trabalhar com tags no console e, em seguida, retorne a esta página.

> **Note**
>
> Você pode marcar a instância do Amazon EC2 com qualquer chave e valor desejado. Certifique-se de especificar essa chave e esse valor quando fizer a implantação.

Etapa 4: instalar o AWS CodeDeploy agente na instância do Amazon EC2

Para obter instruções sobre como instalar o CodeDeploy agente na instância do Amazon EC2 e verificar se ele está em execuçãoGerenciando as operações CodeDeploy do agente, consulte e prossiga para. Crie um aplicativo com CodeDeploy

Trabalhando com instâncias locais para CodeDeploy

Uma instância local é qualquer dispositivo físico que não seja uma instância do Amazon EC2 que possa executar CodeDeploy o agente e se conectar a endpoints de serviço AWS público.

A implantação de uma revisão do CodeDeploy aplicativo em uma instância local envolve duas etapas principais:

- Etapa 1 — Configure cada instância local, registre-a e CodeDeploy, em seguida, marque-a.
- Etapa 2: implantar revisões de aplicações na instância on-premises.

> **Note**
>
> Para fazer testes com a criação e implantação de uma revisão de aplicativo de amostra em uma instância local corretamente configurada e registrada, consulte Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server

ou Red Hat Enterprise Linux). Para obter informações sobre instâncias locais e como elas funcionam CodeDeploy, consulteWorking with On-Premises Instances.

Se não quiser mais que uma instância on-premises seja usada nas implantações, basta remover as tags de instâncias on-premises dos grupos de implantação. Para uma abordagem mais robusta, remova as tags de instâncias locais da instância. Você também pode cancelar explicitamente o registro de uma instância local para que ela não possa mais ser usada em nenhuma implantação. Para obter mais informações, consulte Gerenciando operações de instâncias locais em CodeDeploy.

As instruções nesta seção mostram como configurar uma instância local e, em seguida, registrá-la e marcá-la CodeDeploy para que ela possa ser usada em implantações. Esta seção também descreve como usar CodeDeploy para obter informações sobre instâncias locais e cancelar o registro de uma instância local depois que você não planeja mais implantá-la.

Tópicos

- Pré-requisitos para configurar uma instância on-premises
- Registre uma instância local com CodeDeploy
- Gerenciando operações de instâncias locais em CodeDeploy

Pré-requisitos para configurar uma instância on-premises

Os seguintes pré-requisitos devem ser atendidos para que você possa registrar uma instância local.

Important

Se você estiver usando o comando register-on-premises-instance e atualizar periodicamente as credenciais temporárias geradas com o (), há outros pré-requisitos. AWS Security Token Service AWS STS Para mais informações, consulte Pré-requisitos de registro do ARN de sessão do IAM.

Requisitos do dispositivo

O dispositivo que você deseja preparar, registrar e marcar como uma instância local CodeDeploy deve estar executando um sistema operacional compatível. Para ver uma lista, consulte Sistemas operacionais suportados pelo CodeDeploy agente.

Se seu sistema operacional não for suportado, o CodeDeploy agente estará disponível como código aberto para você se adaptar às suas necessidades. Para obter mais informações, consulte o repositório do CodeDeploy agente em GitHub.

Comunicação de saída

A instância local deve ser capaz de se conectar aos endpoints de AWS serviço público com os quais se comunicar. CodeDeploy

O CodeDeploy agente se comunica de saída usando HTTPS pela porta 443.

Controle administrativo

A conta local ou de rede usada na instância on-premises para configurar essa instância deve ser capaz de ser executada como `sudo` ou `root` (para o Ubuntu Server) ou como administrador (para o Windows Server).

Permissões do IAM

A identidade do IAM que você usa para registrar a instância on-premises deve receber permissões para concluir o registro (e cancelar o registro da instância on-premises, conforme necessário).

Além da política descrita em Etapa 3: limitar as permissões do CodeDeploy usuário, verifique se a identidade do IAM responsável pela chamada também tem a política adicional a seguir anexada.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateAccessKey",
        "iam:CreateUser",
        "iam:DeleteAccessKey",
        "iam:DeleteUser",
        "iam:DeleteUserPolicy",
        "iam:ListAccessKeys",
        "iam:ListUserPolicies",
        "iam:PutUserPolicy",
        "iam:GetUser"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

Para obter informações sobre como anexar políticas do IAM, consulte Gerenciar políticas do IAM.

Registre uma instância local com CodeDeploy

Para registrar uma instância local, você deve usar uma identidade do IAM para autenticar suas solicitações. É possível escolher entre as seguintes opções para o método de identidade e registro do IAM utilizado:

- Use um ARN de função do IAM para autenticar solicitações.
  - Use o comando register-on-premises-instance e as credenciais temporárias atualizadas periodicamente geradas com o AWS Security Token Service ()AWS STS para configurar manualmente a maioria das opções de registro. Essa opção oferece o mais alto nível de segurança, porque a autenticação ocorre usando um token temporário que expira e deve ser atualizado periodicamente. Essa opção é recomendada para implantações de produção com qualquer tamanho. Para mais informações, consulte Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises.
- (Não recomendado) Use um ARN de usuário do IAM para autenticar solicitações.
  - Use o comando register para o processo de registro mais automatizado. Essa opção só deve ser usada para implantações que não sejam de produção, nas quais a segurança seja menos preocupante. Essa opção é menos segura porque usa credenciais estáticas (permanentes) para autenticação. Essa opção funciona bem para registrar uma única instância on-premises. Para mais informações, consulte Usar o comando register (ARN de usuário do IAM) para registrar uma instância on-premises.
  - Use o comando register-on-premises-instance para configurar manualmente a maioria das opções de registro. Adequado para registrar um pequeno número de instâncias locais. Para mais informações, consulte Usar o comando register-on-premises-instance (ARN de usuário do IAM) para registrar uma instância on-premises.

Tópicos

- Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises

- Usar o comando register (ARN de usuário do IAM) para registrar uma instância on-premises
- Usar o comando register-on-premises-instance (ARN de usuário do IAM) para registrar uma instância on-premises

Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises

Para obter o máximo controle sobre a autenticação e o registro de suas instâncias locais, você pode usar o comando register-on-premises-instance e atualizar periodicamente as credenciais temporárias geradas com o (). AWS Security Token Service AWS STS Uma função estática do IAM para a instância assume a função dessas AWS STS credenciais atualizadas para realizar operações de implantação. CodeDeploy

Esse método é mais útil quando você precisa registrar um grande número de instâncias. Ele permite que você automatize o processo de registro com CodeDeploy. Você pode usar seu próprio sistema de identidade e autenticação para autenticar instâncias locais e distribuir as credenciais de sessão do IAM do serviço para as instâncias com as quais serão usadas. CodeDeploy

> **Note**
>
> Como alternativa, você pode usar um usuário compartilhado do IAM distribuído para todas as instâncias locais para chamar a AWS STS AssumeRoleAPI e recuperar as credenciais da sessão para instâncias locais. Esse método é menos seguro e não é recomendado para ambientes de produção ou críticos.

Use as informações nos tópicos a seguir para configurar uma instância local usando credenciais de segurança temporárias geradas com. AWS STS

Tópicos

- Pré-requisitos de registro do ARN de sessão do IAM
- Etapa 1: criar o perfil do IAM que as instâncias on-premises assumirão
- Etapa 2: gerar credenciais temporárias para uma instância individual usando AWS STS
- Etapa 3: adicionar um arquivo de configuração à instância on-premises
- Etapa 4: Preparar uma instância local para CodeDeploy implantações
- Etapa 5: registrar a instância local com CodeDeploy

- Etapa 6: marcar a instância on-premises
- Etapa 7: implantar revisões de aplicações na instância on-premises
- Etapa 8: acompanhar implantações na instância on-premises

Pré-requisitos de registro do ARN de sessão do IAM

Além dos pré-requisitos listados em Pré-requisitos para configurar uma instância on-premises, os seguintes requisitos adicionais devem ser atendidos:

Permissões do IAM

A identidade do IAM que você usa para registrar uma instância local deve receber permissões para realizar CodeDeploy operações. Verifique se a política gerenciada AWSCodeDeployFullAccess está anexada à identidade do IAM. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

Sistema para atualizar credenciais temporárias

Se você usar um ARN de sessão do IAM para registrar instâncias locais, deverá ter um sistema em vigor para atualizar periodicamente as credenciais temporárias. As credenciais temporárias expiram depois de uma hora ou antes quando um período mais curto é especificado no momento em que as credenciais são geradas. Existem dois métodos para atualizar as credenciais:

- Método 1: use o sistema de identidade e autenticação instalado na sua rede corporativa com um script CRON que faça uma sondagem periódica do sistema de identidade e autenticação e que copie as credenciais da última sessão para a instância. Isso permite que você integre sua estrutura de autenticação e identidade AWS sem precisar fazer alterações no CodeDeploy agente ou no serviço para oferecer suporte aos tipos de autenticação que você usa em sua organização.
- Método 2: Execute periodicamente um trabalho CRON na instância para chamar a AWS STS AssumeRoleação e gravar as credenciais da sessão em um arquivo que o CodeDeploy agente possa acessar. Esse método ainda requer um usuário do IAM e a cópia de credenciais na instância local, mas você pode reutilizar o mesmo usuário do IAM e as mesmas credenciais por toda a sua frota de instâncias locais.

Para obter informações sobre como criar e trabalhar com AWS STS credenciais, consulte Referência AWS Security Token Service da API e Uso de credenciais de segurança temporárias para solicitar acesso aos AWS recursos.

Etapa 1: criar o perfil do IAM que as instâncias on-premises assumirão

Você pode usar o console do IAM AWS CLI ou o console do IAM para criar uma função do IAM que será usada por suas instâncias locais para autenticação e interação. CodeDeploy

Você só precisa criar uma função do IAM. Cada uma das suas instâncias locais pode assumir essa função para recuperar as credenciais de segurança temporárias que fornecem as permissões concedidas a essa função.

A função que você criar exigirá as seguintes permissões para acessar os arquivos necessários para instalar o CodeDeploy agente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Recomendamos que você restrinja essa política somente para os buckets do Amazon S3 que a sua instância on-premises precisa acessar. Se você restringir essa política, certifique-se de dar acesso aos buckets do Amazon S3 que contêm o agente. CodeDeploy Caso contrário, poderá ocorrer um erro sempre que o CodeDeploy agente for instalado ou atualizado na instância local. Para obter mais informações sobre como controlar o acesso aos buckets do Amazon S3, consulte Gerenciar permissões de acesso aos recursos do Amazon S3.

Para criar perfil do IAM

1. Chame o comando create-role usando a opção `--role-name` para especificar um nome para a função do IAM (por exemplo, `CodeDeployInstanceRole`) e a opção `--assume-role-policy-document` para fornecer as permissões.

Ao criar a função do IAM para essa instância, você pode dar a ela o nome `CodeDeployInstanceRole` e incluir as permissões necessárias em um arquivo chamado `CodeDeployRolePolicy.json`:

```
aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-
document file://CodeDeployRolePolicy.json
```

2. Na saída da chamada para o comando create-role, anote o valor do campo ARN. Por exemplo:

```
arn:aws:iam::123456789012:role/CodeDeployInstanceRole
```

Você precisará do ARN da função ao usar a AWS STS AssumeRoleAPI para gerar credenciais de curto prazo para cada instância.

Para obter mais informações sobre a criação de funções do IAM, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Para obter informações sobre como atribuir permissões a uma função existente, consulte put-role-policy na Referência de comandos da AWS CLI.

Etapa 2: gerar credenciais temporárias para uma instância individual usando AWS STS

Antes de gerar as credenciais temporárias que serão usadas para registrar uma instância local, você deve criar ou escolher a identidade do IAM (usuário ou função) para a qual você gerará essas credenciais temporárias. A permissão `sts:AssumeRole` deve ser incluída nas configurações de política para essa identidade do IAM.

Para obter informações sobre como conceder `sts:AssumeRole` permissões a uma identidade do IAM, consulte Criação de uma função para delegar permissões a um AWS serviço e. AssumeRole

Existem duas maneiras de gerar as credenciais temporárias:

- Use o comando assume-role com o AWS CLI. Por exemplo:

```
aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-
session-name session-name
```

Em que:

- *12345ACCOUNT*é o número da conta de 12 dígitos da sua organização.

- *role-arn*é o ARN da função a ser assumida, que você gerou em. Etapa 1: criar o perfil do IAM que as instâncias on-premises assumirão
- *session-name*é o nome que você deseja dar à sessão de função que você está criando agora.

> **Note**
>
> Se você usa um script CRON que pesquisa periodicamente o sistema de identidade e autenticação e copia as credenciais da sessão mais recente para a instância (método 1 para atualizar as credenciais temporárias descrito emPré-requisitos de registro do ARN de sessão do IAM), você pode usar qualquer SDK compatível para chamar. AWS AssumeRole

- Use uma ferramenta fornecida pela AWS.

  A ferramenta aws-codedeploy-session-helper gera AWS STS credenciais e as grava em um arquivo que você coloca na instância. Essa ferramenta é mais adequada para o método 2 de atualização de credenciais temporárias, descrito em Pré-requisitos de registro do ARN de sessão do IAM. Nesse método, a ferramenta aws-codedeploy-session-helper é colocada em cada instância e executa o comando usando as permissões de um usuário do IAM. Cada instância usa as mesmas credenciais do usuário do IAM em conjunto com essa ferramenta.

  Para obter mais informações, consulte o repositório GitHub aws-codedeploy-session-helper.

  > **Note**
  >
  > Depois de ter criado as credenciais de sessão do IAM, coloque-as em qualquer parte da instância local. Na próxima etapa, você configurará o CodeDeploy agente para acessar as credenciais nesse local.

Antes de continuar, verifique se o sistema que você usará para atualizar periodicamente as credenciais temporárias está em vigor. Se as credenciais temporárias não forem atualizadas, as implantações na instância local falharão. Para obter mais informações, consulte "Sistema para atualizar credenciais temporárias", em Pré-requisitos de registro do ARN de sessão do IAM.

Etapa 3: adicionar um arquivo de configuração à instância on-premises

Adicione um arquivo de configuração à instância on-premises usando permissões raiz ou de administrador. Esse arquivo de configuração é usado para declarar as credenciais do IAM e a AWS região de destino a ser usada. CodeDeploy O arquivo deve ser adicionado a um local específico na instância on-premises. O arquivo deve incluir o ARN da sessão temporária do IAM, seu ID de chave secreta e chave de acesso secreta e a região de destino AWS .

Para adicionar um arquivo de configuração

1. Crie um arquivo chamado `codedeploy.onpremises.yml` (para uma instância on-premises do Ubuntu Server ou do RHEL) ou `conf.onpremises.yml` (para uma instância on-premises do Windows Server) na seguinte localização da instância on-premises:
   - Para Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Para Windows Server: `C:\ProgramData\Amazon\CodeDeploy`
2. Use um editor de texto para adicionar as seguintes informações ao arquivo `codedeploy.onpremises.yml` (Linux) ou `conf.onpremises.yml` (Windows) recém-criado:

   ```
   ---
   iam_session_arn: iam-session-arn
   aws_credentials_file: credentials-file
   region: supported-region
   ```

   Em que:

   - *`iam-session-arn`*é o ARN da sessão do IAM em que você anotou. Etapa 2: gerar credenciais temporárias para uma instância individual usando AWS STS
   - *`credentials-file`*é o local do arquivo de credenciais para o ARN da sessão temporária, conforme observado em. Etapa 2: gerar credenciais temporárias para uma instância individual usando AWS STS
   - *`supported-region`*é uma das regiões que CodeDeploy oferece suporte, conforme listado em Região e endpoints em Referência geral da AWS.

Etapa 4: Preparar uma instância local para CodeDeploy implantações

Instale e configure o AWS CLI

Instale e configure o AWS CLI na instância local. (Eles AWS CLI serão usados para baixar e instalar o CodeDeploy agente na instância local.)

1. Para instalar o AWS CLI na instância local, siga as instruções em Como configurar com o AWS CLI no Guia do AWS Command Line Interface usuário.

   > **Note**
   >
   > CodeDeploy comandos para trabalhar com instâncias locais foram disponibilizados na versão 1.7.19 do. AWS CLI Se você tiver uma versão do AWS CLI já instalada, poderá verificar sua versão ligando paraaws --version.

2. Para configurar o AWS CLI na instância local, siga as instruções em Configurando o AWS CLI no Guia do AWS Command Line Interface usuário.

   > **Important**
   >
   > Ao configurar o AWS CLI (por exemplo, chamando o aws configure comando), certifique-se de especificar o ID da chave secreta e a chave de acesso secreta de um usuário do IAM que tenha, no mínimo, as permissões descritas emPré-requisitos de registro do ARN de sessão do IAM.

Definir a variável de ambiente AWS_REGION (somente para o Ubuntu Server e o RHEL)

Se você não estiver executando o Ubuntu Server ou o RHEL na sua instância local, pule esta etapa e vá diretamente para “Instalar o CodeDeploy agente”.

Instale o CodeDeploy agente em uma instância local do Ubuntu Server ou do RHEL e habilite a instância para atualizar o CodeDeploy agente sempre que uma nova versão estiver disponível. Você faz isso definindo a variável de `AWS_REGION` ambiente na instância como o identificador de uma das regiões suportadas pelo CodeDeploy. Recomendamos que você defina o valor para a região em que seus CodeDeploy aplicativos, grupos de implantação e revisões de aplicativos estão localizados (por exemplo,`us-west-2`). Para obter uma lista de regiões, consulte Região e endpoints no Referência geral da AWS.

Para definir a variável de ambiente, chame o seguinte usando o terminal:

```
export AWS_REGION=supported-region
```

Onde *supported-region* está o identificador da região (por exemplo,`us-west-2`).

Instale o CodeDeploy agente

- Para uma instância on-premises do Ubuntu Server, siga as instruções em Instale o CodeDeploy agente para o Ubuntu Server e depois retorne a esta página.
- Para uma instância on-premises do RHEL, siga as instruções em Instale o CodeDeploy agente para Amazon Linux ou RHEL e depois retorne a esta página.
- Para uma instância on-premises do Windows Server, siga as instruções em Instale o CodeDeploy agente para Windows Server e depois retorne a esta página.

Etapa 5: registrar a instância local com CodeDeploy

As instruções nesta etapa supõem que você esteja registrando a instância on-premises a partir dela mesma. Você pode registrar uma instância local a partir de um dispositivo ou instância separada que a tenha AWS CLI instalada e configurada.

Use o AWS CLI para registrar a instância local CodeDeploy para que ela possa ser usada em implantações.

Antes de usar o AWS CLI, você precisará do ARN das credenciais de sessão temporária que você criou. Etapa 3: adicionar um arquivo de configuração à instância on-premises Por exemplo, para uma instância identificada como `AssetTag12010298EX`:

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Chame o comando register-on-premises-instance, especificando o seguinte:

- Um nome que identifique de forma exclusiva a instância on premises (com a opção `--instance-name`).

> **Important**
>
> Para ajudar a identificar a instância local, especialmente para fins de depuração, recomendamos que você especifique um nome que seja mapeado para alguma característica exclusiva da instância local (por exemplo, o nome da sessão das credenciais do STS e o número de série ou um identificador interno de ativo, se aplicável). Se você especificar um endereço MAC como nome, saiba que os endereços MAC contêm

caracteres que CodeDeploy não são permitidos, como dois pontos (:). Para obter uma lista de caracteres permitidos, consulte CodeDeploy cotas.

- O ARN da sessão do IAM que você configurou para autenticar várias instâncias locais em Etapa 1: criar o perfil do IAM que as instâncias on-premises assumirão.

Por exemplo:

```
aws deploy register-on-premises-instance --instance-name name-of-instance --iam-session-arn arn:aws:sts::account-id:assumed-role/role-to-assume/session-name
```

Em que:

- *name-of-instance*é o nome que você usa para identificar a instância local, como`AssetTag12010298EX`.
- *account-id*é o ID da conta de 12 dígitos da sua organização, como. `111222333444`
- *role-to-assume*é o nome da função do IAM que você criou para a instância, como`CodeDeployInstanceRole`.
- *session-name*é o nome da função da sessão em que você especificouEtapa 2: gerar credenciais temporárias para uma instância individual usando AWS STS.

Etapa 6: marcar a instância on-premises

Você pode usar o console AWS CLI ou o CodeDeploy console para marcar a instância local. (CodeDeployusa tags de instância local para identificar os alvos de implantação durante uma implantação.)

Para marcar a instância on-premises (CLI)

- Chame o comando add-tags-to-on-premises-instances, especificando o seguinte:
  - O nome que identifica de forma exclusiva a instância on-premises (com a opção `--instance-names`).
  - O nome da chave da tag de instância on-premises e o valor da tag que você deseja usar (com a opção `--tags`). Você deve especificar um nome e um valor. CodeDeploy não permite tags de instância local que tenham somente valores.

    Por exemplo:

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
 --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Para marcar a instância on-premises (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Deploy e escolha On-premises instâncias.
3. Na lista de instâncias locais, selecione nome da instância local que deseja marcar.
4. Na lista de tags, selecione ou insira a chave e o valor da tag desejados. Após inserir a chave e o valor da tag, outra linha será exibida. Isso pode ser repetido para até 10 tags. Para remover uma tag, selecione Remover.
5. Depois de adicionar as tags, escolha Atualizar tags.

Etapa 7: implantar revisões de aplicações na instância on-premises

Agora, você está pronto para implantar revisões de aplicativo na instância on-premises registrada e marcada.

Você implementa revisões de aplicativos em instâncias on-premises de uma maneira semelhante à implantação de revisões de aplicativos em instâncias do Amazon EC2. Para instruções, consulte Crie uma implantação com CodeDeploy. Essas instruções incluem um link para pré-requisitos, que incluem: criar um aplicativo, criar um grupo de implantação e preparar uma revisão de aplicativo. Se você precisa de uma simples revisão de aplicativo de amostra para implantar, poderá criar aquela descrito em Etapa 2: criar uma revisão de aplicativo de amostra, no Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).

> **Important**
>
> Se você reutilizar uma função de CodeDeploy serviço como parte da criação de um grupo de implantação que tenha como alvo instâncias locais, deverá `Tag:get*` incluí-la na `Action` parte da declaração de política da função de serviço. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Etapa 8: acompanhar implantações na instância on-premises

Depois de implantar uma revisão de aplicativo em instâncias on-premises registradas e marcadas, você poderá controlar o progresso da implantação.

Você controla as implantações em instâncias on-premises de uma maneira semelhante ao controle de implantações em instâncias do Amazon EC2. Para instruções, consulte Exibir detalhes CodeDeploy da implantação.

Usar o comando register (ARN de usuário do IAM) para registrar uma instância on-premises

> **Important**
>
> O registro de uma instância usando um usuário do IAM não é recomendado porque usa credenciais estáticas (permanentes) para autenticação. Para melhorar a segurança, recomendamos registrar uma instância usando credenciais temporárias para autenticação. Para obter mais informações, consulte Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises.

> **Important**
>
> Certifique-se de ter um plano para alternar as chaves de acesso do usuário do IAM (credenciais permanentes). Para obter mais informações, consulte Alternando as chaves de acesso.

Esta seção descreve como configurar uma instância local e registrá-la e marcá-la com o CodeDeploy com o mínimo de esforço. O comando register é mais útil quando você está trabalhando com frotas

únicas ou pequenas de instâncias locais. Você pode usar o comando register somente quando está usando um ARN de usuário do IAM para autenticar uma instância. Você não pode usar o comando register com um ARN de sessão do IAM para autenticação.

Ao usar o register comando, você pode deixar CodeDeploy fazer o seguinte:

- Crie um usuário do IAM AWS Identity and Access Management para a instância local, se você não especificar um com o comando.
- Salvar as credenciais do usuário do IAM em um arquivo de configuração da instância on-premises.
- Registre a instância local com CodeDeploy.
- Adicione tags à instância local se você as especificar como parte do comando.

> **Note**
>
> O comando register-on-premises-instance é uma alternativa ao comando register. Você usa o comando register-on-premises-instance para configurar uma instância local e registrá-la e marcá-la no CodeDeploy principalmente por sua conta. O comando register-on-premises-instance também fornece a opção de usar um ARN de sessão do IAM para registrar instâncias em vez de um ARN de usuário do IAM. Essa abordagem proporciona uma grande vantagem quando você tem grandes frotas de instâncias locais. Especificamente, é possível usar um único ARN de sessão do IAM para autenticar várias instâncias em vez de ter que criar um usuário do IAM para cada instância on-premises, uma a uma. Para obter mais informações, consulte Usar o comando register-on-premises-instance (ARN de usuário do IAM) para registrar uma instância on-premises e Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises.

Tópicos

- Etapa 1: instalar e configurar o AWS CLI na instância local
- Etapa 2: chamar o comando register
- Etapa 3: chamar o comando install
- Etapa 4: implantar revisões de aplicações na instância on-premises
- Etapa 5: acompanhar implantações na instância on-premises

Etapa 1: instalar e configurar o AWS CLI na instância local

1. Instale o AWS CLI na instância local. Siga as instruções em Conceitos básicos do AWS CLI no Guia do usuário do AWS Command Line Interface .

> **Note**
>
> CodeDeploy os comandos para trabalhar com instâncias locais estão disponíveis na AWS CLI versão 1.7.19 e posterior. Se você AWS CLI já tiver o instalado, ligue aws --version para verificar sua versão.

2. Configure o AWS CLI na instância local. Siga as instruções em Configuração do AWS CLI no Guia do usuário do AWS Command Line Interface .

> **Important**
>
> Ao configurar o AWS CLI (por exemplo, chamando o aws configure comando), certifique-se de especificar o ID da chave secreta e a chave de acesso secreta de um usuário do IAM que tenha, no mínimo, as seguintes permissões de AWS acesso, além das permissões especificadas emPré-requisitos para configurar uma instância on-premises. Isso possibilita baixar e instalar o CodeDeploy agente na instância local. As permissões de acesso podem parecer semelhantes a estas:
>
> JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*",
        "iam:CreateAccessKey",
        "iam:CreateUser",
        "iam:DeleteAccessKey",
        "iam:DeleteUser",
        "iam:DeleteUserPolicy",
        "iam:ListAccessKeys",
        "iam:ListUserPolicies",
        "iam:PutUserPolicy",
```

```
        "iam:GetUser",
        "tag:getTagKeys",
        "tag:getTagValues",
        "tag:GetResources"
      ],
      "Resource" : "*"
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource" : [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

> **Note**
>
> Se você ver erros de acesso negado ao tentar acessar um dos buckets do Amazon S3 mostrados anteriormente, tente omitir a parte `/*` do ARN do recurso do bucket, por exemplo, `arn:aws:s3:::aws-codedeploy-sa-east-1`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*",
        "iam:CreateAccessKey",
        "iam:CreateUser",
        "iam:DeleteAccessKey",
```

```
        "iam:DeleteUser",
        "iam:DeleteUserPolicy",
        "iam:ListAccessKeys",
        "iam:ListUserPolicies",
        "iam:PutUserPolicy",
        "iam:GetUser",
        "tag:GetResources"
      ],
      "Resource" : "*"
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource" : [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket2/*"
      ]
    }
  ]
}
```

Etapa 2: chamar o comando register

Para esta etapa, estamos supondo que você esteja registrando a instância local a partir dela mesma. Você também pode registrar uma instância local a partir de um dispositivo ou instância separada que tenha o AWS CLI instalado e configurado conforme descrito na etapa anterior.

Use o AWS CLI para chamar o comando register, especificando:

- Um nome que identifica de forma exclusiva a instância local para CodeDeploy (com a opção). `--instance-name`

> **Important**
>
> Para ajudar a identificar a instância local, especialmente para fins de depuração, recomendamos que você use um nome que seja mapeado para algumas características exclusivas da instância local (por exemplo, o número de série ou algum identificador de

ativo interno exclusivo, se aplicável). Se você especificar um endereço MAC para um nome, lembre-se de que os endereços MAC contêm caracteres que CodeDeploy não permitem, como dois pontos (`:`). Para obter uma lista de caracteres permitidos, consulte CodeDeploy cotas.

- Opcionalmente, o ARN de um usuário do IAM existente que você deseja associar a essa instância on-premises (com a opção `--iam-user-arn`). Para acessar o ARN de um usuário do IAM, chame o comando get-user ou escolha o nome do usuário do IAM na seção Usuários do console do IAM e localize o valor de ARN do usuário na seção Resumo. Se essa opção não for especificada, CodeDeploy criará um usuário do IAM em seu nome em sua AWS conta e o associará à instância local.

> **Important**
>
> Se você especificar a opção `--iam-user-arn`, também deverá criar manualmente o arquivo de configuração da instância local, conforme descrito em Etapa 4: adicionar um arquivo de configuração à instância on-premises.
> É possível associar apenas um usuário do IAM a apenas uma instância on-premises. Tentar associar um único usuário do IAM a várias instâncias on-premises pode resultar em erros, implantações com falhas nessas instâncias on-premises ou implantações em instâncias on-premises que estão presas em um estado pendente perpétuo.

- Opcionalmente, um conjunto de tags de instância local (com a `--tags` opção) que CodeDeploy serão usadas para identificar o conjunto de instâncias do Amazon EC2 nas quais implantar. Especifique cada tag com `Key=`*`tag-key`*`,Value=`*`tag-value`* (por exemplo, `Key=Name,Value=Beta Key=Name,Value=WestRegion`). Se essa opção não for especificada, nenhuma tag será registrada. Para registrar tags mais tarde, chame o comando add-tags-to-on-premises-instances.
- Opcionalmente, a AWS região na qual a instância local será registrada CodeDeploy (com a `--region` opção). Essa deve ser uma das regiões suportadas listadas em Região e endpoints em Referência geral da AWS (por exemplo,`us-west-2`). Se essa opção não for especificada, a AWS região padrão associada ao usuário chamador do IAM será usada.

Por exemplo:

```
aws deploy register --instance-name AssetTag12010298EX --iam-user-
arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem --tags
 Key=Name,Value=CodeDeployDemo-OnPrem --region us-west-2
```

O comando register faz o seguinte:

1. Se nenhum usuário existente do IAM for especificado, cria um usuário do IAM, anexa as permissões necessárias a ele e gera uma chave secreta e uma chave de acesso secreta correspondentes. A instância local usará esse usuário do IAM e suas permissões e credenciais para autenticar e interagir com ele. CodeDeploy
2. Registra a instância local com. CodeDeploy
3. Se especificado, associa CodeDeploy nas tags especificadas à `--tags` opção com o nome da instância local registrada.
4. Se um usuário do IAM tiver sido criado, também criará o arquivo de configuração necessário no mesmo diretório do qual o comando register foi chamado.

Se esse comando encontrar erros, será exibida uma mensagem de erro descrevendo como você pode concluir manualmente as etapas restantes. Caso contrário, será exibida uma mensagem de sucesso, descrevendo como chamar o comando install, conforme listado na próxima etapa.

Etapa 3: chamar o comando install

Na instância local, use o AWS CLI para chamar o comando de instalação, especificando:

- O caminho para o arquivo de configuração (com a opção `--config-file`).
- Opcionalmente, seja para substituir o arquivo de configuração já existente na instância local (com a opção `--override-config`). Se não for especificado, o arquivo de configuração existente não será substituído.
- Opcionalmente, a AWS região na qual a instância local será registrada CodeDeploy (com a `--region` opção). Essa deve ser uma das regiões suportadas listadas em Região e endpoints em Referência geral da AWS (por exemplo,`us-west-2`). Se essa opção não for especificada, a AWS região padrão associada ao usuário chamador do IAM será usada.
- Opcionalmente, um local personalizado a partir do qual instalar o CodeDeploy agente (com a `--agent-installer` opção). Essa opção é útil para instalar uma versão personalizada do CodeDeploy agente que CodeDeploy não oferece suporte oficial (como uma versão personalizada baseada no repositório do CodeDeploy agente em GitHub). O valor deve ser o caminho para um bucket do Amazon S3 que contém:

- Um script de instalação do CodeDeploy agente (para sistemas Unix-based operacionais ou Linux, semelhante ao arquivo de instalação no repositório do CodeDeploy agente em GitHub).
- Um arquivo de pacote do instalador do CodeDeploy agente (.msi) (para sistemas Windows-based operacionais).

Se essa opção não for especificada, CodeDeploy tentará instalar de seu próprio local uma versão oficialmente suportada do CodeDeploy agente que seja compatível com o sistema operacional na instância local.

Por exemplo:

```
aws deploy install --override-config --config-file /tmp/codedeploy.onpremises.yml --
region us-west-2 --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-
agent.msi
```

O comando install faz o seguinte:

1. Verifica se a instância on-premises é uma instância do Amazon EC2. Se for, uma mensagem de erro será exibida.
2. Copia o arquivo de configuração das instâncias locais do local especificado na instância para o local onde o CodeDeploy agente espera encontrá-lo, desde que o arquivo ainda não esteja nesse local.

   Para Ubuntu Server e Red Hat Enterprise Linux (RHEL)), isso é `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`.

   Para o Windows Server, isso é `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.

   Se a opção `--override-config` tiver sido especificada, criará ou substituirá o arquivo.
3. Instala o CodeDeploy agente na instância local e depois a inicia.

Etapa 4: implantar revisões de aplicações na instância on-premises

Agora, você está pronto para implantar revisões de aplicativo na instância on-premises registrada e marcada.

Você implementa revisões de aplicativos em instâncias on-premises de uma maneira semelhante à implantação de revisões de aplicativos em instâncias do Amazon EC2. Para instruções, consulte Crie uma implantação com CodeDeploy. Essas instruções contêm um link para pré-requisitos, que incluem: criar um aplicativo, criar um grupo de implantação e preparar uma revisão de aplicativo. Se você precisa de uma simples revisão de aplicativo de amostra para implantar, poderá criar aquela descrito em Etapa 2: criar uma revisão de aplicativo de amostra, no Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).

> **Important**
>
> Se você reutilizar uma função de CodeDeploy serviço existente como parte da criação de um grupo de implantação voltado `Tag:get*` para instâncias locais, deverá incluir na `Action` parte da declaração de política da função de serviço. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Etapa 5: acompanhar implantações na instância on-premises

Depois de implantar uma revisão de aplicativo em instâncias on-premises registradas e marcadas, você poderá controlar o progresso da implantação.

Você controla as implantações em instâncias on-premises de uma maneira semelhante ao controle de implantações em instâncias do Amazon EC2. Para instruções, consulte Exibir detalhes CodeDeploy da implantação.

Para obter mais opções, consulte Gerenciando operações de instâncias locais em CodeDeploy.

Usar o comando register-on-premises-instance (ARN de usuário do IAM) para registrar uma instância on-premises

> **Important**
>
> O registro de uma instância usando um usuário do IAM não é recomendado porque usa credenciais estáticas (permanentes) para autenticação. Para melhorar a segurança, recomendamos registrar uma instância usando credenciais temporárias para autenticação. Para obter mais informações, consulte Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises.

> **Important**
>
> Certifique-se de ter um plano para alternar as chaves de acesso do usuário do IAM (credenciais permanentes). Para obter mais informações, consulte Alternando as chaves de acesso.

Siga essas instruções para configurar uma instância local e registrá-la e marcá-la no CodeDeploy principalmente por conta própria, usando credenciais estáticas de usuário do IAM para autenticação.

Tópicos

- Etapa 1: criar um usuário do IAM para a instância on-premises
- Etapa 2: atribuir permissões ao usuário do IAM
- Etapa 3: obter as credenciais do usuário do IAM
- Etapa 4: adicionar um arquivo de configuração à instância on-premises
- Etapa 5: instalar e configurar o AWS CLI
- Etapa 6: definir a variável de ambiente AWS_REGION (somente para o Ubuntu Server e o RHEL)
- Etapa 7: instalar o CodeDeploy agente
- Etapa 8: registrar a instância local com CodeDeploy
- Etapa 9: marcar a instância on-premises
- Etapa 10: implantar revisões de aplicações na instância on-premises
- Etapa 11: acompanhar implantações na instância on-premises

Etapa 1: criar um usuário do IAM para a instância on-premises

Crie um usuário do IAM que a instância local usará para autenticar e interagir com ele. CodeDeploy

> **Important**
>
> Você deve criar um usuário do IAM separado para cada instância on-premises participante. Se você tentar reutilizar um usuário individual do IAM para várias instâncias locais, talvez não consiga registrar ou marcar essas instâncias locais com êxito. CodeDeploy As implementações nessas instâncias locais podem estar presas em um estado pendente perpétuo ou podem falhar completamente.

Recomendamos que você atribua ao usuário do IAM um nome que identifique sua finalidade, como CodeDeployUser-OnPrem.

Você pode usar o console do IAM AWS CLI ou o console do IAM para criar um usuário do IAM. Para obter informações, consulte Criação de um usuário do IAM na sua conta da AWS.

> **Important**
>
> Se você usa o console do IAM AWS CLI ou o console do IAM para criar um novo usuário do IAM, anote o ARN do usuário fornecido para o usuário. Você precisará dessas informações mais tarde em Etapa 4: adicionar um arquivo de configuração à instância on-premises e Etapa 8: registrar a instância local com CodeDeploy.

Etapa 2: atribuir permissões ao usuário do IAM

Se a instância on-premises for implantar revisões de aplicativos de buckets do Amazon S3, será necessário atribuir ao usuário do IAM as permissões para interagir com esses buckets. Você pode usar o console AWS CLI ou o IAM para atribuir permissões.

> **Note**
>
> Se você estiver implantando revisões de aplicativos somente a partir de GitHub repositórios, ignore esta etapa e vá diretamente para o. Etapa 3: obter as credenciais do usuário do IAM (Você ainda precisará das informações sobre o usuário do IAM criado em Etapa 1: criar um usuário do IAM para a instância on-premises. Elas serão usadas em etapas posteriores.)

Para atribuir permissões (CLI)

1. Crie um arquivo com o seguinte conteúdo de política na instância do Amazon EC2 ou no dispositivo que você está usando para chamar a AWS CLI. Nomeie o arquivo de algo como **`CodeDeploy-OnPrem-Permissions.json`** e, em seguida, salve-o.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
```

```
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Recomendamos que você restrinja essa política somente para os buckets do Amazon S3 que a sua instância on-premises precisa acessar. Se você restringir essa política, certifique-se de também dar acesso aos buckets do Amazon S3 que contêm o AWS CodeDeploy agente. Caso contrário, poderá ocorrer um erro sempre que o CodeDeploy agente for instalado ou atualizado na instância local associada.
> Por exemplo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
```

```
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
        "arn:aws:s3:::aws-codedeploy-me-central-1/*",
        "arn:aws:s3:::aws-codedeploy-me-south-1/*",
        "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
      ]
    }
  ]
}
```

2. Chame o comando put-user-policy, especificando o nome do usuário do IAM (com a opção `--user-name`), um nome para a política (com a opção `--policy-name`) e o caminho para o documento de política recém-criado (com a opção `--policy-document`). Por exemplo, supondo que o arquivo **`CodeDeploy-OnPrem-Permissions.json`** esteja no mesmo diretório (pasta) a partir do qual você está chamando esse comando:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-
OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

Para atribuir permissões (console)

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies e, em seguida, Create Policy. (Se aparecer um botão Get Started, selecione-o e, em seguida, Create Policy.)

3. Próximo a Create Your Own Policy, escolha Select.
4. Na caixa Policy Name (Nome da política), digite um nome para essa política (por exemplo, **`CodeDeploy-OnPrem-Permissions`**).
5. Na caixa Documento de política, digite ou cole a seguinte expressão de permissões, que permite AWS CodeDeploy implantar revisões de aplicativos de qualquer bucket do Amazon S3 especificado na política na instância local em nome do usuário do IAM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

6. Escolha Create Policy.
7. No painel de navegação, escolha Users.
8. Na lista de usuários, vá até o nome do usuário do IAM que você criou em Etapa 1: criar um usuário do IAM para a instância on-premises e escolha-o em seguida.
9. Na guia Permissões, em Políticas gerenciadas, escolha Anexar política.
10. Selecione a política chamada **`CodeDeploy-OnPrem-Permissions`** e, em seguida, escolha Attach Policy (Anexar política).

Etapa 3: obter as credenciais do usuário do IAM

Obtenha o ID de chave secreta e a chave de acesso secreta para o usuário do IAM. Você precisará deles para Etapa 4: adicionar um arquivo de configuração à instância on-premises. Você pode usar o console AWS CLI ou o IAM para obter o ID da chave secreta e a chave de acesso secreta.

Note

Se você já tem o ID de chave secreta e a chave de acesso secreta, pula etapa e vá diretamente para Etapa 4: adicionar um arquivo de configuração à instância on-premises.
Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do Console de gerenciamento da AWS. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.
Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Recomendado) Use as credenciais do console como credenciais temporárias para assinar solicitações programáticas para os AWS SDKs ou AWS CLI APIs. AWS	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Login para desenvolvimento AWS local no Guia AWS Command Line Interface do usuário. • Para AWS SDKs, consulte Login para desenvolvimento AWS local no Guia de referência de AWS SDKs e ferramentas.

Qual usuário precisa de acesso programático?	Para	Por
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para os AWS SDKs AWS CLI ou APIs. AWS	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Configurando o AWS CLI para uso Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de AWS SDKs e ferrament as.
IAM	Use credenciais temporárias para assinar solicitações programáticas para os AWS SDKs AWS CLI ou APIs. AWS	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitaç ões programáticas para os AWS CLI AWS SDKs ou APIs. AWS	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de AWS SDKs e ferramentas. • Para AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Para obter as credenciais (CLI)

1. Chame o comando list-access-keys, especificando o nome do usuário do IAM (com a opção `--user-name`) e consultando apenas os IDs de chave de acesso (com as opções `--query` e `--output`). Por exemplo:

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
 "AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Se nenhuma chave aparecer na saída ou se informações sobre apenas uma chave aparecerem na saída, chame o comando create-access-key, especificando o nome do usuário do IAM (com a opção `--user-name`):

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

Na saída da chamada para o comando create-access-key, anote os valores dos campos `AccessKeyId` e `SecretAccessKey`. Você precisará dessas informações em Etapa 4: adicionar um arquivo de configuração à instância on-premises.

> **Important**
>
> Esta será a única vez que você terá acesso a essa chave de acesso secreta. Se você esquecer ou perder o acesso a essa chave de acesso secreta, precisará gerar uma nova, seguindo as etapas em Etapa 3: obter as credenciais do usuário do IAM.

3. Se duas chaves de acesso já estiverem listadas, você deverá excluir uma delas, chamando o comando delete-access-key, especificando o nome do usuário do IAM (com a opção `--user-name`) e a ID da chave de acesso a ser excluída (com a opção `--access-key-id`). Em seguida, chame o comando create-access-key, conforme descrito anteriormente nessa etapa. Aqui está um exemplo de como chamar o comando delete-access-key:

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-key-ID
```

> **Important**
>
> Se você chamar o comando delete-access-key para excluir uma dessas chaves de acesso, e uma instância on-premises já estiver usando essa chave de acesso conforme descrito em Etapa 4: adicionar um arquivo de configuração à instância on-premises, será necessário seguir novamente as instruções em Etapa 4: adicionar um arquivo de configuração à instância on-premises para especificar um ID de chave de acesso diferente e a chave de acesso secreta associada a esse usuário do IAM. Caso contrário, quaisquer implementações nessa instância local poderão ficar presas em um estado pendente perpétuo ou falhar completamente.

Para obter as credenciais (console)

1. a. Abra o console do IAM em https://console.aws.amazon.com/iam/.
   b. Se a lista de usuários não for exibida, no painel de navegação, escolha Usuários.
   c. Na lista de usuários, vá até o nome do usuário do IAM que você criou em Etapa 1: criar um usuário do IAM para a instância on-premises e escolha-o em seguida.
2. Na guia Credenciais de segurança, se nenhuma chave ou apenas uma chave for listada, escolha Criar chave de acesso.

   Se duas chaves de acesso estiverem listadas, você deverá excluir uma delas. Escolha Excluir ao lado de uma das chaves de acesso e, em seguida, escolha Criar chave de acesso.

   > **Important**
   >
   > Se você escolher Excluir ao lado de uma dessas chaves de acesso, e uma instância on-premises já estiver usando essa chave de acesso conforme descrito em Etapa 4: adicionar um arquivo de configuração à instância on-premises, você precisará seguir novamente as instruções em Etapa 4: adicionar um arquivo de configuração à instância on-premises para especificar um ID de chave de acesso diferente e a chave de acesso secreta associada a esse usuário do IAM. Caso contrário, as implementações nessa instância local poderão ficar presas em um estado pendente perpétuo ou falhar completamente.

3. Escolha Mostrar e anote o ID de chave de acesso e a chave de acesso secreta. Você precisará dessas informações para a próxima etapa. Como alternativa, você pode escolher Fazer download do arquivo .csv para salvar uma cópia do ID de chave de acesso e da chave de acesso secreta.

   > **Important**
   >
   > A menos que você anote ou baixe as credenciais, este será seu único acesso a essa chave de acesso secreta. Se você esquecer ou perder o acesso a essa chave de acesso secreta, precisará gerar uma nova, seguindo as etapas em Etapa 3: obter as credenciais do usuário do IAM.

4. Escolha Fechar para retornar à *IAM User Name* página Usuários >.

Etapa 4: adicionar um arquivo de configuração à instância on-premises

Adicione um arquivo de configuração à instância on-premises usando permissões raiz ou de administrador. Esse arquivo de configuração será usado para declarar as credenciais do usuário do IAM e a AWS região de destino a ser usada CodeDeploy. O arquivo deve ser adicionado a um local específico na instância on-premises. O arquivo deve incluir o ARN do usuário do IAM, o ID da chave secreta, a chave de acesso secreta e a AWS região de destino. O arquivo deve seguir um formato específico.

1. Crie um arquivo chamado `codedeploy.onpremises.yml` (para uma instância on-premises do Ubuntu Server ou do RHEL) ou `conf.onpremises.yml` (para uma instância on-premises do Windows Server) na seguinte localização da instância on-premises:

   - Para Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Para Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Use um editor de texto para adicionar as seguintes informações ao arquivo `codedeploy.onpremises.yml` ou `conf.onpremises.yml` recém-criado:

   ```
   ---
   aws_access_key_id: secret-key-id
   aws_secret_access_key: secret-access-key
   iam_user_arn: iam-user-arn
   region: supported-region
   ```

   Em que:

   - *`secret-key-id`*é o ID da chave secreta do usuário do IAM correspondente que você Etapa 1: criar um usuário do IAM para a instância on-premises anotou em ouEtapa 3: obter as credenciais do usuário do IAM.
   - *`secret-access-key`*é a chave de acesso secreta do usuário do IAM correspondente que você Etapa 1: criar um usuário do IAM para a instância on-premises anotou em ouEtapa 3: obter as credenciais do usuário do IAM.
   - *`iam-user-arn`*é o ARN do usuário do IAM que você anotou anteriormente. Etapa 1: criar um usuário do IAM para a instância on-premises
   - *`supported-region`*é o identificador de uma região suportada pelo CodeDeploy local onde seus CodeDeploy aplicativos, grupos de implantação e revisões de aplicativos estão localizados (por exemplo,`us-west-2`). Para obter uma lista de regiões, consulte Região e endpoints no Referência geral da AWS.

> **Important**
>
> Se você escolheu Excluir ao lado de uma das chaves de acesso em Etapa 3: obter as credenciais do usuário do IAM e sua instância on-premises já estiver usando o ID da chave de acesso e a chave de acesso secreta associados, será necessário seguir as instruções em Etapa 4: adicionar um arquivo de configuração à instância on-premises para especificar um ID de chave de acesso e uma chave de acesso secreta diferentes associados a esse usuário do IAM. Caso contrário, quaisquer implementações na sua instância local poderão ficar presas em um estado pendente perpétuo ou falhar completamente.

Etapa 5: instalar e configurar o AWS CLI

Instale e configure o AWS CLI na instância local. (Eles AWS CLI serão usados Etapa 7: instalar o CodeDeploy agente para baixar e instalar o CodeDeploy agente na instância local.)

1. Para instalar o AWS CLI na instância local, siga as instruções em Como configurar com o AWS CLI no Guia do AWS Command Line Interface usuário.

   > **Note**
   >
   > CodeDeploy comandos para trabalhar com instâncias locais foram disponibilizados na versão 1.7.19 do. AWS CLI Se você tiver uma versão do AWS CLI já instalada, pode verificar sua versão ligando paraaws --version.

2. Para configurar o AWS CLI na instância local, siga as instruções em Configurando o AWS CLI no Guia do AWS Command Line Interface usuário.

   > **Important**
   >
   > Ao configurar o AWS CLI (por exemplo, chamando o aws configure comando), certifique-se de especificar o ID da chave secreta e a chave de acesso secreta de um usuário do IAM que tenha, no mínimo, as seguintes permissões de AWS acesso, além das permissões de acesso especificadas noPré-requisitos para configurar uma instância on-

premises. Isso possibilita que você baixe e instale o CodeDeploy agente na instância local:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*"
      ],
      "Resource" : "*"
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource" : [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

JSON

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

```
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource" : [
        "*"
      ]
    }
  ]
}
```

Essas permissões de acesso podem ser atribuídas ao usuário do IAM que você criou em Etapa 1: criar um usuário do IAM para a instância on-premises ou a um usuário do IAM diferente. Para atribuir essas permissões a um usuário do IAM, siga as instruções em Etapa 1: criar um usuário do IAM para a instância on-premises usando essas permissões de acesso em vez das especificadas nesta etapa.

Etapa 6: definir a variável de ambiente AWS_REGION (somente para o Ubuntu Server e o RHEL)

Se não estiver executando o Ubuntu Server ou o RHEL na sua instância on-premises, ignore esta etapa e vá diretamente para Etapa 7: instalar o CodeDeploy agente.

Instale o CodeDeploy agente em uma instância local do Ubuntu Server ou RHEL e habilite a instância para atualizar o CodeDeploy agente sempre que uma nova versão estiver disponível. Você faz isso definindo a variável de `AWS_REGION` ambiente na instância como o identificador de uma das regiões suportadas pelo CodeDeploy. Recomendamos que você defina o valor para a região em que seus CodeDeploy aplicativos, grupos de implantação e revisões de aplicativos estão localizados (por exemplo,`us-west-2`). Para obter uma lista de regiões, consulte Região e endpoints no Referência geral da AWS.

Para definir a variável de ambiente, chame o seguinte usando o terminal:

```
export AWS_REGION=supported-region
```

Onde *`supported-region`* está o identificador da região (por exemplo,`us-west-2`).

Etapa 7: instalar o CodeDeploy agente

Instale o CodeDeploy agente na instância local:

- Para uma instância on-premises do Ubuntu Server, siga as instruções em Instale o CodeDeploy agente para o Ubuntu Server e depois retorne a esta página.
- Para uma instância on-premises do RHEL, siga as instruções em Instale o CodeDeploy agente para Amazon Linux ou RHEL e depois retorne a esta página.
- Para uma instância on-premises do Windows Server, siga as instruções em Instale o CodeDeploy agente para Windows Server e depois retorne a esta página.

Etapa 8: registrar a instância local com CodeDeploy

As instruções nesta etapa supõem que você esteja registrando a instância on-premises a partir dela mesma. Você pode registrar uma instância local a partir de um dispositivo ou instância separada que tenha o AWS CLI instalado e configurado, conforme descrito emEtapa 5: instalar e configurar o AWS CLI.

Use o AWS CLI para registrar a instância local CodeDeploy para que ela possa ser usada em implantações.

1. Antes de usar o AWS CLI, você precisará do ARN do usuário do IAM em Etapa 1: criar um usuário do IAM para a instância on-premises que você criou. Se você ainda não possui o ARN do usuário, chame o comando get-user, especificando o nome do usuário do IAM (com a opção `--user-name`) e consultando o ARN do usuário (com as opções `--query` e `--output`):

   ```
   aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
   ```

2. Chame o comando register-on-premises-instance, especificando o seguinte:

   - Um nome que identifique de forma exclusiva a instância on premises (com a opção `--instance-name`).

     > **Important**
     >
     > Para ajudar a identificar a instância local, especialmente para fins de depuração, recomendamos que você especifique um nome que seja mapeado para alguma característica exclusiva da instância local (por exemplo, o número de série ou um identificador interno de ativo, se aplicável). Se você especificar um endereço

MAC como nome, lembre-se de que os endereços MAC contêm caracteres que CodeDeploy não permitem, como dois pontos (`:`). Para obter uma lista de caracteres permitidos, consulte CodeDeploy cotas.

- O ARN do usuário do IAM que você criou em Etapa 1: criar um usuário do IAM para a instância on-premises (com a opção `--iam-user-arn`).

  Por exemplo:

```
aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --
iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
```

Etapa 9: marcar a instância on-premises

Você pode usar o console AWS CLI ou o CodeDeploy console para marcar a instância local. (CodeDeployusa tags de instância locais para identificar os alvos de implantação durante uma implantação.)

Para marcar a instância on-premises (CLI)

- Chame o comando add-tags-to-on-premises-instances, especificando o seguinte:

  - O nome que identifica de forma exclusiva a instância on-premises (com a opção `--instance-names`).
  - O nome da chave da tag de instância on-premises e o valor da tag que você deseja usar (com a opção `--tags`). Você deve especificar um nome e um valor. CodeDeploy não permite tags de instância locais que tenham somente valores.

    Por exemplo:

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
 --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Para marcar a instância on-premises (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. CodeDeploy No menu, escolha On-premises instâncias.
3. Na lista de instâncias locais, escolha a seta ao lado da instância local que você deseja marcar.
4. Na lista de tags, selecione ou insira a chave e o valor da tag desejados. Após inserir a chave e o valor da tag, outra linha será exibida. Isso pode ser repetido para até 10 tags. Para remover uma tag, selecione o ícone de exclusão ( ).
5. Depois de adicionar as tags, escolha Atualizar tags.

Etapa 10: implantar revisões de aplicações na instância on-premises

Agora, você está pronto para implantar revisões de aplicativo na instância on-premises registrada e marcada.

Você implementa revisões de aplicativos em instâncias on-premises de uma maneira semelhante à implantação de revisões de aplicativos em instâncias do Amazon EC2. Para instruções, consulte Crie uma implantação com CodeDeploy. Essas instruções incluem um link para pré-requisitos, que incluem: criar um aplicativo, criar um grupo de implantação e preparar uma revisão de aplicativo. Se você precisa de uma simples revisão de aplicativo de amostra para implantar, poderá criar aquela descrito em Etapa 2: criar uma revisão de aplicativo de amostra, no Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux).

> **Important**
>
> Se você reutilizar uma função de CodeDeploy serviço como parte da criação de um grupo de implantação voltado `Tag:get*` para instâncias locais, deverá incluir na `Action` parte da declaração de política da função de serviço. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Etapa 11: acompanhar implantações na instância on-premises

Depois de implantar uma revisão de aplicativo em instâncias on-premises registradas e marcadas, você poderá controlar o progresso da implantação.

Você controla as implantações em instâncias on-premises de uma maneira semelhante ao controle de implantações em instâncias do Amazon EC2. Para instruções, consulte Exibir detalhes CodeDeploy da implantação.

Gerenciando operações de instâncias locais em CodeDeploy

Siga as instruções nesta seção para gerenciar as operações em suas instâncias locais depois de registrá-las CodeDeploy, como obter mais informações, remover tags, desinstalar e cancelar o registro de instâncias locais.

Tópicos

- Obter informações sobre uma única instância on-premises
- Obter informações sobre várias instâncias on-premises
- Remover manualmente tags de instância local de uma instância on-premises
- Desinstale automaticamente o CodeDeploy agente e remova o arquivo de configuração de uma instância local
- Cancelar automaticamente o registro de uma instância on-premises
- Cancelar manualmente o registro de uma instância on-premises

Obter informações sobre uma única instância on-premises

Você pode obter informações sobre uma única instância local, seguindo as instruções em Exibir detalhes CodeDeploy da implantação. Você pode usar o console AWS CLI ou o CodeDeploy console para obter mais informações sobre uma única instância local.

Para obter informações sobre uma única instância local (CLI)

- Chame o comando get-on-premises-instance, especificando o nome que identifica exclusivamente a instância on-premises (com a opção `--instance-name`):

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Para obter informações sobre uma única instância local (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Deploy e escolha On-premises instâncias.
3. Na lista de instâncias no local, escolha a nome de um nas instâncias locais para visualizar seus detalhes.

Obter informações sobre várias instâncias on-premises

Você pode obter informações sobre instâncias locais seguindo as instruções em Exibir detalhes CodeDeploy da implantação. Você pode usar o console AWS CLI ou o CodeDeploy console para obter mais informações sobre instâncias locais.

Para obter informações sobre várias instâncias locais (CLI)

1. Para obter uma lista de nomes de instâncias on-premises, chame o comando list-on-premises-instances, especificando:

   - Seja para obter informações sobre todas as instâncias locais com ou sem registro (com a opção `--registration-status` e `Registered` ou `Deregistered`, respectivamente). Se você omitir isso, nomes de instâncias com ou sem registro serão retornados.
   - Seja para obter informações apenas sobre instâncias locais marcadas com tags de instância local específicas (com a opção `--tag-filters`). Para cada tag de instância local, especifique `Key`, `Value` e `Type` (que sempre deve ser `KEY_AND_VALUE`). Separe várias marcas de instância local com espaços entre cada tripleto `Key`, `Value` e `Type`.

   Por exemplo:

```
aws deploy list-on-premises-instances --registration-status Registered
 --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
 Key=Name,Value=CodeDeployDemo-OnPrem-Beta,Type=KEY_AND_VALUE
```

2. Para obter informações mais detalhadas, chame o comando batch-get-on-premises-instances, com os nomes das instâncias on-premises (com a opção `--instance-names`):

```
aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
 AssetTag09920444EX
```

Para obter informações sobre várias instâncias locais (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Deploy e escolha On-premises instâncias.

   Informações sobre as instâncias locais são exibidas.

Remover manualmente tags de instância local de uma instância on-premises

Normalmente, você remove uma tag de instância local de uma instância local quando essa tag não está mais sendo usada ou quando você deseja remover a instância local de qualquer grupo de implantação que dependa dessa tag. Você pode usar o console AWS CLI ou o AWS CodeDeploy console para remover as tags de instância local das instâncias locais.

Você não precisa remover as tags de instância local de uma instância local antes de cancelar o registro de instância.

A remoção manual de tags de instância local de uma instância local não cancela o registro dessa instância. Ele não desinstala o CodeDeploy agente da instância. Ele não remove o arquivo de configuração da instância. Ele não exclui o usuário do IAM associado à instância.

Para cancelar automaticamente o registro da instância on-premises, consulte Cancelar automaticamente o registro de uma instância on-premises.

Para cancelar manualmente o registro da instância on-premises, consulte Cancelar manualmente o registro de uma instância on-premises.

Para desinstalar automaticamente o CodeDeploy agente e remover o arquivo de configuração da instância local, consulteDesinstale automaticamente o CodeDeploy agente e remova o arquivo de configuração de uma instância local.

Para desinstalar manualmente somente o CodeDeploy agente da instância local, consulteGerenciando as operações CodeDeploy do agente.

Para excluir manualmente o usuário do IAM associado, consulte Excluir um usuário do IAM da sua conta da AWS.

Para remover tags de instância local de uma instância no local (CLI)

- Chame o comando remove-tags-from-on-premises-instances, especificando:

  - Os nomes que identificam de forma exclusiva a instância local (com a opção `--instance-names`).
  - Os nomes e valores das tags que você deseja remover (com a opção `--tags`).

    Por exemplo:

```
aws deploy remove-tags-from-on-premises-instances --instance-names
 AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Para remover tags de instância local de uma instância no local (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Deploy e escolha On-premises instâncias.
3. Na lista de instâncias locais, escolha a nome da instância no local da qual você deseja remover tags.
4. Nas seções Tags, escolha Remove (Remover) próximo a cada tag que você deseja remover.
5. Depois de excluir as tags, escolha Atualizar tags.

Desinstale automaticamente o CodeDeploy agente e remova o arquivo de configuração de uma instância local

Normalmente, você desinstala o CodeDeploy agente e remove o arquivo de configuração de uma instância local depois de não planejar mais implantá-la.

> **Note**
>
> A desinstalação automática do CodeDeploy agente e a remoção do arquivo de configuração de uma instância local não cancela o registro de uma instância local. Ele não desassocia nenhuma tag de instância local associada à instância local. Ele não exclui o usuário do IAM associado à instância on-premises.
> Para cancelar automaticamente o registro da instância on-premises, consulte Cancelar automaticamente o registro de uma instância on-premises.
> Para cancelar manualmente o registro da instância on-premises, consulte Cancelar manualmente o registro de uma instância on-premises.
> Para desassociar manualmente qualquer tag de instância on-premises associada, consulte Remover manualmente tags de instância local de uma instância on-premises.
> Para desinstalar manualmente o CodeDeploy agente da instância local, consulteGerenciando as operações CodeDeploy do agente.
> Para excluir manualmente o usuário do IAM associado, consulte Excluir um usuário do IAM da sua conta da AWS.

Na instância local, use o AWS CLI para chamar o comando uninstall.

Por exemplo:

```
aws deploy uninstall
```

O comando uninstall faz o seguinte:

1. Interrompe a execução do CodeDeploy agente na instância local.
2. Desinstala o CodeDeploy agente da instância local.
3. Remove o arquivo de configuração da instância local. (Para o Ubuntu Server e o RHEL, isso é `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`. Para o Windows Server, isso é `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

Cancelar automaticamente o registro de uma instância on-premises

Normalmente, você cancela o registro de uma instância on-premises quando não planeja mais implantá-la. Quando você cancela o registro de uma instância local, mesmo que essa instância local possa fazer parte das tags de instância local de um grupo de implantação, ela não é incluída em nenhuma implantação. Você pode usar o AWS CLI para cancelar o registro de instâncias locais.

> **Note**
>
> Você não pode usar o CodeDeploy console para cancelar o registro de uma instância local. Além disso, cancelar o registro de uma instância no local remove qualquer tag de instância no local que esteja associada à instância no local. Ele não desinstala o CodeDeploy agente da instância local. Ele não remove o arquivo de configuração da instância local.
> Para usar o CodeDeploy console para realizar algumas (mas não todas) as atividades desta seção, consulte a seção CodeDeploy console doCancelar manualmente o registro de uma instância on-premises.
> Para desassociar manualmente qualquer tag de instância on-premises associada, consulte Remover manualmente tags de instância local de uma instância on-premises.
> Para desinstalar automaticamente o CodeDeploy agente e remover o arquivo de configuração da instância local, consulteDesinstale automaticamente o CodeDeploy agente e remova o arquivo de configuração de uma instância local.
> Para desinstalar manualmente somente o CodeDeploy agente da instância local, consulteGerenciando as operações CodeDeploy do agente.

Use o AWS CLI para chamar o comando de cancelamento de registro, especificando:

- O nome que identifica de forma exclusiva a instância local para CodeDeploy (com a opção). `--instance-name`

- Opcionalmente, se deseja excluir o usuário do IAM associado à instância on-premises. O comportamento padrão é excluir o usuário do IAM. Se não quiser excluir o usuário do IAM associado à instância no local, especifique a opção `--no-delete-iam-user` no comando.
- Opcionalmente, a AWS região em que a instância local foi registrada CodeDeploy (com a `--region` opção). Essa deve ser uma das regiões suportadas listadas em Região e os endpoints no Referência geral da AWS(por exemplo,`us-west-2`). Se essa opção não for especificada, a AWS região padrão associada ao usuário do IAM chamador será usada.

Um exemplo que cancela o registro de uma instância e exclui o usuário:

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Um exemplo que cancela o registro de uma instância e não exclui o usuário:

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region
 us-west-2
```

O comando deregister faz o seguinte:

1. Cancela o registro da instância local com. CodeDeploy
2. Se especificado, exclui o usuário do IAM associado à instância on-premises.

Depois de cancelar o registro de uma instância on-premises:

- Ela não é mais exibida no console imediatamente.
- Você pode criar outra instância com o mesmo nome imediatamente.

Se esse comando encontrar erros, será exibida uma mensagem de erro descrevendo como você pode concluir manualmente as etapas restantes. Caso contrário, será exibida uma mensagem de sucesso, descrevendo como chamar o comando uninstall.

Cancelar manualmente o registro de uma instância on-premises

Normalmente, você cancela o registro de uma instância on-premises quando não planeja mais implantá-la. Você usa o AWS CLI para cancelar manualmente o registro de instâncias locais.

O cancelamento manual do registro de uma instância local não desinstala o agente. CodeDeploy Ele não remove o arquivo de configuração da instância. Ele não exclui o usuário do IAM associado à instância. Ele não remove tags associadas à instância.

Para desinstalar automaticamente o CodeDeploy agente e remover o arquivo de configuração da instância local, consulteDesinstale automaticamente o CodeDeploy agente e remova o arquivo de configuração de uma instância local.

Para desinstalar manualmente somente o CodeDeploy agente, consulteGerenciando as operações CodeDeploy do agente.

Para excluir manualmente o usuário do IAM associado, consulte Excluir um usuário do IAM da sua conta da AWS.

Para remover manualmente apenas as tags de instância local associadas, consulte Remover manualmente tags de instância local de uma instância on-premises.

- Chame o comando deregister-on-premises-instance, especificando o nome que identifica exclusivamente a instância on-premises (com a opção `--instance-name`):

  ```
  aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
  ```

  Depois de cancelar o registro de uma instância on-premises:

  - Ela não é mais exibida no console imediatamente.
  - Você pode criar outra instância com o mesmo nome imediatamente.

Veja os detalhes da instância com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre as instâncias usadas em uma implantação.

Para obter informações sobre o uso de ações de CodeDeploy API para visualizar instâncias GetDeploymentInstanceListDeploymentInstances, consulte, ListOnPremisesInstancese.

Tópicos

- Exibir detalhes da instância (console)
- Exibir detalhes da instância (CLI)

Exibir detalhes da instância (console)

Para visualizar os detalhes da instância:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Implantações.

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

3. Para visualizar os detalhes da implantação, escolha o ID de implantação da instância.
4. Visualize todas as instâncias na seção Instance activity (Atividade da instância) da página de implantação.
5. Para ver informações sobre eventos de ciclo de vida de implantação individuais para uma instância, na página de detalhes da implantação, na coluna Eventos, escolha Exibir eventos.

   > **Note**
   >
   > Se Falha for exibido para qualquer um dos eventos de ciclo de vida, na página de detalhes da instância, escolha Exibir logs, Exibir no EC2 ou ambos. Você pode encontrar dicas de solução de problemas em Solução de problemas de instância.

6. Se quiser ver mais informações sobre uma instância do Amazon EC2, selecione o ID da instância na coluna ID da instância.

Exibir detalhes da instância (CLI)

Para usar o AWS CLI para ver os detalhes da instância, chame o `get-deployment-instance` comando ou o `list-deployment-instances` comando.

Para visualizar detalhes sobre uma única instância, chame o comando get-deployment-instance, especificando:

- O ID de implantação exclusivo. Para obter o ID de implantação, chame o comando list-deployments.
- O ID de instância exclusivo. Para obter o ID de instância, chame o comando list-deployment-instances.

Para ver uma lista de IDs para instâncias usadas em uma implantação, chame o comando list-deployment-instances, especificando:

- O ID de implantação exclusivo. Para obter o ID de implantação, chame o comando list-deployments.
- Opcionalmente, se você deseja incluir apenas IDs de instâncias específicos por seu status de implantação. (Se não for especificado, todos os IDs de instância correspondentes serão listados, independentemente do status de implantação.)

CodeDeploy integridade da instância

CodeDeploy monitora o status de integridade das instâncias em um grupo de implantação. Ele reprovará as implantações se o número de instâncias íntegras cair abaixo do número mínimo de instâncias íntegras que foram especificadas para o grupo de implantação durante uma implantação. Por exemplo, se 85% das instâncias precisarem permanecer íntegras durante uma implantação e o grupo de implantação contiver 10 instâncias, a implantação geral falhará se a implementação em uma única instância falhar. Isso ocorre porque quando uma instância é colocada off-line para que a revisão mais recente do aplicativo possa ser instalada, a contagem de instâncias saudáveis disponíveis já cai para 90%. Uma instância com falha mais outra instância offline significaria que apenas 80% das instâncias estão íntegras e disponíveis. O CodeDeploy falharia a implantação geral.

É importante recordar que a implantação total terá sucesso se os itens a seguir forem alcançados:

- CodeDeploy é capaz de implantar em cada instância na implantação.

- A implantação em pelo menos uma instância precisa ser bem-sucedida. Isso significa que, mesmo se o valor mínimo de hosts íntegros for 0, a implantação em pelo menos uma instância deve ser bem-sucedida, ou seja, no mínimo uma instância precisa estar íntegra. Dessa maneira, a implantação geral será bem-sucedida.

Tópicos

- Status de integridade
- Sobre o número mínimo de instâncias íntegras
- Sobre o número mínimo de instâncias íntegras por zona de disponibilidade.

Status de integridade

CodeDeploy atribui dois valores de status de integridade a cada instância: integridade da revisão e integridade da instância.

Integridade da revisão

A integridade da revisão se baseia na revisão de aplicativo atualmente instalada na instância. Ela tem os seguintes valores de status:

- Current: a revisão instalada na instância corresponde à revisão da última implantação bem-sucedida do grupo de implantação.
- Old: a revisão instalada na instância corresponde a uma versão mais antiga do aplicativo.
- Unknown: a revisão de aplicativo não foi instalada com sucesso na instância.

Integridade da instância

A integridade da instância se baseia em se as implementações em uma foram ou não bem-sucedidas. Ela tem os seguintes valores:

- Healthy: a última implantação na instância foi bem-sucedida.
- Unhealthy: a tentativa de implantar uma revisão na instância falhou, ou uma revisão ainda não foi implantada na instância.

CodeDeploy usa a integridade da revisão e a integridade da instância para agendar a implantação nas instâncias do grupo de implantação na seguinte ordem:

1. Integridade de instância - não íntegra.

2. Integridade da revisão - desconhecida.
3. Integridade da revisão - antiga.
4. Integridade da revisão - atual.

Se a implantação geral for bem-sucedida, a revisão será atualizada e os valores do status da integridade do grupo de implantação são atualizados para refletir a implantação mais recente.

- Todas as instâncias atuais que tiveram uma implantação bem-sucedida permanecerão atuais. Caso contrário, elas se tornarão desconhecidas.
- Todas as instâncias antigas ou desconhecidas que tiveram uma implantação bem-sucedida se tornarão atuais. Caso contrário, eles permanecerão antigas ou desconhecidas.
- Todas as instâncias íntegras que tiveram uma implantação bem-sucedida permanecerão íntegras. Caso contrário, elas se tornarão não íntegras.
- Todas as instâncias não íntegras que tiveram uma implantação bem-sucedida se tornarão íntegras. Caso contrário, eles permanecerão não íntegras.

Se a implantação geral falhar ou for interrompida:

- Cada instância na qual CodeDeploy tentou implantar a revisão do aplicativo tem sua integridade definida como íntegra ou não íntegra, dependendo se a tentativa de implantação dessa instância foi bem-sucedida ou falhou.
- Cada instância na qual CodeDeploy não tentou implantar a revisão do aplicativo mantém seu valor atual de integridade da instância.
- A revisão do grupo de implantação permanecerá a mesma.

Sobre o número mínimo de instâncias íntegras

O número mínimo necessário de instâncias íntegras é definido como parte de uma configuração de implantação.

> **Important**
>
> Durante uma blue/green implantação, a configuração de implantação e o valor mínimo de hosts íntegros se aplicam às instâncias no ambiente substituto, não às do ambiente original. No entanto, quando as instâncias no ambiente original forem canceladas do balanceador de

carga, a implantação geral será marcada como falha se o registro de até mesmo uma única instância original não for cancelado com sucesso.

CodeDeploy fornece três configurações de implantação padrão que normalmente usam valores mínimos de host íntegro:

Nome da configuração de implantação padrão	Valor mínimo de hosts íntegros predefinido
CodeDeployDefault.OneAtATime	1
CodeDeployDefault.HalfAtATime	50%
CodeDeployDefault.AllAtOnce	0

Você encontrará mais informações sobre configurações de implantação padrão em Trabalhando com configurações de implantação em CodeDeploy.

Você pode criar configurações de implantação personalizadas CodeDeploy para definir seus próprios valores mínimos de host íntegro. Você pode definir esses valores como números inteiros ou porcentagens usando as seguintes operações:

- Como `minimum-healthy-hosts`, quando você usa o comando create-deployment-config na AWS CLI.
- Como `Value` no tipo de MinimumHealthyHostsdados na CodeDeploy API.
- Como `MinimumHealthyHosts` quando você usa AWS::CodeDeploy:: DeploymentConfig em um CloudFormation modelo.

CodeDeploy permite que você especifique um número mínimo de instâncias íntegras para a implantação com duas finalidades principais:

- Para determinar se a implantação geral foi bem-sucedida ou falhou. A implantação será bem-sucedida se a revisão de aplicativo tiver sido implantada com sucesso pelo menos no número mínimo de instâncias íntegras.
- Para determinar o número de instâncias que devem ser íntegras durante uma implantação para permitir que essa implantação continue.

Você pode especificar o número mínimo de instâncias íntegras para o seu grupo de implantação como um número de instâncias ou como uma porcentagem do número total de instâncias. Se você especificar uma porcentagem, no início da implantação, CodeDeploy converterá a porcentagem no número equivalente de instâncias, arredondando todas as instâncias fracionárias.

CodeDeploy rastreia o status de integridade das instâncias do grupo de implantação durante o processo de implantação e usa o número mínimo especificado de instâncias íntegras da implantação para determinar se a implantação deve continuar. O princípio básico é que uma implantação nunca deve fazer com que o número de instâncias íntegras caia abaixo do número mínimo que você especificou. A única exceção a essa regra é quando um grupo de implantação tem inicialmente menos do que o número mínimo especificado de instâncias íntegras. Nesse caso, o processo de implantação não reduz ainda mais o número de instâncias íntegras.

> **Note**
>
> CodeDeploy tentará implantar em todas as instâncias em um grupo de implantação, mesmo aquelas que estão atualmente em um estado Parado. No cálculo do mínimo de hosts íntegros, uma instância interrompida tem o mesmo impacto que uma instância com falha. Para resolver falhas de implantação devido a muitas instâncias interrompidas, reinicie instâncias ou altere suas tags para excluí-las do grupo de implantação.

CodeDeploy inicia o processo de implantação tentando implantar a revisão do aplicativo nas instâncias não íntegras do grupo de implantação. Para cada implantação bem-sucedida, CodeDeploy altera o status de integridade da instância para íntegro e o adiciona às instâncias íntegras do grupo de implantação. CodeDeploy em seguida, compara o número atual de instâncias íntegras com o número mínimo especificado de instâncias íntegras.

- Se o número de instâncias íntegras for menor ou igual ao número mínimo especificado de instâncias íntegras, CodeDeploy cancela a implantação para garantir que o número de instâncias íntegras não diminua com mais implantações.
- Se o número de instâncias íntegras for maior que o número mínimo especificado de instâncias íntegras em pelo menos uma, CodeDeploy implanta a revisão do aplicativo no conjunto original de instâncias íntegras.

Se uma implantação em uma instância íntegra falhar, CodeDeploy alterará o status de integridade dessa instância para não íntegra. Conforme a implantação progride, CodeDeploy atualiza o número

atual de instâncias íntegras e o compara com o número mínimo especificado de instâncias íntegras. Se o número de instâncias íntegras cair para o número mínimo especificado em qualquer ponto do processo de implantação, a implantação CodeDeploy será interrompida. Essa prática impede a possibilidade de falha da próxima implantação, diminuindo o número de instâncias íntegrasabaixo do mínimo especificado.

> **Note**
>
> Certifique-se de que o número mínimo de instâncias íntegras que você especificar seja menor que o número total de instâncias no grupo de implantação. Se você especificar um valor percentual, lembre-se de que ele será arredondado. Caso contrário, quando a implantação for iniciada, o número de instâncias íntegras já será menor que ou igual ao número mínimo especificado de instâncias íntegras, e o CodeDeploy reprovará imediatamente a implantação geral.

CodeDeploy também usa o número mínimo especificado de instâncias íntegras e o número real de instâncias íntegras para determinar se e como implantar a revisão do aplicativo em várias instâncias. Por padrão, CodeDeploy implanta a revisão do aplicativo no maior número possível de instâncias sem nenhum risco de que o número de instâncias íntegras fique abaixo do número mínimo especificado de instâncias íntegras.

Para determinar o número de instâncias que devem ser implantadas de uma só vez, CodeDeploy use o seguinte cálculo:

```
[total-hosts] - [minimum-healthy-hosts] =
        [number-of-hosts-to-deploy-to-at-once]
```

Por exemplo:

- Se seu grupo de implantação tiver 10 instâncias e você definir o número mínimo de instâncias íntegras como 9, CodeDeploy implanta em 1 instância por vez.
- Se seu grupo de implantação tiver 10 instâncias e você definir o número mínimo de instâncias íntegras como 3, CodeDeploy implanta em 7 instâncias ao mesmo tempo no primeiro lote e depois nas 3 restantes no segundo lote.
- Se seu grupo de implantação tiver 10 instâncias e você definir o número mínimo de instâncias íntegras como 0, CodeDeploy implanta em 10 instâncias ao mesmo tempo.

Exemplos

Os exemplos a seguir supõem um grupo de implantação com 10 instâncias.

Instâncias mínimas íntegras: 95%

CodeDeploy arredonda o número mínimo de instâncias íntegras para até 10 instâncias, o que equivale ao número de instâncias íntegras. A implantação global falhará imediatamente sem implantar a revisão em uma instância.

Instâncias mínimas íntegras: 9

CodeDeploy implanta a revisão em uma instância por vez. Se a implantação em qualquer uma das instâncias falhar, a implantação geral falhará CodeDeploy imediatamente, pois o número de instâncias íntegras é igual ao número mínimo de instâncias íntegras. A exceção a essa regra é que, se a última instância falhar, a implantação ainda será bem-sucedida.

CodeDeploy continua a implantação, uma instância por vez, até que qualquer implantação falhe ou a implantação geral seja concluída. Se todas as 10 implantações forem bem-sucedidas, o grupo de implantação tem 10 instâncias íntegras.

Instâncias mínimas íntegras: 8

CodeDeploy implanta a revisão em duas instâncias por vez. Se duas dessas implantações falharem, a implantação geral falhará CodeDeploy imediatamente. A exceção a essa regra é que, se a última instância for a segunda a falhar, a implantação ainda será bem-sucedida.

Instâncias mínimas íntegras: 0

CodeDeploy implanta a revisão em todo o grupo de implantação de uma só vez. A menos uma implantação a uma instância deve suceder para implantação total suceda. Se 0 instâncias estiverem íntegras, então a implantação falhará. Isso ocorre porque, para marcar uma implantação geral como bem-sucedida, no mínimo uma instância precisa estar íntegra quando a implantação geral for concluída, mesmo que o valor mínimo de instâncias íntegras seja 0.

Sobre o número mínimo de instâncias íntegras por zona de disponibilidade.

> **Note**
>
> Esta seção utiliza os termos instância e host de forma intercambiável para se referir às instâncias do Amazon EC2.

Se você estiver implantando em instâncias em várias zonas de disponibilidade, você pode habilitar opcionalmente o zonal configuration recurso, que permite CodeDeploy implantar em uma zona de disponibilidade por vez.

Quando esse recurso estiver ativado, CodeDeploy garantirá que o número de hospedeiros saudáveis permaneça acima dos valores de “hospedeiros saudáveis mínimos por zona” e dos valores de “hospedeiros saudáveis mínimos”. Se o número de hosts íntegros ficar abaixo de qualquer um dos valores, a implantação CodeDeploy falhará em todas as zonas de disponibilidade.

Para calcular o número de hosts nos quais implantar de uma só vez, CodeDeploy use os valores de “hosts íntegros mínimos por zona” e “hosts íntegros mínimos”. CodeDeploy usará o menor dos cálculos `[A]` e`[B]`, onde `[A]` e onde `[B]` estão:

```
[A] = [total-hosts] - [min-healthy-hosts] =
        [number-of-hosts-to-deploy-to-at-once]
```

```
[B] = [total-hosts-per-AZ] - [min-healthy-hosts-per-AZ] =
        [number-of-hosts-to-deploy-to-at-once-per-AZ]
```

Depois de determinar o número de hosts nos quais implantar de uma vez, CodeDeploy implanta nos hosts em lotes desse número, uma zona de disponibilidade por vez, com uma pausa opcional (ou “tempo de cozimento”) entre as zonas.

Exemplo

Se a implantação estiver configurada como:

- `[total-hosts]` é `200`
- `[minimum-healthy-hosts]` é `160`
- `[total-hosts-per-AZ]` é `100`
- `[minimum-healthy-hosts-per-AZ]` é `50`

Então...

- `[A]` = `200 - 160 = 40`
- `[B]` = `100 - 50 = 50`
- `40` é menor que `50`

Portanto, CodeDeploy será implantado nos 40 hosts de uma só vez.

Nesse cenário, a implantação ocorre da seguinte forma:

1. CodeDeploy é implantado na primeira zona de disponibilidade:
   a. CodeDeploy é implantado nos primeiros 40 hosts.
   b. CodeDeploy é implantado nos próximos 40 hosts.
   c. CodeDeploy é implantado nos 20 hosts restantes.

   A implantação na primeira zona de disponibilidade agora está concluída.
2. (Opcional) CodeDeploy aguarda enquanto a implantação na primeira zona “cozinha”, conforme definido pela configuração Duração do monitor ou Adicionar uma duração do monitor para a primeira zona. Se não houver problemas, CodeDeploy continua.
3. CodeDeploy é implantado na segunda zona de disponibilidade:
   a. CodeDeploy é implantado nos primeiros 40 hosts.
   b. CodeDeploy é implantado nos próximos 40 hosts.
   c. CodeDeploy é implantado nos 20 hosts restantes.

   A implantação na segunda zona de disponibilidade final agora está concluída.

Para saber mais sobre o recurso de configuração de zonas e como especificar o número mínimo de instâncias íntegras por zona de disponibilidade, consulte zonal configuration.

Trabalhando com configurações de implantação em CodeDeploy

Uma configuração de implantação é um conjunto de regras e condições de sucesso e falha que é usado pelo CodeDeploy durante uma implantação. Essas regras e condições são diferentes, dependendo se você implementa em uma plataforma de EC2/On-Premises computação, plataforma de computação AWS Lambda ou plataforma de computação Amazon ECS.

Configurações de implantação em uma plataforma de EC2/on-premises computação

Quando você implanta em uma plataforma EC2/On-Premises computacional, a configuração de implantação especifica, por meio do uso de um valor de “hosts íntegros mínimos” e um valor opcional de “hosts íntegros mínimos por zona”, o número ou a porcentagem de instâncias que devem permanecer disponíveis a qualquer momento durante a implantação.

Você pode usar uma das três configurações de implantação predefinidas fornecidas por AWS ou criar uma configuração de implantação personalizada. Para obter mais informações sobre como criar configurações de implantação personalizadas, consulte Create a Deployment Configuration. Se você não especificar uma configuração de implantação, CodeDeploy use a configuração CodeDeployDefault.OneAtATime de implantação.

Para obter mais informações sobre como CodeDeploy monitora e avalia a integridade da instância durante uma implantação, consulteInstance Health. Para ver uma lista das configurações de implantação já registradas em sua AWS conta, consulteView Deployment Configuration Details.

Configurações de implantação predefinidas para uma plataforma de computação EC2/on-premises

A tabela a seguir lista as configurações de implantação predefinidas.

> **Note**
>
> Não há configurações de implantação predefinidas compatíveis com o recurso zonal configuration (que permite especificar o número de hosts íntegros por zona de

disponibilidade). Se quiser utilizar esse recurso, crie a sua própria configuração de implantação.

Configuração de implantação	Description
CodeDeployDefault.AllAtOnce	In-place implantações: Tenta implantar uma revisão de aplicativo em quantas instâncias for possível de uma só vez. O status da implantação geral é exibido como Succeeded (Bem-sucedido) se a revisão do aplicativo é implantada em uma ou mais das instâncias. O status da implantação geral é exibido como Failed (Com falha) se a revisão de aplicativo não é implantada em nenhuma das instâncias. Usando um exemplo de nove instâncias, CodeDeployDefault.AllAtOnce tenta implantar em todas as nove instâncias de uma só vez. A implantação geral será bem-suced ida se a implantação em até mesmo em uma única instância for bem-sucedida. Ele falhará somente se as implementações nas nove instâncias falharem. Blue/green implantações: • Implantação em ambiente de substituição: segue as mesmas CodeDeployDefault. AllAtOnce regras de implantação das implantações no local. • Redirecionamento do tráfego: direciona o tráfego para todas as instâncias no ambiente de substituição ao mesmo tempo. Será bem-sucedido se o tráfego for redirecionado com êxito para pelo menos uma instância.

Configuração de implantação	Description
	Falhará se o redirecionamento para todas as instâncias falhar.

Configuração de implantação	Description
CodeDeployDefault.HalfAtATime	In-place implantações: Implementa em até metade das instâncias de cada vez (com frações arredondadas para baixo). A implantação geral será bem-sucedida se a revisão de aplicativo for implantada em pelo menos metade das instâncias (com frações arredondadas). Caso contrário, ocorrerá uma falha na implantação. No exemplo de nove instâncias, ele implanta em até quatro instâncias de cada vez. A implantação geral será bem-sucedida se for feita com êxito em cinco ou mais instâncias. Caso contrário, ocorrerá uma falha na implantação. **Note** Se você estiver implantando em instâncias em vários grupos de Auto Scaling CodeDeploy , implantará até metade das instâncias por vez, independentemente do grupo de Auto Scaling em que estejam. Por exemplo, vamos supor que você tenha dois grupos do Auto Scaling, `ASG1` e `ASG2`, cada um com 10 instâncias. Nesse cenário, CodeDeploy pode implantar em 10 instâncias em apenas `ASG1` e considerar isso um sucesso porque foi implantado em pelo menos metade das instâncias. Blue/green implantações:

Configuração de implantação	Description
	• Implantação em ambiente de substituição: segue as mesmas CodeDeployDefault. HalfAtATime regras de implantação das implantações no local. • Redirecionamento do tráfego: direciona o tráfego para metade das instâncias no ambiente de substituição de cada vez. Será bem-sucedido se o redirecionamento para pelo menos metade das instâncias for bem-sucedido. Caso contrário, falha.

Configuração de implantação	Description
CodeDeployDefault.OneAtATime	In-place implantações: Implanta a revisão de aplicativo em apenas uma instância de cada vez. Para grupos de implantação que contêm mais de uma instância: • A implantação geral será bem-sucedida se a revisão de aplicativo for implantada em todas as instâncias. A exceção a essa regra é que, se ocorrer uma falha na implantação da última instância, a implantação geral ainda será bem-sucedida. Isso ocorre porque CodeDeploy permite que somente uma instância por vez seja colocada off-line com a CodeDeployDefault.OneAtATime configuração. • A implantação geral falhará se a revisão de aplicativo não conseguir ser implantada em qualquer outra instância que não seja a última. • Em um exemplo usando nove instâncias, ele implanta em uma instância de cada vez. A implantação geral será bem-sucedida se a implantação nas primeiras oito instâncias for bem-sucedida. Ocorrerá uma falha na implantação geral se ocorrer uma falha em qualquer uma das primeiras oito instâncias. Para grupos de implantação que contêm apenas uma instância, a implantação geral apenas será bem-sucedida se a implantação na instância única for bem-sucedida.

Configuração de implantação	Description
	Blue/green implantações: • Implantação em ambiente de substituição: segue as mesmas regras CodeDeployDefault.OneAtATime de implantação das implantações locais. • Redirecionamento do tráfego: direciona o tráfego para uma instância no ambiente de substituição de cada vez. Será bem-sucedido se o tráfego for redirecionado com êxito para todas as instâncias de substituição. Falhará após a primeira falha de redirecionamento. A exceção a essa regra é que, se ocorrer uma falha no registro da última instância, a implantação geral ainda será bem-sucedida.

Configurações de implantação em uma plataforma de computação do Amazon ECS

Quando você implanta em uma plataforma de computação do Amazon ECS, a configuração de implantação especifica como o tráfego é deslocado para o conjunto de tarefas do Amazon ECS atualizado. Você pode mudar o tráfego usando uma configuração de implantação canária, linear ou tudo de uma vez. Para obter mais informações, consulte Configuração de implantação.

Você também pode criar sua própria configuração de implantação linear ou canary personalizada. Para obter mais informações, consulte Create a Deployment Configuration.

Configurações de implantação predefinidas para uma plataforma de computação do Amazon ECS

A tabela a seguir lista as configurações predefinidas disponíveis para implantações do Amazon ECS.

> **Note**
>
> Quando um Network Load Balancer é usado, somente a configuração de implantação `CodeDeployDefault.ECSAllAtOnce` é permitida.

Configuração de implantação	Description
CodeDeployDefault.ECSLinear10PercentEvery1Minutes	Desloca 10 por cento do tráfego a cada minuto até que todo o tráfego seja deslocado.
CodeDeployDefault.ECSLinear10PercentEvery3Minutes	Desloca 10 por cento do tráfego a cada três minutos até que todo o tráfego seja deslocado.
CodeDeployDefault.ECSCanary10Percent5Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90% restantes são implantados cinco minutos depois.
CodeDeployDefault.ECSCanary10Percent15Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90% restantes são implantados 15 minutos depois.
CodeDeployDefault.ECSAllAtOnce	Desloca todo o tráfego para o contêiner atualizado do Amazon ECS de uma só vez.

Configurações de implantação para CloudFormation blue/green implantações (Amazon ECS)

Quando você implanta em uma plataforma computacional do Amazon ECS por meio de CloudFormation blue/green implantações, a configuração de implantação especifica como o tráfego é transferido para o contêiner atualizado do Amazon ECS. Você pode mudar o tráfego usando uma configuração de implantação canária, linear ou tudo de uma vez. Para obter mais informações, consulte Configuração de implantação.

Com CloudFormation blue/green implantações, você não pode criar sua própria configuração personalizada de implantação canária ou linear. Para obter instruções passo a passo sobre como usar CloudFormation para gerenciar suas implantações do Amazon ECS, consulte Automatizar

blue/green implantações do ECS por meio do uso no Guia do usuário blue/green . CodeDeploy CloudFormationCloudFormation

> **Note**
>
> O gerenciamento de blue/green implantações do Amazon ECS com não CloudFormation está disponível nas regiões da Europa (Milão), África (Cidade do Cabo) e Ásia-Pacífico (Osaka).

Configurações de implantação em um AWS Lambda Plataforma de computação

Quando você implanta em uma plataforma de computação AWS Lambda, a configuração de implantação especifica a forma como o tráfego é transferido para as novas versões da função Lambda em seu aplicativo. Você pode mudar o tráfego usando uma configuração de implantação canária, linear ou tudo de uma vez. Para obter mais informações, consulte Configuração de implantação.

Você também pode criar sua própria configuração de implantação linear ou canary personalizada. Para obter mais informações, consulte Create a Deployment Configuration.

Configurações de implantação predefinidas para um AWS Lambda Plataforma de computação

A tabela a seguir lista as configurações predefinidas disponíveis para implantações do AWS Lambda .

Configuração de implantação	Description
CodeDeployDefault.LambdaCanary10Percent5Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90% restantes são implantados cinco minutos depois.
CodeDeployDefault.LambdaCanary10Percent10Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90 por cento restantes são implantados 10 minutos depois.

Configuração de implantação	Description
CodeDeployDefault.LambdaCanary10Percent15Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90% restantes são implantados 15 minutos depois.
CodeDeployDefault.LambdaCanary10Percent30Minutes	Desloca 10 por cento do tráfego no primeiro incremento. Os 90 por cento restantes são implantados 30 minutos depois.
CodeDeployDefault.LambdaLinear10PercentEvery1Minute	Desloca 10 por cento do tráfego a cada minuto até que todo o tráfego seja deslocado.
CodeDeployDefault.LambdaLinear10PercentEvery2Minutes	Desloca 10 por cento do tráfego a cada dois minutos até que todo o tráfego seja deslocado.
CodeDeployDefault.LambdaLinear10PercentEvery3Minutes	Desloca 10 por cento do tráfego a cada três minutos até que todo o tráfego seja deslocado.
CodeDeployDefault.LambdaLinear10PercentEvery10Minutes	Desloca 10 por cento do tráfego a cada 10 minutos até que todo o tráfego seja deslocado.
CodeDeployDefault.LambdaAllAtOnce	Desloca todo o tráfego para as funções do Lambda atualizadas de uma única vez.

Tópicos

- Create a Deployment Configuration
- View Deployment Configuration Details
- Delete a Deployment Configuration

Crie uma configuração de implantação com CodeDeploy

Se você não quiser usar uma das configurações de implantação padrão fornecidas CodeDeploy, você pode criar sua própria usando as instruções a seguir.

Você pode usar o CodeDeploy console AWS CLI, as CodeDeploy APIs ou um CloudFormation modelo para criar configurações de implantação personalizadas.

Para obter informações sobre como usar um CloudFormation modelo para criar uma configuração de implantação, consulteCloudFormation modelos para CodeDeploy referência.

Tópicos

- Criar uma configuração de implantação (console)
- Criando uma configuração de implantação com CodeDeploy (AWS CLI)

Criar uma configuração de implantação (console)

Utilize as instruções a seguir para criar uma configuração de implantação utilizando o console da AWS .

Para criar uma configuração de implantação CodeDeploy usando o console

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, escolha Configurações de implantação.

   Uma lista de configurações de implantação integradas é exibida.
3. Escolha Criar configuração de implantação.
4. Em Nome da configuração de implantação, digite um nome para a configuração de implantação. Por exemplo, .**`my-deployment-config`**
5. Em Plataforma de computação, escolha uma das seguintes opções:

   - EC2/On-premises
   - AWS Lambda
   - Amazon ECS
6. Execute um destes procedimentos:

- Se você escolheu EC2/On-premises:

  1. Em Mínimo de hosts íntegros, especifique o número ou o percentual de instâncias que devem estar disponíveis durante todo o tempo de uma implantação. Para obter mais informações sobre como o CodeDeploy monitora e avalia a integridade da instância durante uma implantação, consulte Instance Health.
  2. (Opcional) Em Configuração zonal, selecione Ativar configuração zonal para CodeDeploy implantar seu aplicativo em uma zona de disponibilidade por vez, dentro de uma AWS região. Ao implantar em uma zona de disponibilidade por vez, é possível expor a implantação a um público cada vez maior à medida que aumenta a confiança no desempenho e na viabilidade da implantação. Se você não habilitar uma configuração zonal, CodeDeploy implanta seu aplicativo em uma seleção aleatória de hosts em uma região.

     Se você ativar o recurso de configuração de zona, observe o seguinte:

     - O recurso de configuração zonal só é suportado com implantações locais em instâncias do Amazon EC2. (Blue/greenimplantações e instâncias locais não são suportadas.) Para obter mais informações sobre implantações no local, consulte Tipo de implantação.
     - O recurso de configuração de zona não é compatível com Configurações de implantação predefinidas. Para usar uma configuração de zona, crie uma configuração de implantação personalizada, conforme descrito aqui.
     - Se CodeDeploy precisar reverter uma implantação, CodeDeploy executará as operações de reversão em hosts aleatórios. (não CodeDeploy reverterá uma zona por vez, como você poderia esperar.) Esse comportamento de reversão foi escolhido por motivos de desempenho. Para obter mais informações sobre reversões, consulte Reimplante e reverta uma implantação com CodeDeploy.
  3. Se você marcou a caixa de seleção Ativar configuração de zona, opcionalmente, especifique as seguintes opções:

     - (Opcional) Em Duração do monitor, especifique o período de tempo, em segundos, que CodeDeploy deve esperar após a conclusão da implantação em uma zona de disponibilidade. CodeDeploy aguardará esse tempo antes de iniciar uma implantação na próxima zona de disponibilidade. Considere adicionar uma duração de monitoramento para dar à implantação algum tempo para se provar (ou de “incorporação”) em uma zona de disponibilidade antes de ser liberada na próxima zona. Se você não especificar a duração do monitor, CodeDeploy iniciará imediatamente a implantação na próxima zona de disponibilidade. Para obter mais informações sobre como a configuração da Duração

do monitoramento funciona, consulte Sobre o número mínimo de instâncias íntegras por zona de disponibilidade..

- (Opcional) Selecione Adicionar uma duração do monitoramento para a primeira zona para definir uma duração de monitoramento que se aplique somente à primeira zona de disponibilidade. Você pode definir essa opção se quiser permitir um tempo de incorporação extra para a primeira zona de disponibilidade. Se você não especificar um valor em Adicionar uma duração do monitor da primeira zona, CodeDeploy use o valor da duração do monitor para a primeira zona de disponibilidade.
- (Opcional) Em Mínimo de hosts íntegros por zona, especifique o número ou o percentual de instâncias que devem estar disponíveis por zona de disponibilidade durante uma implantação. Escolha FLEET_PERCENT para especificar uma porcentagem ou HOST_COUNT para especificar um número. Esse campo funciona em conjunto com o campo Mínimo de hosts íntegros. Para obter mais informações, consulte Sobre o número mínimo de instâncias íntegras por zona de disponibilidade..

  Se você não especificar um valor em Mínimo de hosts íntegros por zona, CodeDeploy usará um valor padrão de `0` porcentagem.

- Se você escolher AWS Lambda ou Amazon ECS:

  1. Em Tipo, escolha Linear ou Canário.
  2. Nos campos Etapa e Intervalo, faça o seguinte:

     - Se você escolher Canário, para Etapa, digite uma porcentagem do tráfego, entre 1 e 99, que será deslocada. Essa é a porcentagem do tráfego que será deslocada no primeiro incremento. O tráfego restante é deslocado após o intervalo selecionado no segundo incremento.

       Em Intervalo, insira o número de minutos entre o primeiro e o segundo deslocamento de tráfego.

     - Se você escolher Linear, para Etapa, digite uma porcentagem do tráfego, entre 1 e 99, que será deslocada. Essa será a porcentagem do tráfego que será deslocada no início de cada intervalo.

       Em Intervalo, insira o número de minutos entre cada deslocamento incremental.

7. Escolha Criar configuração de implantação.

Agora você tem uma configuração de implantação que pode associar a um grupo de implantação.

Criando uma configuração de implantação com CodeDeploy (AWS CLI)

Para usar o AWS CLI para criar uma configuração de implantação, chame o comando create-deployment-config.

O exemplo a seguir cria uma configuração de EC2/On-Premises implantação chamada `ThreeQuartersHealthy` que exige que 75% das instâncias de destino permaneçam íntegras durante uma implantação:

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --
minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

O exemplo a seguir cria uma configuração de EC2/On-Premises implantação chamada `300Total50PerAZ` que exige que 300 instâncias de destino permaneçam íntegras no total por implantação e 50 permaneçam íntegras por zona de disponibilidade. Ele também define uma duração do monitoramento de uma hora.

```
aws deploy create-deployment-config --deployment-config-name 300Total50PerAZ
 --minimum-healthy-hosts type=HOST_COUNT,value=300 --zonal-config
 '{"monitorDurationInSeconds":3600,"minimumHealthyHostsPerZone":
{"type":"HOST_COUNT","value":50}}'
```

O exemplo a seguir cria uma configuração de implantação do AWS Lambda chamada. `Canary25Percent45Minutes` Ela usa o deslocamento de tráfego canário para deslocar 25% do tráfico no primeiro incremento. Os 75 por cento restantes são deslocados para 45 minutos depois:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

O exemplo a seguir cria uma configuração de implantação do Amazon ECS chamada `Canary25Percent45Minutes`. Ela usa o deslocamento de tráfego canário para deslocar 25% do tráfico no primeiro incremento. Os 75 por cento restantes são deslocados para 45 minutos depois:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
```

```
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

Veja os detalhes da configuração de implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre as configurações de implantação associadas à sua AWS conta. Para obter descrições das configurações de CodeDeploy implantação predefinidas, consulte. Configurações de implantação predefinidas para uma plataforma de computação EC2/on-premises

Tópicos

- Visualizar detalhes de configurações de implantação (console)
- Visualizar configuração de implantação (CLI)

Visualizar detalhes de configurações de implantação (console)

Para usar o CodeDeploy console para ver uma lista de nomes de configuração de implantação:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Implantar e escolha Configurações de implantação.

   Aqui você pode ver nomes e critérios de configuração de implantação para cada configuração de implantação.

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

Visualizar configuração de implantação (CLI)

Para usar o AWS CLI para visualizar os detalhes da configuração de implantação, chame o `get-deployment-config` comando ou o `list-deployment-configs` comando.

Para ver detalhes sobre uma única configuração de implantação, chame o comando get-deployment-config, especificando o nome exclusivo da configuração de implantação.

Para visualizar detalhes sobre várias configurações de implantação, chame o comando list-deployments.

Exclua uma configuração de implantação com CodeDeploy

Você pode usar as APIs AWS CLI ou as CodeDeploy APIs para excluir configurações de implantação personalizadas associadas à sua AWS conta. Não é possível excluir configurações de implantação internas, como `CodeDeployDefault.AllAtOnce`, `CodeDeployDefault.HalfAtATime` e `CodeDeployDefault.OneAtATime`.

> **Warning**
>
> Não é possível excluir uma configuração de implantação personalizada que ainda esteja em uso. Se você excluir uma configuração de implantação personalizada não utilizada, não poderá mais associá-la a novas implantações e a novos grupos de implantação. Esta ação não pode ser desfeita.

Para usar o AWS CLI para excluir uma configuração de implantação, chame o comando delete-deployment-config, especificando o nome da configuração de implantação. Para visualizar uma lista de nomes de configuração de implantação, chame o comando list-deployment-configs.

O exemplo a seguir exclui uma configuração de implantação chamada ThreeQuartersHealthy.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Trabalhando com aplicativos em CodeDeploy

Depois de configurar instâncias, mas antes de poder implantar uma revisão, você deve criar um aplicativo no CodeDeploy. Um aplicativo é simplesmente um nome ou contêiner usado pelo CodeDeploy para garantir que a revisão, a configuração da implantação e o grupo de implantação corretos sejam referenciados durante uma implantação.

Use as informações da tabela a seguir nas próximas etapas:

Plataforma de computação	Cenário	Informações para a próxima etapa
EC2/On-Premises	Ainda não criei instâncias.	Consulte Trabalhando com instâncias para CodeDeploy e depois retorne a esta página.
EC2/On-Premises	Eu criei instâncias, mas não terminei de marcá-las.	Consulte Tagging Instances for Deployments e depois retorne a esta página.
EC2/On-Premises, AWS Lambda e Amazon ECS	Ainda não criei um aplicativo.	Consulte Crie um aplicativo com CodeDeploy
EC2/On-Premises, AWS Lambda e Amazon ECS	Já criei um aplicativo, mas não criei um grupo de implantação.	Consulte Crie um grupo de implantação com CodeDeploy.
EC2/On-Premises, AWS Lambda e Amazon ECS	Já criei um aplicativo e um grupo de implantação, mas não criei uma revisão de aplicativo.	Consulte Trabalhando com revisões de aplicativos para CodeDeploy.
EC2/On-Premises, AWS Lambda e Amazon ECS	Já criei um aplicativo e um grupo de implantação e já carreguei minha revisão de aplicativo. Estou pronto para implantar.	Consulte Crie uma implantaç ão com CodeDeploy.

Tópicos

- Crie um aplicativo com CodeDeploy
- Veja os detalhes do aplicativo com CodeDeploy
- Criar uma regra de notificação
- Renomear um aplicativo CodeDeploy
- Excluir um aplicativo no CodeDeploy

Crie um aplicativo com CodeDeploy

Um aplicativo é simplesmente um nome ou contêiner usado CodeDeploy para garantir que a revisão, a configuração de implantação e o grupo de implantação corretos sejam referenciados durante uma implantação. Você pode usar o CodeDeploy console AWS CLI, as CodeDeploy APIs ou um CloudFormation modelo para criar aplicativos.

Seu código, ou revisão do aplicativo, é instalado nas instâncias por meio de um processo chamado implantação. CodeDeploy oferece suporte a dois tipos de implantações:

- In-place implantação: o aplicativo em cada instância do grupo de implantação é interrompido, a revisão mais recente do aplicativo é instalada e a nova versão do aplicativo é iniciada e validada. Você pode usar um balanceador de carga de forma que cada registro de instância é cancelado durante sua implantação e, em seguida, restaurado para o serviço após a conclusão da implantação. Somente implantações que usam a plataforma de EC2/On-Premises computação podem usar implantações no local. Para obter mais informações sobre implantações no local, consulte Visão geral de uma implantação no local.
- Blue/green implantação: o comportamento da sua implantação depende da plataforma de computação que você usa:
  - Blue/green em uma plataforma de EC2/On-Premises computação: as instâncias em um grupo de implantação (o ambiente original) são substituídas por um conjunto diferente de instâncias (o ambiente substituto) usando estas etapas:
    - As instâncias são provisionadas para o ambiente de substituição.
    - A revisão de aplicativo mais recente será instalada nas instâncias de substituição.
    - Um tempo de espera opcional ocorre para atividades como teste de aplicativos e verificação do sistema.
    - As instâncias no ambiente de substituição são registradas com um ou mais balanceadores de carga do Elastic Load Balancing, fazendo com que o tráfego seja redirecionado para elas. O

registro das instâncias no ambiente original é cancelado e pode ser encerrado ou mantido em execução para outros usos.

> **Note**
>
> Se você usa uma plataforma EC2/On-Premises computacional, saiba que as blue/green implantações funcionam somente com instâncias do Amazon EC2.

- Blue/green em uma plataforma computacional AWS Lambda ou Amazon ECS: o tráfego é deslocado em incrementos de acordo com uma configuração de implantação canária, linear ou completa.
- Blue/green implantações por meio de CloudFormation: o tráfego é transferido de seus recursos atuais para seus recursos atualizados como parte de uma atualização de CloudFormation pilha. Atualmente, somente blue/green implantações do ECS são suportadas.

Para obter mais informações sobre blue/green implantações, consulteVisão geral de uma blue/green implantação.

Ao usar o CodeDeploy console para criar um aplicativo, você configura seu primeiro grupo de implantação ao mesmo tempo. Ao usar o AWS CLI para criar um aplicativo, você cria seu primeiro grupo de implantação em uma etapa separada.

Para ver uma lista de aplicativos já registrados em sua AWS conta, consulteVeja os detalhes do aplicativo com CodeDeploy. Para obter informações sobre como usar um CloudFormation modelo para criar um aplicativo, consulteCloudFormation modelos para CodeDeploy referência.

Os dois tipos de implantação não se aplicam a todos os destinos. A tabela a seguir relaciona quais tipos de implantação funcionam com implantações nos três tipos de destinos de implantação.

Destino da implantação	In-place	Blue/green
Amazon EC2	Sim	Sim
On-premises	Sim	Não
Funções Lambda sem servidor AWS	Não	Sim

Destino da implantação	In-place	Blue/green
Aplicativos do Amazon ECS	Não	Sim

Tópicos

- Criar um aplicativo para uma implantação no local (console)
- Criar um aplicativo para uma blue/green implantação (console)
- Criar um aplicativo para uma implantação do serviço do Amazon ECS (console)
- Crie um aplicativo para um AWS Lambda implantação de funções (console)
- Criar um aplicativo (CLI)

Criar um aplicativo para uma implantação no local (console)

Para usar o CodeDeploy console para criar um aplicativo para uma implantação local:

Warning

Não siga essas etapas se:

- Você não preparou suas instâncias para serem usadas em CodeDeploy implantações. Para configurar suas instâncias, siga as instruções em Trabalhando com instâncias para CodeDeploy e depois siga as etapas neste tópico.
- Você deseja criar um aplicativo que usa uma configuração de implantação personalizada, mas ainda não criou essa configuração de implantação. Siga as instruções em Create a Deployment Configuration e depois siga as etapas neste tópico.
- Você não tem uma função de serviço que CodeDeploy confie na confiança e nas permissões mínimas necessárias. Para criar e configurar uma função de serviço com as permissões necessárias, siga as instruções em Etapa 2: criar uma função de serviço para CodeDeploy e depois volte para as etapas deste tópico.
- Você deseja selecionar um Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing para a implantação no local, mas ainda não o criou.

Para criar um aplicativo para uma implantação local usando o CodeDeploy console:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Implantar e escolha Introdução.
3. Selecione Criar aplicativo.
4. Em Nome da aplicação, digite o nome do aplicativo.
5. Em Compute Platform, escolha EC2/On-premises.
6. Escolha Criar aplicação.
7. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação.
8. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

   > **Note**
   >
   > Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação, tags e/ou nomes de grupos do Amazon EC2 Auto Scaling, bem como a configuração de implantação), especifique essas configurações nessa página. Embora esse novo grupo de implantação e o grupo de implantação existente tenham o mesmo nome, eles CodeDeploy são tratados como grupos de implantação separados, pois cada um deles está associado a aplicativos separados.

9. Em Service role (Função de serviço), escolha uma função de serviço que conceda acesso do CodeDeploy à sua instância de destino.
10. Em Tipo de implantação, escolha In-place.
11. Em Configuração do ambiente, selecione uma das opções a seguir:

    a. Grupos do Amazon EC2 Auto Scaling: insira ou escolha o nome de um grupo do Amazon EC2 Auto Scaling no qual implantar a revisão do aplicativo. Quando novas instâncias do Amazon EC2 são lançadas como parte de um grupo do Amazon EC2 Auto Scaling

CodeDeploy , você pode implantar suas revisões nas novas instâncias automaticamente. É possível adicionar até 10 grupos do Amazon EC2 Auto Scaling a um grupo de implantação.

b. Instâncias ou instâncias do Amazon EC2: nos campos Chave e Valor, insira os valores do par chave-valor que você usou para marcar as On-premisesinstâncias. É possível marcar até 10 pares de chave/valor em um único grupo de tags.

   i. Você pode usar caracteres curinga no campo Valor para identificar todas as instâncias marcadas em determinados padrões, como instâncias do Amazon EC2, centros de custo e nomes de grupos semelhantes e assim por diante. Por exemplo, se você escolher Nome no campo Chave e inserir **`GRP-*a`** no campo Valor, CodeDeploy identificará todas as instâncias que se encaixam nesse padrão**`GRP-1a`**, como**`GRP-2a`**, e. **`GRP-XYZ-a`**

   ii. O campo Valor diferencia maiúsculas de minúsculas.

   iii. Para remover um par de chave/valor da lista, escolha o ícone Remove tag (Remover tag).

   Quando CodeDeploy encontra instâncias que correspondem a cada par de valores-chave especificado ou ao nome do grupo Amazon EC2 Auto Scaling, ele exibe o número de instâncias correspondentes. Escolha o número para ver mais informações sobre as instâncias.

   Se você quiser refinar os critérios para as instâncias implantadas, escolha Adicionar grupo de tags para criar um grupo de tags. É possível criar até três grupos de tags com até dez pares chave/valor de cada. Quando você usa grupos de várias tags em um grupo de implantação, apenas as instâncias que são identificadas por todos os grupos de tags são incluídas no grupo de implantação. Isso significa que uma instância deve corresponder pelo menos a uma das tags em cada um dos grupos a serem incluídos no grupo de implantação.

   Para obter mais informações sobre como usar grupos de tags para refinar o grupo de implantação, consulte Tagging Instances for Deployments.

12. Em Deployment settings (Configurações da implantação), escolha uma configuração de implantação para controlar a taxa em que os aplicativos são implantados nas instâncias, por exemplo, uma de cada vez ou todas ao mesmo tempo. Para obter mais informações sobre configurações de implantação, consulte Trabalhando com configurações de implantação em CodeDeploy.

13. (Opcional) No Load Balancer, selecione Ativar balanceamento de carga e, nas listas, selecione os grupos-alvo Classic Load Balancers, Application Load Balancer e Network Load Balancer para gerenciar o tráfego para as instâncias durante a implantação. CodeDeploy Você pode selecionar até 10 Classic Load Balancers e 10 grupos de destino, totalizando 20 itens. Verifique se as instâncias do Amazon EC2 nas quais você deseja implantar estão registradas nos balanceadores de carga (Classic Load Balancers) ou nos grupos de destino (Application Load Balancers e Network Load Balancers) selecionados.

    Durante uma implantação, o registro das instâncias originais dos balanceadores de carga e dos grupos de destino selecionados é cancelado para evitar que o tráfego seja roteado para essas instâncias durante a implantação. Quando a implantação for concluída, cada instância será registrada novamente com todos os Classic Load Balancers e grupos de destino selecionados.

    Para obter mais informações sobre balanceadores de carga para CodeDeploy implantações, consulte. Integrating CodeDeploy with Elastic Load Balancing
14. (Opcional) Expanda Avançado e configure todas as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon ou reversões automáticas.

    Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantação.
15. Selecione Criar grupo de implantação.

A próxima etapa é preparar uma revisão a ser implantada no aplicativo e grupo de implantação. Para instruções, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

Criar um aplicativo para uma blue/green implantação (console)

Para usar o CodeDeploy console para criar um aplicativo para uma blue/green implantação:

> **Note**
>
> Uma implantação na plataforma de computação AWS Lambda é sempre uma implantação. blue/green Não especifique uma opção de tipo de implantação.

> **Warning**
>
> Não siga essas etapas se:
>
> - Você não tem instâncias com o CodeDeploy agente instalado que queira substituir durante o processo blue/green de implantação. Para configurar suas instâncias, siga as instruções em Trabalhando com instâncias para CodeDeploy e depois siga as etapas neste tópico.
> - Você deseja criar um aplicativo que usa uma configuração de implantação personalizada, mas ainda não criou essa configuração de implantação. Siga as instruções em Create a Deployment Configuration e depois siga as etapas neste tópico.
> - Você não tem uma função de serviço que CodeDeploy confie, no mínimo, na confiança e nas permissões descritas emEtapa 2: criar uma função de serviço para CodeDeploy. Para criar e configurar um perfil de serviço, siga as instruções em Etapa 2: criar uma função de serviço para CodeDeploy e depois siga as etapas neste tópico.
> - Você não criou um Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing para o registro das instâncias em seu ambiente substituto. Para obter mais informações, consulte Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda Implantar e escolha Introdução.
3. Em Nome da aplicação, digite o nome do aplicativo.
4. Em Compute platform (Plataforma de computação), selecione EC2/On-Premises.
5. Escolha Criar aplicação.
6. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação.
7. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

> **Note**
>
> Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo as tags do nome do grupo de implantação, os nomes de grupos do Amazon EC2 Auto Scaling e a configuração de implantação), escolha essas configurações nessa página. Embora esse novo grupo de implantação e o grupo de implantação existente tenham o mesmo nome, o CodeDeploy os tratará como grupos de implantação separados, porque cada um deles está associado a um aplicativo separado.

8. Em Service role (Função de serviço), escolha uma função de serviço que conceda acesso do CodeDeploy à sua instância de destino.
9. Em Tipo de implantação, escolha Blue/green.
10. Em Configuração do ambiente, escolha o método que você deseja usar para fornecer instâncias ao seu ambiente de substituição:

    a. Copiar automaticamente o grupo Amazon EC2 Auto Scaling: cria um CodeDeploy grupo Amazon EC2 Auto Scaling copiando um que você especificar.

    b. Fornecer instâncias manualmente: você apenas especificará as instâncias para o seu ambiente de substituição depois de criar uma implantação. Você deve criar as instâncias antes de iniciar a implantação. Em vez disso, aqui, você especifica as instâncias que deseja substituir.

11. Dependendo da sua escolha na etapa 10, siga um destes procedimentos:

    - Se tiver escolhido Cópia automática do grupo do Amazon EC2 Auto Scaling: em Grupo do Amazon EC2 Auto Scaling, escolha ou insira o nome do grupo do Amazon EC2 Auto Scaling que você deseja usar como modelo para o grupo do Amazon EC2 Auto Scaling para as instâncias no seu ambiente de substituição. O número de instâncias atualmente íntegras no grupo do Amazon EC2 Auto Scaling que você selecionar é criado no seu ambiente de substituição.
    - Se você optar por Fornecer instâncias manualmente: habilite Grupos do Amazon EC2 Auto Scaling, Instâncias do Amazon EC2 ou ambos para especificar instâncias a serem adicionadas nesse grupo de implantação. Insira valores de tags do Amazon EC2 ou nomes de grupo do Amazon EC2 Auto Scaling para identificar as instâncias no ambiente original (ou seja, as instâncias que você deseja substituir ou que estão executando a revisão de aplicativo atual).

12. Em Balanceador de carga, selecione Habilitar balanceamento de carga e, nas listas, selecione os grupos de destino Classic Load Balancers, Application Load Balancer e Network Load Balancer com os quais você deseja registrar suas instâncias substitutas do Amazon EC2. Cada instância substituta será registrada em todos os Classic Load Balancers e grupos de destino selecionados. Você pode selecionar até 10 Classic Load Balancers e 10 grupos de destino, totalizando 20 itens.

    O tráfego será redirecionado das instâncias originais para as instâncias substitutas de acordo com as definições de Redirecionamento de tráfego e Configuração de implantação escolhidas.

    Para obter mais informações sobre balanceadores de carga para CodeDeploy implantações, consulte. Integrating CodeDeploy with Elastic Load Balancing
13. Em Configurações de implantação, reveja as opções padrão para o redirecionamento do tráfego ao ambiente de substituição, qual configuração de implantação usar para a implantação e como as instâncias no ambiente original serão tratadas após a implantação.

    Se quiser alterar as configurações, continue na próxima etapa. Caso contrário, vá para a etapa 15.
14. Para alterar as configurações de implantação da blue/green implantação, altere qualquer uma das configurações a seguir.

Configuração	Opções
Redirecionamento de tráfego	• Rotear novamente o tráfego imediatam ente: assim que as instâncias no ambiente de substituição forem provisionadas e receberem a instalação da revisão mais recente do aplicativo, elas serão registrad as nos balanceadores de carga e grupos de destino especificados automaticamente, fazendo com que o tráfego seja roteado novamente para elas. O registro das instâncias no ambiente original será então cancelado. • Decidirei sobre rotear novamente o tráfego: as instâncias no ambiente de substituição apenas serão registradas

Configuração	Opções
	nos balanceadores de carga e grupos de destino especificados se você rotear novamente o tráfego manualmente. Se o tempo de espera especificado passar sem que o tráfego seja redirecionado, o status da implantação será alterado para parado.
Configuração de implantação	Escolha em que taxa as instâncias no ambiente de substituição são registradas nos balanceadores de carga e grupos de destino, por exemplo, uma de cada vez ou todos ao mesmo tempo. **Note** Depois que o tráfego for direciona do com sucesso ao ambiente de substituição, o registro de todas as instâncias no ambiente original será cancelado de uma só vez, independentemente da configuração de implantação selecionada. Para obter mais informações, consulte Trabalhando com configurações de implantação em CodeDeploy.

Configuração	Opções
Instâncias originais	• Encerrar as instâncias originais no grupo de implantação: quando o tráfego tiver sido roteado novamente ao ambiente de substituição, as instâncias cujo registro foi cancelado dos balanceadores de carga e grupos de destino serão encerradas após o período de espera especificado. • Manter as instâncias originais executando no grupo de implantação: quando o tráfego tiver sido roteado novamente ao ambiente de substituição, as instâncias cujo registro foi cancelado dos balanceadores de carga e grupos de destino continuarão a ser executadas.

15. (Opcional) Em Avançado, configure as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon ou reversões automáticas.

    Para obter informações sobre como especificar opções avançadas em grupos de implantação, consulte Configurar opções avançadas para um grupo de implantação.

16. Selecione Criar grupo de implantação.

A próxima etapa é preparar uma revisão a ser implantada no aplicativo e grupo de implantação. Para instruções, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

Criar um aplicativo para uma implantação do serviço do Amazon ECS (console)

Você pode usar o CodeDeploy console para criar um aplicativo para a implantação de um serviço Amazon ECS.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e selecione Conceitos básicos.
3. Na página Criar aplicativo, escolha Usar CodeDeploy.
4. Em Nome da aplicação, digite o nome do aplicativo.
5. Em Plataforma de computação, selecione Amazon ECS.
6. Escolha Criar aplicativo.
7. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação. Para obter mais informações sobre o que você precisa para criar um grupo de implantação para uma implantação do Amazon ECS, consulte Antes de começar uma implantação do.
8. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

> **Note**
>
> Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação e a configuração de implantação), escolha essas configurações nessa página. Embora esse novo grupo e o grupo existente possam ter o mesmo nome, CodeDeploy trate-os como grupos de implantação separados, pois cada um está associado a um aplicativo separado.

9. Em Função de serviço, escolha uma função de serviço que conceda CodeDeploy acesso ao Amazon ECS. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.
10. Em Nome do balanceador de carga, escolha o nome do balanceador de carga que distribui o tráfego para seu serviço do Amazon ECS.
11. Em Porta do receptor de produção, escolha a porta e o protocolo para o receptor que fornece o tráfego de produção para seu serviço do Amazon ECS.
12. (Opcional) Em Porta do receptor de teste, escolha a porta e o protocolo de um receptor de teste que fornece tráfego para o conjunto de tarefas de substituição em seu serviço do Amazon ECS durante a implantação. Você pode especificar uma ou mais funções Lambda no AppSpec arquivo que são executadas durante o hook. `AfterAllowTestTraffic` As funções podem executar testes de validação. Se houver falha em um teste de validação, uma reversão da

implantação será acionada. Se o teste de validação for bem-sucedido, o próximo gancho no ciclo de vida da implantação, `BeforeAllowTraffic`, será acionado. Se a porta do receptor de teste não for especificada, nada acontecerá durante o hook `AfterAllowTestTraffic`. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.

13. Em Nome do grupo-alvo 1 e Nome do grupo-alvo 2, escolha os grupos-alvo usados para rotear o tráfego durante sua implantação. CodeDeploy vincula um grupo-alvo ao conjunto de tarefas original do seu serviço Amazon ECS e o outro ao conjunto de tarefas de substituição. Para obter mais informações, consulte Grupos de destino para seus Application Load Balancers.
14. Escolha Redirecionar tráfego imediatamente ou Especificar quando redirecionar o tráfego para determinar quando redirecionar o tráfego para seu serviço do Amazon ECS atualizado.

    Se você escolher Redirecionar tráfego imediatamente, a implantação redirecionará automaticamente o tráfego após o provisionamento do conjunto de tarefas de substituição.

    Se você escolher Especificar quando redirecionar o tráfego, escolha o número de dias, horas e minutos a serem aguardados após o provisionamento bem-sucedido do conjunto de tarefas de substituição. Durante esse tempo de espera, os testes de validação nas funções Lambda especificadas no AppSpec arquivo são executados. Se o tempo de espera expirar antes que o tráfego seja redirecionado, o status de implantação mudará para `Stopped`.
15. Em Encerramento da revisão original, escolha o número de dias, horas e minutos a serem aguardados após uma implantação bem-sucedida antes que a tarefa original definida em seu serviço do Amazon ECS seja encerrada.
16. (Opcional) Em Avançado, configure todas as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon ou reversões automáticas.

    Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantação.

Crie um aplicativo para um AWS Lambda implantação de funções (console)

Você pode usar o CodeDeploy console para criar um aplicativo para a implantação de uma AWS Lambda função.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e selecione Conceitos básicos.
3. Na página Criar aplicativo, escolha Usar CodeDeploy.
4. Insira o nome do aplicativo em Application name (Nome do aplicativo).
5. Em Compute platform (Plataforma de computação), selecione AWS Lambda.
6. Escolha Criar aplicação.
7. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação.
8. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

> **Note**
>
> Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação e a configuração de implantação), escolha essas configurações nessa página. Embora esse novo grupo de implantação e o grupo de implantação existente possam ter o mesmo nome, CodeDeploy trate-os como grupos de implantação separados, pois cada um está associado a um aplicativo separado.

9. Em Função de serviço, escolha uma função de serviço que conceda CodeDeploy acesso AWS Lambda a. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.
10. Se você deseja usar uma configuração de implantação predefinida, selecione uma opção em Deployment configuration (Configuração de implantação) e vá para a etapa 12. Para criar uma configuração personalizada, vá para a próxima etapa.

    Para obter mais informações sobre configurações de implantação, consulte Configurações de implantação em um AWS Lambda Plataforma de computação.
11. Para criar uma configuração personalizada, escolha Create deployment configuration (Criar uma configuração de implantação) e faça o seguinte:

    a. Em Deployment configuration name (Nome da configuração de implantação), digite um nome para a configuração.

b. Em Type (Tipo), escolha um tipo de configuração. Se você escolher Canário, o tráfego será deslocado em dois incrementos. Se você escolher Linear, o tráfego será deslocado em incrementos iguais com um número igual de minutos entre cada incremento.

c. Em Etapa, digite uma porcentagem do tráfego, entre 1 e 99, que será deslocada. Se o seu tipo de configuração é Canário, essa é a porcentagem do tráfego que será deslocada no primeiro incremento. O tráfego restante é deslocado após o intervalo selecionado no segundo incremento. Se o seu tipo de configuração é Linear, essa é a porcentagem do tráfego que será deslocada no início de cada intervalo.

d. Em Interval (Intervalo), digite um valor em minutos. Se o seu tipo de configuração é Canário, esse é o número de minutos entre o primeiro e o segundo deslocamento de tráfego. Se o seu tipo de configuração é Linear, esse é o número de minutos entre cada deslocamento incremental.

> **Note**
>
> A duração máxima de uma AWS Lambda implantação é de dois dias ou 2.880 minutos. Portanto, o valor máximo especificado por Intervalo para uma configuração canário é 2.800 minutos. O valor máximo de uma configuração linear depende do valor de Etapa. Por exemplo, se a porcentagem de etapa de um deslocamento de tráfego linear for 25%, haverá quatro deslocamentos de tráfego. O valor máximo do intervalo é 2.880 dividido por quatro, ou 720 minutos.

e. Escolha Criar configuração de implantação.

12. (Opcional) Em Avançado, configure todas as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon ou reversões automáticas.

    Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantação.

13. Selecione Criar grupo de implantação.

Criar um aplicativo (CLI)

Para usar o AWS CLI para criar um aplicativo, chame o comando create-application, especificando um nome que represente exclusivamente o aplicativo. (Em uma AWS conta, o nome do CodeDeploy

aplicativo só pode ser usado uma vez por região. Você pode reutilizar o nome de um aplicativo em diferentes regiões.)

Depois de usar o AWS CLI para criar um aplicativo, a próxima etapa é criar um grupo de implantação que especifique as instâncias nas quais implantar as revisões. Para instruções, consulte Crie um grupo de implantação com CodeDeploy.

Depois de criar o grupo de implantação, a próxima etapa é preparar uma revisão a ser implantada no aplicativo e grupo de implantação. Para instruções, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

Veja os detalhes do aplicativo com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre todos os aplicativos associados à sua AWS conta.

Tópicos

- Visualizar detalhes de aplicativos (console)
- Visualizar detalhes de aplicativos (CLI)

Visualizar detalhes de aplicativos (console)

Para usar o CodeDeploy console para visualizar os detalhes do aplicativo:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e selecione Introdução.
3. Para visualizar detalhes de aplicativos adicionais, escolha o nome do aplicativo na lista.

Visualizar detalhes de aplicativos (CLI)

Para usar o AWS CLI para visualizar os detalhes do aplicativo, chame o get-application batch-get-application comando, o comando ou o list-applications comando.

Para visualizar detalhes sobre um único aplicativo, chame o comando get-application, especificando o nome do aplicativo.

Para visualizar detalhes sobre vários aplicativos, chame o comando batch-get-applications, especificando vários nomes de aplicativos.

Para ver uma lista de nomes de aplicativos, chame o comando list-applications.

Criar uma regra de notificação

É possível usar regras de notificação para notificar os usuários quando houver alterações nos aplicativos de implantação, como êxitos e falhas de implantação. As regras de notificação especificam os eventos e o tópico do Amazon SNS que é usado para enviar notificações. Para obter mais informações, consulte O que são notificações?

Você pode usar o console ou o AWS CLI para criar regras de notificação para AWS CodeDeploy.

Como criar uma regra de notificação (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.
2. Escolha Application (Aplicativo) e selecione um aplicativo em que você deseja adicionar notificações.
3. Na página do aplicativo, escolha Notify (Notificar) e Create notification rule (Criar regra de notificação). Também é possível acessar a página Settings (Configurações) do aplicativo e escolher Create notification rule (Criar regra de notificação).
4. Em Notification name (Nome da notificação), insira um nome para a regra.
5. Em Tipo de detalhe, escolha Básico se quiser que somente as informações fornecidas à Amazon sejam EventBridge incluídas na notificação. Escolha Completo se quiser incluir informações fornecidas à Amazon EventBridge e informações que possam ser fornecidas pelo gerenciador de notificações CodeDeploy ou pelo gerenciador de notificações.

   Para obter mais informações, consulte Noções básicas sobre o conteúdo e a segurança de notificações.
6. Em Events that trigger notifications (Eventos que acionam notificações), selecione os eventos para os quais você deseja enviar notificações.

Categoria	Eventos
Implantação	Failed
	Bem-sucedido
	Iniciou

7. Em Targets (Destinos), escolha Create SNS topic (Criar tópico do SNS).

> **Note**
>
> Quando você cria o tópico, a política que CodeDeploy permite publicar eventos no tópico é aplicada a você. Usar um tópico criado especificamente para CodeDeploy notificações também ajuda a garantir que você adicione somente usuários à lista de assinaturas desse tópico para os quais deseja ver notificações sobre esse aplicativo de implantação.

Após o prefixo codestar-notifications-, insira um nome para o tópico e escolha Submit (Enviar).

> **Note**
>
> Se desejar usar um tópico do Amazon SNS existente em vez de criar um novo, em Targets (Destinos), escolha o ARN. Certifique-se de que o tópico tenha a política de acesso apropriada e que a lista de assinantes contenha somente os usuários que têm permissão para ver informações sobre o aplicativo de implantação. Para obter mais informações, consulte Configurar tópicos do Amazon SNS existentes para notificações e Noções básicas sobre conteúdos de notificações e segurança.

8. Para concluir a criação da regra, escolha Submit (Enviar).
9. Você precisa inscrever os usuários no tópico do Amazon SNS para a regra antes que eles possam receber notificações. Para obter mais informações, consulte Inscrever usuários em tópicos do Amazon SNS que são destinos. Você também pode configurar a integração entre as notificações e o Amazon Q Developer em aplicações de chat a fim de enviar notificações para salas de chat do Amazon Chime ou canais do Slack. Para acessar mais informações, consulte Configure integration between notifications and Amazon Q Developer in chat applications.

Para criar uma regra de notificação (AWS CLI)

1. Em um terminal ou prompt de comando, execute o comando create-notification rule para gerar o esqueleto JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

É possível nomear o arquivo como você quiser. Neste exemplo, o nome do arquivo é *rule.json*.

2. Abra o arquivo JSON em um editor de texto simples e edite-o para incluir o recurso, os tipos de evento e o destino do Amazon SNS que você deseja para a regra. O exemplo a seguir mostra uma regra de notificação com **MyNotificationRule** o nome de um aplicativo nomeado *MyDeploymentApplication* em uma AWS conta com o ID*123456789012*. As notificações são enviadas com o tipo de detalhe completo para um tópico do Amazon SNS chamado *codestar-notifications-MyNotificationTopic* quando as implantações são bem-sucedidas:

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codedeploy-application-deployment-succeeded"
    ],
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyDeploymentApplication",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

Salve o arquivo.

3. Usando o arquivo que você acabou de editar, no terminal ou na linha de comando, execute o comando create-notification-rule novamente para criar a regra de notificação:

```
aws codestar-notifications create-notification-rule --cli-input-json
 file://rule.json
```

4. Se for bem-sucedido, o comando retornará o ARN da regra de notificação, semelhante ao seguinte:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Renomear um aplicativo CodeDeploy

Você pode usar as APIs AWS CLI ou as CodeDeploy APIs para alterar o nome de um aplicativo.

Para ver uma lista de nomes de aplicativos, use o AWS CLI para chamar o comando list-applications.

Para obter informações sobre como usar o AWS CLI para alterar o nome de um aplicativo, consulte update-application.

Para obter informações sobre como usar as CodeDeploy APIs para alterar o nome de um aplicativo, consulte UpdateApplicationAPI_.

Excluir um aplicativo no CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou uma ação de CodeDeploy API para excluir aplicativos. Para obter informações sobre como usar a ação CodeDeploy da API, consulte DeleteApplication.

> **Warning**
>
> A exclusão de um aplicativo remove informações sobre ele do sistema do CodeDeploy, incluindo todas as informações relacionadas ao grupo de implantação e os detalhes da implantação. A exclusão de um aplicativo criado para uma EC2/On-Premises implantação não remove nenhuma revisão do aplicativo das instâncias nem exclui revisões dos buckets do Amazon S3. A exclusão de um aplicativo criado para uma EC2/On-Premises implantação

não encerra nenhuma instância do Amazon EC2 nem cancela o registro de nenhuma instância local. Esta ação não pode ser desfeita.

Tópicos

- Excluir um aplicativo (console)
- Excluir um aplicativo (AWS CLI)

Excluir um aplicativo (console)

Para usar o CodeDeploy console para excluir um aplicativo:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicações.
3. Na lista de aplicativos, escolha o aplicativo que você deseja excluir.

   Uma página contendo detalhes sobre o aplicativo é exibida.
4. Escolha Excluir aplicativo no canto superior direito.
5. Quando solicitado, digite **`delete`** para confirmar que você deseja excluí-lo e, em seguida, escolha Excluir.

Excluir um aplicativo (AWS CLI)

Para usar o AWS CLI para excluir um aplicativo, chame o comando delete-application, especificando o nome do aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.

Trabalhando com grupos de implantação em CodeDeploy

Você pode especificar um ou mais grupos de implantação para um CodeDeploy aplicativo. Cada implantação de aplicativo usa um de seus grupos de implantação. O grupo de implantação contém as definições e configurações usadas durante a implantação. A maioria das configurações do grupo de implantação depende da plataforma de computação usada pelo seu aplicativo. Algumas configurações, como reversões, triggers e alarmes, podem ser definidas para grupos de implantação para qualquer .

Grupos de implantação em implantações da plataforma de computação

Em uma implantação da Amazon ECS, um grupo de implantação especifica o serviço do Amazon ECS, balanceador de carga, receptor de teste opcional e dois grupos de destino. Ele também especifica quando redirecionar o tráfego para o conjunto de tarefas de substituição e quando encerrar o conjunto de tarefas original definida e o aplicativo do Amazon ECS após uma implantação bem-sucedida.

Grupos de implantação em AWS Lambda implantações de plataformas de computação

Em uma implantação do AWS Lambda, um grupo de implantação define um conjunto de CodeDeploy configurações para futuras implantações de uma função. AWS Lambda Por exemplo, o grupo de implantação especifica como rotear o tráfego para uma nova versão de uma função do Lambda. Ele também pode especificar alarmes e reversões. Uma única implantação em um grupo de implantação do AWS Lambda pode substituir uma ou mais configurações de grupo.

Grupos de implantação em implantações da EC2/On-Premises Compute Platform

Em uma EC2/On-Premises implantação, um grupo de implantação é um conjunto de instâncias individuais direcionadas para uma implantação. Um grupo de implantação contém instâncias do Amazon EC2; individualmente marcadas e/ou instâncias do Amazon EC2; em grupos do Amazon EC2 Auto Scaling.

Em uma implantação no local, as instâncias no grupo de implantação são atualizadas com a última revisão de aplicativo.

Em uma blue/green implantação, o tráfego é redirecionado de um conjunto de instâncias para outro cancelando o registro das instâncias originais de um ou mais balanceadores de carga e registrando um conjunto substituto de instâncias que normalmente tem a revisão mais recente do aplicativo já instalada.

É possível associar mais de um grupo de implantação a um aplicativo no CodeDeploy. Isso permite implantar uma revisão de aplicativo em diferentes conjuntos de instâncias em momentos diferentes. Por exemplo, é possível usar um grupo de implantação para implantar uma revisão de aplicativo em um conjunto de instâncias marcadas `Test` nas quais você garante a qualidade do código. Em seguida, você implanta a mesma revisão de aplicativo em um grupo de implantação com instâncias marcadas `Staging` para verificação adicional. Finalmente, quando estiver pronto para lançar o aplicativo mais recente aos clientes, você implantará em um grupo de implantação que inclua instâncias marcadas `Production`.

Você também pode usar vários grupos de tags para refinar ainda mais os critérios das instâncias incluídas em um grupo de implantação. Para mais informações, consulte Tagging Instances for Deployments.

Ao usar o CodeDeploy console para criar um aplicativo, você configura seu primeiro grupo de implantação ao mesmo tempo. Ao usar o AWS CLI para criar um aplicativo, você cria seu primeiro grupo de implantação em uma etapa separada.

Para ver uma lista dos grupos de implantação já associados à sua AWS conta, consulteVeja os detalhes do grupo de implantação com CodeDeploy.

Para obter informações sobre tags de instâncias do Amazon EC2, consulte Trabalhando com tags usando o console. Para obter informações sobre instâncias on-premises, consulte Working with On-Premises Instances. Para obter informações sobre o Amazon EC2 Auto Scaling, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

Tópicos

- the section called “Criar um grupo de implantação”
- the section called “Visualizar detalhes do grupo de implantação.”
- the section called “Altere as configurações do grupo de implantação.”
- the section called “Configurar opções avançadas para um grupo de implantação”

- the section called “Exclua um grupo de implantação”

Crie um grupo de implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, as CodeDeploy APIs ou um CloudFormation modelo para criar grupos de implantação. Para obter informações sobre como usar um CloudFormation modelo para criar um grupo de implantação, consulteCloudFormation modelos para CodeDeploy referência.

Ao usar o CodeDeploy console para criar um aplicativo, você configura seu primeiro grupo de implantação ao mesmo tempo. Ao usar o AWS CLI para criar um aplicativo, você cria seu primeiro grupo de implantação em uma etapa separada.

Como parte da criação de um grupo de implantação, você deve especificar uma função de serviço. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Tópicos

- Criar um grupo de implantação para uma implantação no local (console)
- Crie um grupo de implantação para uma EC2/On-Premises blue/green implantação (console)
- Criar um grupo de implantação para uma implantação do Amazon ECS (console)
- Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2
- Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS
- Criar um grupo de implantação (CLI)

Criar um grupo de implantação para uma implantação no local (console)

Para usar o CodeDeploy console para criar um grupo de implantação para uma implantação local:

> ⚠ Warning
>
> Não siga essas etapas se:
>
> - Você não preparou suas instâncias para serem usadas na primeira implantação do CodeDeploy de um aplicativo. Para configurar suas instâncias, siga as instruções em Trabalhando com instâncias para CodeDeploy e depois siga as etapas neste tópico.

- Você deseja criar um grupo de implantação que usa uma configuração de implantação personalizada, mas ainda não criou essa configuração de implantação. Siga as instruções em Create a Deployment Configuration e depois siga as etapas neste tópico.
- Você não tem uma função de serviço que CodeDeploy confie, no mínimo, na confiança e nas permissões descritas emEtapa 2: criar uma função de serviço para CodeDeploy. Para criar e configurar um perfil de serviço, siga as instruções em Etapa 2: criar uma função de serviço para CodeDeploy e depois siga as etapas neste tópico.
- Você deseja selecionar um Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing para a implantação no local, mas ainda não o criou.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na página Aplicativos, escolha o nome do aplicativo para o qual você deseja criar um grupo de implantação.
4. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação.
5. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

   > **Note**
   >
   > Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação, tags e/ou nomes de grupos do Amazon EC2 Auto Scaling, bem como a configuração de implantação), especifique essas configurações nessa página. Embora esse novo grupo de implantação e o grupo de implantação existente tenham o mesmo nome, eles CodeDeploy são tratados como grupos de implantação separados, pois cada um deles está associado a aplicativos separados.

6. Em Service role (Função de serviço), escolha uma função de serviço que conceda acesso do CodeDeploy à sua instância de destino.
7. Em Tipo de implantação, escolha In-place.
8. Em Configuração do ambiente, faça o seguinte:

   a. Se quiser implantar a aplicação em um grupo do Amazon EC2 Auto Scaling, selecione Grupos do Amazon EC2 Auto Scaling e escolha o nome de um grupo do Amazon EC2 Auto Scaling no qual implantar a revisão da aplicação. Quando novas instâncias do Amazon EC2 são lançadas como parte de um grupo do Amazon EC2 Auto Scaling CodeDeploy , você pode implantar suas revisões nas novas instâncias automaticamente. É possível adicionar até 10 grupos do Amazon EC2 Auto Scaling a um grupo de implantação. Para obter mais informações, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

   b. Se você selecionou grupos do Amazon EC2 Auto Scaling, opcionalmente selecione Adicionar um gancho de terminação aos grupos do Auto Scaling para instalar um gancho de terminação em seu grupo do Auto Scaling CodeDeploy ao criar ou atualizar o grupo de implantação. Quando esse gancho for instalado, CodeDeploy executará implantações de terminação. Para obter mais informações, consulte Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling.

   c. Se você quiser marcar suas instâncias, selecione instâncias ou On-premises instâncias do Amazon EC2. Nos campos Chave e Valor, insira os valores do par de chave/valor utilizado para marcar as instâncias. É possível marcar até 10 pares de chave/valor em um único grupo de tags.

      i. Você pode usar caracteres curinga no campo Valor para identificar todas as instâncias marcadas em determinados padrões, como instâncias do Amazon EC2, centros de custo e nomes de grupos semelhantes e assim por diante. Por exemplo, se você escolher Nome no campo Chave e inserir **`GRP-*a`** no campo Valor, CodeDeploy identificará todas as instâncias que se encaixam nesse padrão**`GRP-1a`**, como**`GRP-2a`**, e. **`GRP-XYZ-a`**

      ii. O campo Valor diferencia maiúsculas de minúsculas.

      iii. Para remover um par de chave/valor da lista, escolha o ícone Remover.

      Quando CodeDeploy encontra instâncias que correspondem a cada par de valores-chave especificado ou ao nome do grupo Amazon EC2 Auto Scaling, ele exibe o número de

instâncias correspondentes. Para ver mais informações sobre as instâncias, clique no número.

Se você quiser refinar os critérios para as instâncias implantadas, escolha Adicionar grupo de tags para criar um grupo de tags. É possível criar até três grupos de tags com até 10 pares chave/valor de cada. Quando você usa grupos de várias tags em um grupo de implantação, apenas as instâncias que são identificadas por todos os grupos de tags são incluídas no grupo de implantação. Isso significa que uma instância deve corresponder pelo menos a uma das tags em cada um dos grupos a serem incluídos no grupo de implantação.

Para obter mais informações sobre como usar grupos de tags para refinar o grupo de implantação, consulte Tagging Instances for Deployments.

9. Em Configuração do agente com Systems Manager, especifique como você gostaria de instalar e atualizar o CodeDeploy agente nas instâncias do seu grupo de implantação. Para obter mais informações sobre o CodeDeploy agente, consulte Trabalhando com o CodeDeploy agente. Para obter mais informações sobre o Systems Manager, consulte O que é o Systems Manager?

   a. Nunca: ignore a configuração da CodeDeploy instalação com o Systems Manager. As instâncias devem ter o agente instalado para ser usado em implantações, portanto, escolha essa opção somente se você instalar o agente do CodeDeploy de outra maneira.

   b. Somente uma vez: o Systems Manager instalará o CodeDeploy agente uma vez em cada instância do seu grupo de implantação.

   c. Agora e agende as atualizações: o Systems Manager criará uma associação com o State Manager que instalará o CodeDeploy agente de acordo com a programação que você configurar. Para obter mais informações sobre o gerenciador de estados e as associações, consulte Sobre o State Manager.

10. Em Deployment configuration (Configuração da implantação), escolha uma configuração de implantação para controlar a taxa em que as instâncias são implantadas, por exemplo, uma de cada vez ou todas ao mesmo tempo. Para obter mais informações sobre configurações de implantação, consulte Trabalhando com configurações de implantação em CodeDeploy.

11. (Opcional) No Load Balancer, selecione Ativar balanceamento de carga e, nas listas, selecione os grupos-alvo Classic Load Balancers, Application Load Balancer e Network Load Balancer para gerenciar o tráfego para as instâncias durante a implantação. CodeDeploy Você pode selecionar até 10 Classic Load Balancers e 10 grupos de destino, totalizando 20 itens. Verifique se as instâncias do Amazon EC2 nas quais você deseja implantar estão registradas nos

balanceadores de carga (Classic Load Balancers) ou nos grupos de destino (Application Load Balancers e Network Load Balancers) selecionados.

Durante uma implantação, o registro das instâncias originais dos balanceadores de carga e dos grupos de destino selecionados é cancelado para evitar que o tráfego seja roteado para essas instâncias durante a implantação. Quando a implantação for concluída, cada instância será registrada novamente com todos os Classic Load Balancers e grupos de destino selecionados.

Para obter mais informações sobre balanceadores de carga para CodeDeploy implantações, consulte. Integrating CodeDeploy with Elastic Load Balancing

> ⚠ Warning
>
> Se você estiver configurando grupos de Auto Scaling e um load balancer do Elastic Load Balancing nesse grupo de implantação e quiser anexar balanceadores de carga aos grupos do Auto Scaling, recomendamos preencher este anexo CodeDeploy antes de criar a implantação a partir desse grupo de implantação. A tentativa de concluir o anexo após criar a implantação pode fazer com que o registro de todas as instâncias dos balanceadores de carga seja cancelado inesperadamente.

12. (Opcional) Expanda Advanced e configure todas as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon, opções de Auto Scaling ou reversões automáticas.

    Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantação.

13. Selecione Criar grupo de implantação.

Crie um grupo de implantação para uma EC2/On-Premises blue/green implantação (console)

Para usar o CodeDeploy console para criar um grupo de implantação para uma blue/green implantação:

> ⚠ Warning
>
> Não siga essas etapas se:

- Você não tem instâncias com o CodeDeploy agente instalado que queira substituir durante o processo blue/green de implantação. Para configurar suas instâncias, siga as instruções em Trabalhando com instâncias para CodeDeploy e depois siga as etapas neste tópico.
- Você deseja criar um aplicativo que usa uma configuração de implantação personalizada, mas ainda não criou essa configuração de implantação. Siga as instruções em Create a Deployment Configuration e depois siga as etapas neste tópico.
- Você não tem uma função de serviço que CodeDeploy confie, no mínimo, na confiança e nas permissões descritas emEtapa 2: criar uma função de serviço para CodeDeploy. Para criar e configurar um perfil de serviço, siga as instruções em Etapa 2: criar uma função de serviço para CodeDeploy e depois siga as etapas neste tópico.
- Você não criou um Classic Load Balancer nem um Application Load Balancer no Elastic Load Balancing para o registro das instâncias no seu ambiente de substituição. Para obter mais informações, consulte Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na página Aplicativos, escolha o nome do aplicativo para o qual você deseja criar um grupo de implantação.
4. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação.
5. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

   > **Note**
   >
   > Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação, as tags, os nomes de grupos do Amazon EC2 Auto Scaling e a configuração de implantação), escolha essas configurações nessa página. Embora esse novo grupo de implantação e o grupo de implantação

existente tenham o mesmo nome, o CodeDeploy os tratará como grupos de implantação separados, porque eles estão associados a aplicativos separados.

6. Em Service role (Função de serviço), escolha uma função de serviço que conceda acesso do CodeDeploy à sua instância de destino.
7. Em Tipo de implantação, escolha Blue/green.
8. Em Configuração do ambiente, faça o seguinte:

   - Selecione o método a ser utilizado para fornecer instâncias para o ambiente de substituição: Você tem as seguintes opções:
     - Copiar automaticamente o grupo Amazon EC2 Auto Scaling: cria um CodeDeploy grupo Amazon EC2 Auto Scaling copiando um que você especificar.
     - Fornecer instâncias manualmente: você apenas especificará as instâncias para o seu ambiente de substituição depois de criar uma implantação. Você deve criar as instâncias antes de iniciar a implantação. Em vez disso, aqui, você especifica as instâncias que deseja substituir.
   - Se você selecionou Copiar automaticamente o grupo Amazon EC2 Auto Scaling, opcionalmente selecione Adicionar um gancho de terminação aos grupos do Auto Scaling para instalar um gancho de terminação em seu grupo do Auto Scaling CodeDeploy ao criar ou atualizar o grupo de implantação. Quando esse gancho for instalado, CodeDeploy executará implantações de terminação. Para obter mais informações, consulte Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling.
9. Em Configuração do agente com Systems Manager, especifique como você gostaria de instalar e atualizar o CodeDeploy agente nas instâncias do seu grupo de implantação. Para obter mais informações sobre o CodeDeploy agente, consulte Trabalhando com o CodeDeploy agente. Para obter mais informações sobre o Systems Manager, consulte O que é o Systems Manager?

   a. Nunca: ignore a configuração da CodeDeploy instalação com o Systems Manager. As instâncias devem ter o agente instalado para ser usado em implantações, portanto, escolha essa opção somente se você instalar o agente do CodeDeploy de outra maneira.

   b. Somente uma vez: o Systems Manager instalará o CodeDeploy agente uma vez em cada instância do seu grupo de implantação.

   c. Agora e agende as atualizações: o Systems Manager criará uma associação com o State Manager que instalará o CodeDeploy agente de acordo com a programação que você configurar. Para obter mais informações sobre o gerenciador de estados e as associações, consulte Sobre o State Manager.

10. Dependendo da sua escolha na etapa 8, siga um destes procedimentos:

    - Se tiver escolhido Cópia automática do grupo do Amazon EC2 Auto Scaling: em Grupo do Amazon EC2 Auto Scaling, escolha ou insira o nome do grupo do Amazon EC2 Auto Scaling que você deseja usar como modelo para o grupo do Amazon EC2 Auto Scaling criado para as instâncias no seu ambiente de substituição. O número de instâncias atualmente íntegras no grupo do Amazon EC2 Auto Scaling que você selecionar é criado no seu ambiente de substituição.
    - Se você escolher Provisionar instâncias manualmente: selecione grupos do Amazon EC2 Auto Scaling, instâncias do Amazon EC2 Auto Scaling ou ambos para especificar instâncias a serem adicionadas a esse grupo de implantação. Insira valores de tags do Amazon EC2 Auto Scaling ou nomes de grupo do Amazon EC2 Auto Scaling para identificar as instâncias no ambiente original (ou seja, as instâncias que você deseja substituir ou que estão executando a revisão de aplicativo atual).

11. Em Balanceador de carga, selecione Habilitar balanceamento de carga e, nas listas, selecione os grupos de destino Classic Load Balancers, Application Load Balancer e Network Load Balancer com os quais você deseja registrar suas instâncias substitutas do Amazon EC2. Cada instância substituta será registrada em todos os Classic Load Balancers e grupos de destino selecionados. Você pode selecionar até 10 Classic Load Balancers e 10 grupos de destino, totalizando 20 itens.

    O tráfego será redirecionado das instâncias originais para as instâncias substitutas de acordo com as definições de Redirecionamento de tráfego e Configuração de implantação escolhidas.

    Para obter mais informações sobre balanceadores de carga para CodeDeploy implantações, consulte. Integrating CodeDeploy with Elastic Load Balancing

    > **Warning**
    >
    > Se você estiver configurando grupos de Auto Scaling e balanceadores de carga do Elastic Load Balancing nesse grupo de implantação e quiser anexar os balanceadores de carga aos grupos do Auto Scaling, recomendamos preencher este anexo antes de criar a implantação a partir desse grupo de implantação. CodeDeploy A tentativa de concluir o anexo após criar a implantação pode fazer com que o registro de todas as instâncias dos balanceadores de carga seja cancelado inesperadamente.

12. Em Configurações de implantação, reveja as opções padrão para o redirecionamento do tráfego ao ambiente de substituição, qual configuração de implantação usar para a implantação e como as instâncias no ambiente original serão tratadas após a implantação.

    Se quiser alterar as configurações, continue na próxima etapa. Caso contrário, vá para a etapa 14.

13. Para alterar as configurações de implantação da blue/green implantação, escolha qualquer uma das configurações a seguir.

Configuração	Opções
Redirecionamento de tráfego	• Rotear novamente o tráfego imediatam ente: assim que as instâncias no ambiente de substituição forem provisionadas e receberem a instalação da revisão mais recente do aplicativo, elas serão registrad as nos balanceadores de carga e grupos de destino especificados automaticamente, fazendo com que o tráfego seja roteado novamente para elas. O registro das instâncias no ambiente original será então cancelado. • Decidirei sobre rotear novamente o tráfego: as instâncias no ambiente de substituição apenas serão registradas nos balanceadores de carga e grupos de destino especificados se você rotear novamente o tráfego manualmente. Se o tempo de espera especificado passar sem que o tráfego seja redirecionado, o status da implantação será alterado para parado.
Configuração de implantação	Escolha em que taxa as instâncias no ambiente de substituição são registradas nos balanceadores de carga e grupos de destino, por exemplo, uma de cada vez ou todos ao mesmo tempo.

Configuração	Opções
	Note Depois que o tráfego for direciona do com sucesso ao ambiente de substituição, o registro de todas as instâncias no ambiente original será cancelado de uma só vez, independentemente da configuração de implantação selecionada. Para obter mais informações, consulte Trabalhando com configurações de implantação em CodeDeploy.

Configuração	Opções
Instâncias originais	• Encerrar as instâncias originais no grupo de implantação: quando o tráfego tiver sido roteado novamente ao ambiente de substituição, as instâncias cujo registro foi cancelado dos balanceadores de carga e grupos de destino serão encerradas após o período de espera especificado. Observe o seguinte: • Se seu grupo de Auto Scaling tiver a Proteção de Exclusão ativada (`prevent-force-deletion` ou `prevent-all-deletion`), a CodeDeploy blue/green implantação desativará essa proteção no grupo de Auto Scaling original (azul) antes de excluí-la durante a limpeza da implantação. O grupo substituto (verde) do Auto Scaling manterá as mesmas configurações de Proteção contra Exclusão do grupo original. • Manter as instâncias originais executando no grupo de implantação: quando o tráfego tiver sido roteado novamente ao ambiente de substituição, as instâncias cujo registro foi cancelado dos balanceadores de carga e grupos de destino continuarão a ser executadas.

14. (Opcional) Em Avançado, configure as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon, opções de Auto Scaling ou reversões automáticas.

    Para obter informações sobre como especificar opções avançadas em grupos de implantação, consulte Configurar opções avançadas para um grupo de implantação.

15. Selecione Criar grupo de implantação.

Criar um grupo de implantação para uma implantação do Amazon ECS (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Em Applications table (Tabela de aplicativos), escolha o nome do aplicativo que está associado ao grupo de implantação que você deseja editar.
4. Na página de seu aplicativo, em Grupos de implantação, escolha o nome do grupo de implantação que você deseja editar.
5. Na página do aplicativo, na guia Grupos de implantação, selecione Criar grupo de implantação. Para obter mais informações sobre o que você precisa para criar um grupo de implantação para uma implantação do Amazon ECS, consulte Antes de começar uma implantação do.
6. Em Nome do grupo de implantação digite um nome que descreva o grupo de implantação.

   > **Note**
   >
   > Se quiser usar as mesmas configurações usadas em outro grupo de implantação (incluindo o nome do grupo de implantação e a configuração de implantação), escolha essas configurações nessa página. Embora esse novo grupo e o grupo existente possam ter o mesmo nome, CodeDeploy trate-os como grupos de implantação separados, pois cada um está associado a um aplicativo separado.

7. Em Função de serviço, escolha uma função de serviço que conceda CodeDeploy acesso ao Amazon ECS. Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.
8. Em Nome do balanceador de carga, escolha o nome do balanceador de carga que distribui o tráfego para seu serviço do Amazon ECS.
9. Em Porta do receptor de produção, escolha a porta e o protocolo para o receptor que fornece o tráfego de produção para seu serviço do Amazon ECS.

10. (Opcional) Em Porta do receptor de teste, escolha a porta e o protocolo de um receptor de teste que fornece tráfego para o conjunto de tarefas de substituição em seu serviço do Amazon ECS durante a implantação. Você pode especificar uma ou mais funções Lambda no AppSpec arquivo que são executadas durante o hook. `AfterAllowTestTraffic` As funções podem executar testes de validação. Se houver falha em um teste de validação, uma reversão da implantação será acionada. Se o teste de validação for bem-sucedido, o próximo gancho no ciclo de vida da implantação, `BeforeAllowTraffic`, será acionado. Se a porta do receptor de teste não for especificada, nada acontecerá durante o hook `AfterAllowTestTraffic`. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.
11. Em Nome do grupo-alvo 1 e Nome do grupo-alvo 2, escolha os grupos-alvo usados para rotear o tráfego durante sua implantação. CodeDeploy vincula um grupo-alvo ao conjunto de tarefas original do seu serviço Amazon ECS e o outro ao conjunto de tarefas de substituição. Para obter mais informações, consulte Grupos de destino para seus Application Load Balancers.
12. Escolha Redirecionar tráfego imediatamente ou Especificar quando redirecionar o tráfego para determinar quando redirecionar o tráfego para seu serviço do Amazon ECS atualizado.

    Se você escolher Redirecionar tráfego imediatamente, a implantação redirecionará automaticamente o tráfego após o provisionamento do conjunto de tarefas de substituição.

    Se você escolher Especificar quando redirecionar o tráfego, escolha o número de dias, horas e minutos a serem aguardados após o provisionamento bem-sucedido do conjunto de tarefas de substituição. Durante esse tempo de espera, os testes de validação nas funções Lambda especificadas no AppSpec arquivo são executados. Se o tempo de espera expirar antes que o tráfego seja redirecionado, o status de implantação mudará para `Stopped`.
13. Em Encerramento da revisão original, escolha o número de dias, horas e minutos a serem aguardados após uma implantação bem-sucedida antes que a tarefa original definida em seu serviço do Amazon ECS seja encerrada.
14. (Opcional) Em Avançado, configure todas as opções que você deseja incluir na implantação, como acionadores de notificação do Amazon SNS, alarmes da CloudWatch Amazon ou reversões automáticas.

    Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantação.

Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2

Antes de executar qualquer blue/green implantação ou uma implantação local para a qual você deseja especificar um balanceador de carga opcional no grupo de implantação, você deve ter criado pelo menos um Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing. Para blue/green implantações, você usa esse balanceador de carga para registrar as instâncias que compõem seu ambiente substituto. As instâncias no seu ambiente original podem, opcionalmente, ser registradas nesse mesmo balanceador de carga. Para implantações no local, o balanceador de carga é usado para cancelar o registro de instâncias que estão sendo trabalhadas e registrá-las novamente CodeDeploy quando o trabalho for concluído.

CodeDeploy oferece suporte blue/green e implantação local em instâncias do Amazon EC2 por meio de vários balanceadores de carga. Por exemplo, supondo que você tenha 200 instâncias do Amazon EC2, em que 100 delas estão registradas com 2 Classic Load Balancers e outras 100 delas estão registradas com 4 grupos-alvo em 2 Application Load Balancers. Nesse cenário, CodeDeploy permitirá que você faça blue/green implantações no local em todas as 200 instâncias, mesmo que elas estejam espalhadas por 2 Classic Load Balancers, 2 Application Load Balancers e 4 grupos-alvo.

CodeDeploy suporta até 10 balanceadores de carga clássicos e 10 grupos-alvo, totalizando 20 itens.

Para configurar um ou mais Classic Load Balancers, siga as instruções no Tutorial: Criar um Classic Load Balancer no Guia do usuário para Classic Load Balancers. Observe o seguinte:

- Na Etapa 2: definir o balanceador de carga, em Criar LB interno, escolha o mesmo VPC que você selecionou quando criou suas instâncias.
- Na Etapa 5: Registrar instâncias do EC2 com seu Load Balancer, selecione as instâncias atualmente em seu grupo de implantação (implantações locais) ou que você designou para estarem em seu ambiente original (implantações). blue/green
- Na Etapa 7: criar e verificar seu balanceador de carga, anote o endereço DNS do seu balanceador de carga.

  Por exemplo, se você tiver chamado o load balancer de `my-load-balancer`, o endereço DNS aparecerá em um formato como `my-load-balancer-1234567890.us-east-2.elb.amazonaws.com`.

Para configurar um ou mais Application Load Balancers, siga as instruções em um dos seguintes tópicos:

- Criar um Application Load Balancer
- Tutorial: Crie um Application Load Balancer usando o AWS CLI

Para configurar um ou mais Network Load Balancers, siga as instruções em um dos seguintes tópicos:

- Criar um Network Load Balancer
- Tutorial: Crie um Network Load Balancer usando o AWS CLI

Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS

Antes de executar uma implantação usando a plataforma de computação do Amazon ECS, você deve criar um Application Load Balancer ou um Network Load Balancer, dois grupos de destino e um ou dois receptores. Este tópico mostra como criar um Application Load Balancer. Para obter mais informações, consulte Antes de começar uma implantação do.

Um dos grupos de destino direciona o tráfego para o conjunto de tarefas original do aplicativo do Amazon ECS. O outro grupo de destino direciona o tráfego para o conjunto de tarefas de substituição. Durante a implantação, CodeDeploy cria um conjunto de tarefas de substituição e redireciona o tráfego do conjunto de tarefas original para o novo. CodeDeploy determina qual grupo-alvo é usado para cada conjunto de tarefas.

Um listener é usado pelo load balancer para direcionar o tráfego para os grupos de destino. Um listener de produção é necessário. Você pode especificar um listener de teste opcional que direciona o tráfego para o conjunto de tarefas de substituição enquanto você executa testes de validação.

O load balancer deve usar uma VPC com duas sub-redes públicas em diferentes zonas de disponibilidade. As etapas a seguir mostram como confirmar sua VPC padrão, criar um Application Load Balancer do Amazon EC2 e criar dois grupos de destino para o balanceador de carga. Para obter mais informações, consulte Grupos de destino para seus Network Load Balancers.

Verificar a VPC padrão, as sub-redes públicas e o grupo de segurança

Este tópico mostra como criar um Amazon EC2 Application Load Balancer, dois grupos-alvo e duas portas que podem ser usadas durante uma implantação do Amazon ECS. Uma das portas é opcional e necessária somente se você direcionar o tráfego para uma porta de teste de validação durante a implantação.

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon VPC em. https://console.aws.amazon.com/vpc/
2. Verifique a VPC padrão a ser usada. No painel de navegação, escolha Your VPCs (Suas VPCs). Observe qual VPC exibe Sim na coluna de VPC padrão. Essa é a sua VPC padrão. Ela contém sub-redes padrão que você usa.
3. Selecione Subnets (Sub-redes). Anote os IDs das duas sub-redes que mostram Sim na coluna Sub-rede padrão. Use esses IDs ao criar o load balancer.
4. Depois de escolher cada sub-rede, escolha a guia Description (Descrição). Verifique se as sub-redes que você deseja utilizar se encontram em diferentes zonas de disponibilidade.
5. Selecione as sub-redes e escolha a guia Route Table (Tabela de rotas). Para verificar se cada sub-rede que você deseja usar é uma sub-rede pública, confirme se uma linha com um link para um gateway da Internet está incluída na tabela de rotas.
6. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
7. No painel de navegação, escolha Security groups (Grupos de segurança).
8. Verifique se o grupo de segurança que você deseja usar está disponível e anote o ID do grupo (por exemplo, sg-abcd1234). Você o usará ao criar o load balancer.

Crie um Application Load Balancer do Amazon EC2, dois grupos-alvo e receptores (console)

Para usar o console do Amazon EC2 para criar um Application Load Balancer do Amazon EC2:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em. https://console.aws.amazon.com/ec2/
2. No painel de navegação, selecione Load Balancers.
3. Selecione Criar load balancer.
4. Selecione Application Load Balancer e clique em Create (Criar).

5. Em Name (Nome), informe o nome do load balancer.
6. Em Scheme (Esquema), selecione Internet-facing (Voltado para a Internet).
7. Em IP address type (Tipo de endereço IP), selecione ipv4.
8. (Opcional) Configure uma segunda porta do listener para o load balancer. Você pode executar testes de validação da implantação usando o tráfego de teste fornecido para essa porta.

    a. Em Load Balancer Protocol (Protocolo do load balancer), escolha Add listener (Adicionar listener).
    b. Em Load Balancer Protocol (Protocolo do load balancer) para o segundo listener, selecione HTTP.
    c. Em Load Balancer Port (Porta do load balancer), insira **`8080`**.
9. Em Availability Zones (Zonas de disponibilidade), em VPC, escolha a VPC padrão e selecione as duas sub-redes padrão que você deseja usar.
10. Selecione Next: Configure Security Settings (Próximo: Definir configurações de segurança).
11. Selecione Next: Configure Security Groups (Próximo: Configurar grupos de segurança).
12. Escolha Select an existing security group (Selecionar um grupo de segurança existente), escolha o grupo de segurança padrão e anote seu ID.
13. Selecione Next: Configure Routing (Próximo: Configurar roteamento).
14. Em Target group (Grupo de destino), selecione New target group (Novo grupo de destino) e configure o primeiro grupo de destino:

    a. Em Nome, informe o nome do grupo de destino (por exemplo, **`target-group-1`**).
    b. Em Tipo de destino, selecione IP.
    c. Em Protocolo: Selecione HTTP. Em Porta, insira **`80`**.
    d. Selecione Next: Register Targets (Próximo: Registrar destinos).
15. Selecione Next: Review (Próximo: Análise) e Create (Criar).

Criar um grupo de destino para seu load balancer.

1. Após a provisão do balanceador de carga, abra o console do Amazon EC2. No painel de navegação, selecione Grupos de destino.
2. Selecione Criar grupo de destino.
3. Em Nome, informe o nome do grupo de destino (por exemplo, **`target-group-2`**).
4. Em Tipo de destino, selecione IP.

5. Em Protocolo: Selecione HTTP. Em Porta, insira **80**.
6. Em VPC, escolha a VPC padrão.
7. Escolha Criar.

> **Note**
>
> Para que sua implantação do Amazon ECS seja executada, é necessário ter dois grupos de destino criados para seu balanceador de carga. Você usará o ARN de um de seus grupos de destino ao criar o serviço do Amazon ECS. Para obter mais informações, consulte Etapa 4: Criar um serviço do Amazon ECS no Guia do usuário do Amazon ECS.

Crie um Application Load Balancer do Amazon EC2, dois grupos de destino e receptores (CLI)

Para criar um Application Load Balancer usando a AWS CLI:

1. Use o comando create-load-balancer para criar um Application Load Balancer. Especifique duas sub-redes que não estejam na mesma zona de disponibilidade e um grupo de segurança.

```
aws elbv2 create-load-balancer --name bluegreen-alb \
--subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
region us-east-1
```

   A saída inclui o nome de recurso da Amazon (ARN) do load balancer, no seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642
```

2. Use o comando create-target-group para criar seu primeiro grupo de destino. O CodeDeploy roteia o tráfego desse grupo de destino para o conjunto de tarefas original ou de substituição no serviço.

```
aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

   A saída inclui o ARN do primeiro grupo de destino, no seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4
```

3. Use o comando create-target-group para criar o segundo grupo de destino. O CodeDeploy roteia o tráfego do grupo de destino para o conjunto de tarefas que não é atendido pelo primeiro grupo de destino. Por exemplo, se o primeiro grupo de destino rotear o tráfego para o conjunto de tarefas original, o tráfego desse grupo de destino roteará o tráfego para o conjunto de tarefas de substituição.

```
aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

A saída inclui o ARN do segundo grupo de destino, no seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Use o comando create-listener para criar um listener com uma regra padrão que encaminha o tráfego de produção para a porta 80.

```
aws elbv2 create-listener --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 80 \
--default-actions
 Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

A saída inclui o ARN do listener, no seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. (Opcional) Use o comando create-listener para criar um segundo listener com uma regra padrão que encaminha o tráfego de teste para a porta 8080. Você pode executar testes de validação da implantação usando o tráfego de teste que é atendido por essa porta.

```
aws elbv2 create-listener --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 8080 \
--default-actions
 Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup
bluegreentarget2/209a844cd01825a4 --region us-east-1
```

A saída inclui o ARN do listener, no seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

Criar um grupo de implantação (CLI)

Para usar o AWS CLI para criar um grupo de implantação, chame o comando create-deployment-group, especificando:

- O nome do aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- Um nome para o grupo de implantação. Um grupo de implantação com esse nome é criado para o aplicativo especificado. Um grupo de implantação só pode ser associado a um aplicativo.
- Informações sobre as tags, grupos de tags ou os nomes de grupo do Amazon EC2 Auto Scaling que identificam as instâncias a serem incluídas no grupo de implantação.
- O identificador Amazon Resource Name (ARN) da função de serviço que permite CodeDeploy agir em nome da sua AWS conta ao interagir com outros serviços. AWS Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (CLI). Para obter mais informações sobre perfis de serviço, consulte Termos e conceitos de funções no Guia do usuário do IAM.
- Informações sobre o tipo de implantação, seja no local ou blue/green associado ao grupo de implantação.
- (Opcional) O nome de uma configuração de implantação existente. Para ver uma lista de configurações de implantação, consulte View Deployment Configuration Details. Se não for especificado, o CodeDeploy usará uma configuração de implantação padrão.
- (Opcional) Comandos para criar um gatilho que envia notificações sobre eventos de implantação e instância para aqueles que estão inscritos em um tópico do Amazon Simple Notification

Service. Para obter mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications.

- (Opcional) Comandos para adicionar CloudWatch alarmes existentes ao grupo de implantação que são ativados se uma métrica especificada em um alarme cair abaixo ou exceder um limite definido.
- (Opcional) Comandos para que uma implantação seja revertida para a última revisão válida quando uma implantação falha ou um CloudWatch alarme é ativado.
- (Opcional) Comandos para uma implantação gerar hooks de eventos de ciclo de vida durante um evento de redução da escala horizontal do Auto Scaling. Para obter mais informações, consulte Como o Amazon EC2 Auto Scaling funciona com CodeDeploy.
- Para implantações no local:
  - (Opcional) Os nomes dos Classic Load Balancers, Application Load Balancers ou Network Load Balancers no Elastic Load Balancing que gerenciam o tráfego para as instâncias durante os processos de implantação.
- Para blue/green implantações:
  - Configuração do processo blue/green de implantação:
    - Como novas instâncias no ambiente de substituição são provisionadas.
    - Se o tráfego deve ser redirecionado ao ambiente de substituição imediatamente ou se é necessário aguardar um período especificado para o tráfego ser redirecionado manualmente.
    - Se as instâncias no ambiente original devem ser encerradas.
  - Os nomes dos Classic Load Balancers, Application Load Balancers ou Network Load Balancers no Elastic Load Balancing a serem usados para instâncias registradas no ambiente de substituição.

> **Warning**
>
> Se você estiver configurando um grupo de Auto Scaling e um balanceador de carga do Elastic Load Balancing em seu grupo de implantação e quiser anexar o balanceador de carga ao grupo de Auto Scaling, recomendamos preencher este anexo CodeDeploy antes de criar a implantação a partir desse grupo de implantação. A tentativa de concluir o anexo após criar a implantação pode fazer com que o registro de todas as instâncias do balanceador de carga seja cancelado inesperadamente.

Veja os detalhes do grupo de implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre todos os grupos de implantação associados a um aplicativo.

Tópicos

- Visualizar detalhes do grupo de implantação (console)
- Visualizar detalhes do grupo de implantação (CLI)

Visualizar detalhes do grupo de implantação (console)

Para usar o CodeDeploy console para ver os detalhes do grupo de implantação:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na página Aplicativos, escolha o nome do aplicativo associado ao grupo de implantação.

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

4. Para visualizar detalhes sobre um grupo de implantação individual, na guia Deployment groups (Grupos de implantação), escolha o nome do grupo de implantação.

Visualizar detalhes do grupo de implantação (CLI)

Para usar o AWS CLI para visualizar os detalhes do grupo de implantação, chame o `get-deployment-group` comando ou o `list-deployment-groups` comando.

Para visualizar detalhes sobre um único grupo de implantação, chame o comando get-deployment-group, especificando:

- O nome do aplicativo associado ao grupo de implantação. Para obter o nome do aplicativo, chame o comando list-applications.
- O nome do grupo de implantação. Para obter o nome do grupo de implantação, chame comando list-deployment-groups.

Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups, especificando o nome do aplicativo associado aos grupos de implantação. Para obter o nome do aplicativo, chame o comando list-applications.

Altere as configurações do grupo de implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para alterar as configurações de um grupo de implantação.

> **Warning**
>
> Não use essas etapas se quiser que o grupo de implantação use um grupo de implantação personalizado ainda não criado. Em vez disso, siga as instruções em Create a Deployment Configuration e, em seguida, volte para esse tópico. Não use essas etapas se quiser que o grupo de implantação use uma função de serviço diferente ainda não criada. A função de serviço deve confiar CodeDeploy , no mínimo, nas permissões descritas emEtapa 2: criar uma função de serviço para CodeDeploy. Para criar e configurar uma função de serviço com as permissões corretas, siga as instruções em Etapa 2: criar uma função de serviço para CodeDeploy e depois volte para este tópico.

Tópicos

- Alterar configurações do grupo de implantação (console)
- Alterar configurações do grupo de implantação (CLI)

Alterar configurações do grupo de implantação (console)

Para usar o CodeDeploy console para alterar as configurações do grupo de implantação:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na lista de aplicativos, escolha o nome do aplicativo que está associado com o grupo de implantação que você deseja mudar.

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

4. Escolha a guia Deployment groups (Grupos de implantação) e escolha o nome do grupo de implantação que você deseja alterar.
5. Na página do grupo de implantação, escolha Editar.
6. Edite as opções de grupo de implantação, conforme necessário.

   Para obter informações sobre componentes do grupo de implantação, consulte Crie um grupo de implantação com CodeDeploy.

7. Escolha Salvar alterações.

Alterar configurações do grupo de implantação (CLI)

Para usar o AWS CLI para alterar as configurações do grupo de implantação, chame o comando update-deployment-group, especificando:

- Implantações EC2/On-Premises For e AWS Lambda:
  - O nome do aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.

- O nome do grupo de implantação atual. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.
- (Opcional) Um nome de grupo de implantação diferente.
- (Opcional) Um nome de recurso da Amazon (ARN) diferente que corresponde a uma função de serviço que permite CodeDeploy agir em nome da sua AWS conta ao interagir com outros serviços. AWS Para obter o ARN do perfil de serviço, consulte Obter o ARN do perfil de serviço (CLI). Para obter mais informações sobre perfis de serviço, consulte Termos e conceitos de funções no Guia do usuário do IAM.
- (Opcional) O nome da configuração de implantação. Para ver uma lista de configurações de implantação, consulte View Deployment Configuration Details. (Se não for especificado, CodeDeploy usa uma configuração de implantação padrão.)
- (Opcional) Comandos para adicionar um ou mais CloudWatch alarmes existentes ao grupo de implantação que são ativados se uma métrica especificada em um alarme ficar abaixo ou exceder um limite definido.
- (Opcional) Comandos para que uma implantação seja revertida para a última revisão válida quando uma implantação falha ou um CloudWatch alarme é ativado.
- (Opcional) Comandos para uma implantação gerar hooks de eventos de ciclo de vida durante um evento de redução da escala horizontal do Auto Scaling. Para obter mais informações, consulte Como o Amazon EC2 Auto Scaling funciona com CodeDeploy.
- (Opcional) Comandos para criar ou atualizar um gatilho que publica em um tópico no , para que os assinantes desse tópico recebam notificações sobre eventos de implantação e instância nesse grupo de implantação. Para mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications.

- Somente para EC2/On-Premises implantações:
  - (Opcional) Tags de substituição ou grupos de tags que identificam de forma exclusiva as instâncias a serem incluídas no grupo de implantação.
  - (Opcional) Os nomes dos grupos do Amazon EC2 Auto Scaling de substituição a serem adicionados ao grupo de implantação.
- Somente para implantações do Amazon ECS:
  - Para excluir uma serviço do Amazon ECS
  - Informações do balanceador de carga, incluindo o Application Load Balancer ou o Network Load Balancer, os grupos de destino necessários para uma implantação do Amazon ECS e as informações do receptor de teste opcional e de produção.

Configurar opções avançadas para um grupo de implantação

Ao criar ou atualizar um grupo de implantação, você pode configurar uma série de opções para fornecer mais controle e supervisão sobre as implementações desse grupo de implantação.

Use as informações nessa página para ajudá-lo a configurar opções avançadas ao trabalhar com grupos de implantação nos seguintes tópicos:

- Crie um aplicativo com CodeDeploy
- Crie um grupo de implantação com CodeDeploy
- Altere as configurações do grupo de implantação com CodeDeploy

Acionadores de notificação do Amazon SNS: você pode adicionar gatilhos a um grupo de CodeDeploy implantação para receber notificações sobre eventos relacionados às implantações nesse grupo de implantação. Essas notificações são enviadas aos destinatários inscritos em um tópico do Amazon SNS que você incluiu como parte da ação do gatilho.

Você já deve ter configurado o tópico do Amazon SNS para o qual esse gatilho apontará e CodeDeploy deve ter permissão para publicar no tópico desse grupo de implantação. Se você ainda não tiver concluído essas etapas de configuração, poderá adicionar gatilhos ao grupo de implantação mais tarde.

Se quiser criar um gatilho agora para receber notificações sobre eventos de implantação no grupo de implantação desse aplicativo, escolha Criar gatilho.

Se a sua implantação se destina a uma instância do Amazon EC2, é possível criar notificações referentes a instâncias e receber notificações sobre elas.

Para obter mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications.

CloudWatch Alarmes da Amazon: você pode criar um CloudWatch alarme que monitora uma única métrica durante um período de tempo especificado e executa uma ou mais ações com base no valor da métrica em relação a um determinado limite em vários períodos. Para uma implantação do Amazon EC2, você pode criar um alarme para uma instância ou grupo do Amazon EC2 Auto Scaling que você está usando em suas operações. CodeDeploy Para uma implantação do AWS Lambda e do Amazon ECS, você pode criar um alarme para erros em uma função do Lambda.

Você pode configurar uma implantação para parar quando um CloudWatch alarme da Amazon detectar que uma métrica caiu abaixo ou excedeu um limite definido.

Você já deve ter criado o alarme CloudWatch antes de poder adicioná-lo a um grupo de implantação.

1. Para adicionar monitoramento de alarmes ao grupo de implantação, em Alarms (Alarmes), escolha Add alarm (Adicionar alarme).
2. Insira o nome de um CloudWatch alarme que você já configurou para monitorar essa implantação.

   Você deve inserir o CloudWatch alarme exatamente como ele foi criado em CloudWatch. Para ver uma lista de alarmes, abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/ e escolha ALARME.

Opções adicionais:

- Se quiser que as implantações prossigam sem levarem em conta os alarmes que você adicionou, escolha Ignorar configuração de alarme.

  Essa opção é útil quando você deseja desativar temporariamente o monitoramento de alarmes para um grupo de implantação sem precisar adicionar os mesmos alarmes novamente mais tarde.
- (Opcional) Se você quiser que CodeDeploy as implantações continuem caso de impossibilidade de recuperar o status do alarme da Amazon CloudWatch, escolha Continuar implantações mesmo que o status do alarme não esteja disponível.

> **Note**
>
> Essa opção corresponde ao ignorePollAlarmFailure AlarmConfigurationobjeto na CodeDeploy API.

Para obter mais informações, consulte Monitorando implantações com CloudWatch alarmes em CodeDeploy.

Reversões automáticas: você pode configurar um grupo de implantação ou uma implantação para reversão automática quando uma implantação falhar ou quando um limite de monitoramento especificado for atendido. Nesse caso, a última versão conhecida em bom estado de uma revisão de aplicativo é implantada. Você pode definir configurações opcionais para um grupo de implantação ao usar o console para criar um aplicativo, criar um grupo de implantação ou atualizar um grupo de

implantação. Ao criar uma nova implantação, você também pode optar por substituir a configuração de reversão automática que foi especificada para o grupo de implantação.

- É possível habilitar implantações para que elas sejam revertidas para a revisão em bom estado mais recente conhecida quando algo der errado, escolhendo uma das seguintes opções, ou ambas:

  - Reverta quando uma implantação falhar. CodeDeploy reimplantará a última revisão válida conhecida como uma nova implantação.
  - Reverter quando limites de alarme forem atendidos. Se você adicionou um alarme a esse aplicativo na etapa anterior, CodeDeploy reimplantará a última revisão válida quando um ou mais dos alarmes especificados forem ativados.

  > **Note**
  >
  > Para ignorar temporariamente uma configuração de reversão, escolha Desabilitar reversões. Essa opção é útil quando você deseja desabilitar temporariamente as reversões automáticas sem precisar definir a mesma configuração novamente mais tarde.

  Para obter mais informações, consulte Reimplante e reverta uma implantação com CodeDeploy.

Atualizações automáticas para instâncias desatualizadas: sob certas circunstâncias, CodeDeploy pode implantar uma revisão desatualizada do seu aplicativo em suas instâncias do Amazon EC2. Por exemplo, se suas instâncias do EC2 forem lançadas em um grupo de Auto Scaling (ASG) enquanto CodeDeploy a implantação estiver em andamento, essas instâncias receberão a revisão mais antiga do seu aplicativo em vez da versão mais recente. Para atualizar essas instâncias, inicia CodeDeploy automaticamente uma implantação subsequente (imediatamente após a primeira) para atualizar todas as instâncias desatualizadas. Se você quiser alterar esse comportamento padrão para que as instâncias EC2 desatualizadas sejam deixadas na revisão mais antiga, você pode fazer isso por meio da CodeDeploy API ou da AWS Command Line Interface (CLI).

Para configurar atualizações automáticas de instâncias desatualizadas por meio da API, inclua o parâmetro de solicitação `outdatedInstancesStrategy` na ação `UpdateDeploymentGroup` ou `CreateDeploymentGroup`. Para obter detalhes, consulte a Referência da API do AWS CodeDeploy .

Para configurar as atualizações automáticas por meio do AWS CLI, use um dos seguintes comandos:

```
aws deploy update-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

Ou...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

... where *arguments* é substituído pelos argumentos necessários para sua implantação e *UPDATE|IGNORE* é substituído por UPDATE para habilitar atualizações automáticas ou IGNORE para desativá-las.

Exemplo:

```
aws deploy update-deployment-group --application-name "MyApp" --current-deployment-group-name "MyDG" --region us-east-1 --outdated-instances-strategy IGNORE
```

Para obter detalhes sobre esses AWS CLI comandos, consulte a Referência de AWS CLI Comandos.

Exclua um grupo de implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para excluir grupos de implantação associados à sua AWS conta.

> **Warning**
>
> Se você excluir um grupo de implantação, todos os detalhes associados a esse grupo de implantação também serão excluídos CodeDeploy. As instâncias utilizadas no grupo de implantação permanecerão inalteradas. Esta ação não pode ser desfeita.

Tópicos

- Excluir um grupo de implantação (console)
- Excluir um grupo de implantação (CLI)

Excluir um grupo de implantação (console)

Para usar o CodeDeploy console para excluir um grupo de implantação:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na lista de aplicativos, escolha o nome do aplicativo associado ao grupo de implantação.
4. Na página Application details (Detalhes do aplicativo), na guia Deployment groups (Grupos de implantação), escolha o nome do grupo de implantação que você deseja excluir.
5. Na página Deployment details (Detalhes da implantação), escolha Delete (Excluir).
6. Quando solicitado, digite o nome do grupo de implantação para confirmar que você deseja excluí-lo e, em seguida, escolha Excluir.

Excluir um grupo de implantação (CLI)

Para usar o AWS CLI para excluir um grupo de implantação, chame o comando delete-deployment-group, especificando:

- O nome do aplicativo associado ao grupo de implantação. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- O nome do grupo de implantação associado ao aplicativo. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.

Trabalhando com revisões de aplicativos para CodeDeploy

Em CodeDeploy, uma revisão contém uma versão dos arquivos de origem que CodeDeploy serão implantados em suas instâncias ou scripts CodeDeploy serão executados em suas instâncias.

Você planeja a revisão, adiciona um AppSpec arquivo à revisão e, em seguida, envia a revisão para o Amazon S3 ou. GitHub Depois de enviar a revisão, você poderá implantá-la.

Tópicos

- Planeje uma revisão para CodeDeploy
- Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy
- Escolha um tipo de CodeDeploy repositório
- Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações)
- Visualize os detalhes da revisão do aplicativo com CodeDeploy
- Registre uma revisão do aplicativo no Amazon S3 com CodeDeploy

Planeje uma revisão para CodeDeploy

Um bom planejamento facilita muito mais a implantação de revisões.

Para implantações em uma plataforma computacional AWS Lambda ou Amazon ECS, uma revisão é igual ao arquivo. AppSpec As informações a seguir não se aplicam. Para obter mais informações, consulte Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

Para implantações em uma plataforma de EC2/On-Premises computação, comece criando um diretório raiz vazio (pasta) na máquina de desenvolvimento. Este é o local onde você armazenará os arquivos de origem (como arquivos de texto e binários, executáveis, pacotes e assim por diante) a serem implantados nas instâncias ou os scripts a serem executados nessas instâncias.

Por exemplo, na pasta raiz `/tmp/` no Linux, macOS ou Unix ou na pasta raiz `c:\temp` no Windows:

```
/tmp/ or c:\temp (root folder)
  |--content (subfolder)
  |    |--myTextFile.txt
  |    |--mySourceFile.rb
  |    |--myExecutableFile.exe
```

```
|    |--myInstallerFile.msi
|    |--myPackage.rpm
|    |--myImageFile.png
|--scripts (subfolder)
|    |--myShellScript.sh
|    |--myBatchScript.bat
|    |--myPowerShellScript.ps1
|--appspec.yml
```

A pasta raiz também deve incluir um arquivo de especificação do aplicativo (AppSpec arquivo), conforme mostrado aqui. Para obter mais informações, consulte Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy

Este tópico mostra como adicionar um AppSpec arquivo à sua implantação. Também inclui modelos para criar um AppSpec arquivo para um AWS Lambda e EC2/On-Premises implantação.

Tópicos

- Adicionar um AppSpec arquivo para uma implantação do Amazon ECS
- Adicionar um AppSpec arquivo para um AWS implantação do Lambda
- Adicionar um AppSpec arquivo para uma EC2/On-Premises implantação

Adicionar um AppSpec arquivo para uma implantação do Amazon ECS

Para uma implantação em uma plataforma de computação do Amazon ECS:

- O AppSpec arquivo especifica a definição de tarefa do Amazon ECS usada para a implantação, um nome de contêiner e mapeamento de portas usados para rotear o tráfego e funções Lambda opcionais executadas após eventos do ciclo de vida da implantação.
- Uma revisão é o mesmo que um AppSpec arquivo.
- Um AppSpec arquivo pode ser gravado usando JSON ou YAML.
- Um AppSpec arquivo pode ser salvo como um arquivo de texto ou inserido diretamente em um console quando você cria uma implantação. Para obter mais informações, consulte Criar uma implantação da plataforma de computação do Amazon ECS (console).

Para criar um AppSpec arquivo

1. Copie o modelo JSON ou YAML em um editor de texto ou no AppSpec editor no console.
2. Modifique o template conforme necessário.
3. Use um validador JSON ou YAML para validar seu arquivo. AppSpec Se você usar o AppSpec editor, o arquivo será validado quando você escolher Criar implantação.
4. Se você usar um editor de texto, salve o arquivo. Se você usar o AWS CLI para criar sua implantação, faça referência ao AppSpec arquivo se ele estiver no seu disco rígido ou em um bucket do Amazon S3. Se você usa o console, deve enviar seu AppSpec arquivo para o Amazon S3.

Modelo de AppSpec arquivo YAML para uma implantação do Amazon ECS com instruções

A seguir está um modelo YAML de um AppSpec arquivo para uma implantação do Amazon ECS com todas as opções disponíveis. Para obter informações sobre os eventos de ciclo de vida que serão usados na seção `hooks`, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.

```
# This is an appspec.yml template file for use with an Amazon ECS deployment in
 CodeDeploy.
# The lines in this template that start with the hashtag are
#   comments that can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# In the Resources section, you must specify the following: the Amazon ECS service,
 task definition name,
# and the name and port of the load balancer to route traffic,
# target version, and (optional) the current version of your AWS Lambda function.
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "" # Specify the ARN of your task definition
 (arn:aws:ecs:region:account-id:task-definition/task-definition-family-name:task-
definition-revision-number)
        LoadBalancerInfo:
```

```
          ContainerName: "" # Specify the name of your Amazon ECS application's
 container
          ContainerPort: "" # Specify the port for your container where traffic
 reroutes
# Optional properties
        PlatformVersion: "" # Specify the version of your Amazon ECS Service
        NetworkConfiguration:
          AwsvpcConfiguration:
            Subnets: ["",""] # Specify one or more comma-separated subnets in your
 Amazon ECS service
            SecurityGroups: ["",""] # Specify one or more comma-separated security
 groups in your Amazon ECS service
            AssignPublicIp: "" # Specify "ENABLED" or "DISABLED"
# (Optional) In the Hooks section, specify a validation Lambda function to run during
# a lifecycle event.
Hooks:
# Hooks for Amazon ECS deployments are:
    - BeforeInstall: "" # Specify a Lambda function name or ARN
    - AfterInstall: "" # Specify a Lambda function name or ARN
    - AfterAllowTestTraffic: "" # Specify a Lambda function name or ARN
    - BeforeAllowTraffic: "" # Specify a Lambda function name or ARN
    - AfterAllowTraffic: "" # Specify a Lambda function name or ARN
```

AppSpec Arquivo JSON para um modelo de implantação do Amazon ECS

A seguir está um modelo JSON para um AppSpec arquivo para uma implantação do Amazon ECS com todas as opções disponíveis. Para obter instruções do modelo, consulte os comentários na versão YAML da seção anterior. Para obter informações sobre os eventos de ciclo de vida que serão usados na seção `hooks`, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.

```
{
 "version": 0.0,
 "Resources": [
  {
   "TargetService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
       "TaskDefinition": "",
       "LoadBalancerInfo": {
        "ContainerName": "",
        "ContainerPort":
       },
       "PlatformVersion": "",
```

```
       "NetworkConfiguration": {
        "AwsvpcConfiguration": {
         "Subnets": [
          "",
          ""
         ],
         "SecurityGroups": [
          "",
          ""
         ],
         "AssignPublicIp": ""
        }
       }
     }
   }
  }
 ],
 "Hooks": [
  {
   "BeforeInstall": ""
  },
  {
   "AfterInstall": ""
  },
  {
   "AfterAllowTestTraffic": ""
  },
  {
   "BeforeAllowTraffic": ""
  },
  {
   "AfterAllowTraffic": ""
  }
 ]
}
```

Adicionar um AppSpec arquivo para um AWS implantação do Lambda

Para uma implantação em uma plataforma de computação AWS Lambda:

- O AppSpec arquivo contém instruções sobre as funções do Lambda a serem implantadas e usadas para validação da implantação.
- Uma revisão é o mesmo que um AppSpec arquivo.

- Um AppSpec arquivo pode ser gravado usando JSON ou YAML.
- Um AppSpec arquivo pode ser salvo como um arquivo de texto ou inserido diretamente em um AppSpec editor de console ao criar uma implantação. Para obter mais informações, consulte Crie um AWS Implantação da Lambda Compute Platform (console).

Para criar um AppSpec arquivo:

1. Copie o modelo JSON ou YAML em um editor de texto ou no AppSpec editor no console.
2. Modifique o template conforme necessário.
3. Use um validador JSON ou YAML para validar seu arquivo. AppSpec Se você usar o AppSpec editor, o arquivo será validado quando você escolher Criar implantação.
4. Se você usar um editor de texto, salve o arquivo. Se você usar o AWS CLI para criar sua implantação, faça referência ao AppSpec arquivo se ele estiver no seu disco rígido ou em um bucket do Amazon S3. Se você usa o console, deve enviar seu AppSpec arquivo para o Amazon S3.

Modelo de AppSpec arquivo YAML para um AWS Lambda implantação com instruções

Para obter informações sobre os eventos de ciclo de vida utilizados na seção "Hooks", consulte AppSpec seção 'ganchos' para um AWS implantação do Lambda.

```
# This is an appspec.yml template file for use with an AWS Lambda deployment in
 CodeDeploy.
# The lines in this template starting with the hashtag symbol are
#   instructional comments and can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# In the Resources section specify the name, alias,
# target version, and (optional) the current version of your AWS Lambda function.
Resources:
  - MyFunction: # Replace "MyFunction" with the name of your Lambda function
      Type: AWS::Lambda::Function
      Properties:
        Name: "" # Specify the name of your Lambda function
```

```
        Alias: "" # Specify the alias for your Lambda function
        CurrentVersion: "" # Specify the current version of your Lambda function
        TargetVersion: "" # Specify the version of your Lambda function to deploy
# (Optional) In the Hooks section, specify a validation Lambda function to run during
# a lifecycle event. Replace "LifeCycleEvent" with BeforeAllowTraffic
# or AfterAllowTraffic.
Hooks:
    - LifeCycleEvent: "" # Specify a Lambda validation function between double-quotes.
```

AppSpec Arquivo JSON para um AWS Lambda modelo de implantação

No modelo a seguir, substitua MyFunction "" pelo nome da sua AWS Lambda função. Na seção Hooks opcional, substitua os eventos do ciclo de vida por ou. BeforeAllowTraffic AfterAllowTraffic

Para obter informações sobre os eventos de ciclo de vida utilizados na seção "Hooks", consulte AppSpec seção 'ganchos' para um AWS implantação do Lambda.

```
{
  "version": 0.0,
  "Resources": [{
   "MyFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "",
     "Alias": "",
     "CurrentVersion": "",
     "TargetVersion": ""
    }
   }
  }],
  "Hooks": [{
    "LifeCycleEvent": ""
   }
  ]
 }
```

Adicionar um AppSpec arquivo para uma EC2/On-Premises implantação

Sem um AppSpec arquivo, CodeDeploy não é possível mapear os arquivos de origem na revisão do aplicativo para seus destinos ou executar scripts para sua implantação em uma plataforma EC2/On-Premises computacional.

Cada revisão deve conter somente um AppSpec arquivo.

Para adicionar um AppSpec arquivo a uma revisão:

1. Copie o template para um editor de texto.
2. Modifique o template conforme necessário.
3. Use um validador YAML para verificar a validade do seu AppSpec arquivo.
4. Salve o arquivo como `appspec.yml` no diretório raiz da revisão.
5. Execute um dos comandos a seguir para verificar se você colocou seu AppSpec arquivo no diretório raiz:

   - Para Linux, macOS ou Unix:

     ```
     find /path/to/root/directory -name appspec.yml
     ```

     Não haverá saída se o AppSpec arquivo não for encontrado lá.

   - Para Windows:

     ```
     dir path\to\root\directory\appspec.yml
     ```

     Um erro Arquivo não encontrado será exibido se o AppSpec arquivo não estiver armazenado lá.

6. Envie a revisão para o Amazon S3 ou. GitHub

   Para instruções, consulte Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações).

AppSpec modelo de arquivo para uma EC2/On-Premises implantação com instruções

> **Note**
>
> As implantações em instâncias do Windows Server não oferecem suporte ao elemento `runas`. Se você estiver implantando em instâncias do Windows Server, não o inclua em seu AppSpec arquivo.

```
# This is an appspec.yml template file for use with an EC2/On-Premises deployment in
 CodeDeploy.
# The lines in this template starting with the hashtag symbol are
#   instructional comments and can be safely left in the file or
#   ignored.
# For help completing this file, see the "AppSpec File Reference" in the
#   "CodeDeploy User Guide" at
#   https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
# Specify "os: linux" if this revision targets Amazon Linux,
#   Red Hat Enterprise Linux (RHEL), or Ubuntu Server
#   instances.
# Specify "os: windows" if this revision targets Windows Server instances.
# (You cannot specify both "os: linux" and "os: windows".)
os: linux
# os: windows
# During the Install deployment lifecycle event (which occurs between the
#   BeforeInstall and AfterInstall events), copy the specified files
#   in "source" starting from the root of the revision's file bundle
#   to "destination" on the Amazon EC2 instance.
# Specify multiple "source" and "destination" pairs if you want to copy
#   from multiple sources or to multiple destinations.
# If you are not copying any files to the Amazon EC2 instance, then remove the
#   "files" section altogether. A blank or incomplete "files" section
#   may cause associated deployments to fail.
files:
  - source:
    destination:
  - source:
    destination:
# For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
#   you can specify a "permissions"
#   section here that describes special permissions to apply to the files
#   in the "files" section as they are being copied over to
#   the Amazon EC2 instance.
#   For more information, see the documentation.
# If you are deploying to Windows Server instances,
#   then remove the
#   "permissions" section altogether. A blank or incomplete "permissions"
#   section may cause associated deployments to fail.
permissions:
  - object:
    pattern:
```

```
    except:
    owner:
    group:
    mode:
    acls:
      -
    context:
      user:
      type:
      range:
    type:
      -
# If you are not running any commands on the Amazon EC2 instance, then remove
#   the "hooks" section altogether. A blank or incomplete "hooks" section
#   may cause associated deployments to fail.
hooks:
# For each deployment lifecycle event, specify multiple "location" entries
#   if you want to run multiple scripts during that event.
# You can specify "timeout" as the number of seconds to wait until failing the
 deployment
#   if the specified scripts do not run within the specified time limit for the
#   specified event. For example, 900 seconds is 15 minutes. If not specified,
#   the default is 1800 seconds (30 minutes).
#   Note that the maximum amount of time that all scripts must finish executing
#   for each individual deployment lifecycle event is 3600 seconds (1 hour).
#   Otherwise, the deployment will stop and CodeDeploy will consider the deployment
#   to have failed to the Amazon EC2 instance. Make sure that the total number of
 seconds
#   that are specified in "timeout" for all scripts in each individual deployment
#   lifecycle event does not exceed a combined 3600 seconds (1 hour).
# For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
#   you can specify "runas" in an event to
#   run as the specified user. For more information, see the documentation.
#   If you are deploying to Windows Server instances,
#   remove "runas" altogether.
# If you do not want to run any commands during a particular deployment
#   lifecycle event, remove that event declaration altogether. Blank or
#   incomplete event declarations may cause associated deployments to fail.
# During the ApplicationStop deployment lifecycle event, run the commands
#   in the script specified in "location" starting from the root of the
#   revision's file bundle.
  ApplicationStop:
    - location:
      timeout:
```

```
      runas:
    - location:
      timeout:
      runas:
# During the BeforeInstall deployment lifecycle event, run the commands
#   in the script specified in "location".
  BeforeInstall:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the AfterInstall deployment lifecycle event, run the commands
#   in the script specified in "location".
  AfterInstall:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the ApplicationStart deployment lifecycle event, run the commands
#   in the script specified in "location".
  ApplicationStart:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
# During the ValidateService deployment lifecycle event, run the commands
#   in the script specified in "location".
  ValidateService:
    - location:
      timeout:
      runas:
    - location:
      timeout:
      runas:
```

Escolha um tipo de CodeDeploy repositório

O local de armazenamento dos arquivos exigidos pelo CodeDeploy é chamado de repositório. O uso de um repositório depende de qual plataforma de computação sua implantação usa.

- EC2/On-Premises: para implantar o código do aplicativo em uma ou mais instâncias, seu código deve ser empacotado em um arquivo e colocado em um repositório onde CodeDeploy você possa acessá-lo durante o processo de implantação. Você agrupa seu conteúdo implantável e um AppSpec arquivo em um arquivo de arquivamento e, em seguida, faz o upload para um dos tipos de repositório suportados pelo. CodeDeploy
- AWS Lambda e Amazon ECS: as implantações exigem um AppSpec arquivo, que pode ser acessado durante a implantação de uma das seguintes formas:
  - De um bucket do Amazon S3.
  - Do texto digitado diretamente no AppSpec editor no console. Para obter mais informações, consulte Crie um AWS Implantação da Lambda Compute Platform (console) e Criar uma implantação da plataforma de computação do Amazon ECS (console).
  - Se você usar o AWS CLI, poderá referenciar um AppSpec arquivo que esteja no seu disco rígido ou em uma unidade de rede. Para obter mais informações, consulte Crie um AWS Implantação da Lambda Compute Platform (CLI) e Crie uma implantação da Amazon ECS Compute Platform (CLI).

CodeDeploy atualmente oferece suporte aos seguintes tipos de repositório:

Tipo de repositório	Detalhes de repositório	Plataformas de computação com suporte
Amazon S3	O Amazon Simple Storage Service (Amazon S3) é a solução da AWS para o armazenamento de objetos seguro e escalável. O Amazon S3 armazena dados como objetos dentro de buckets. Um objeto é composto de um arquivo e, opcionalmente,	As implantações que usam as plataformas de computaçã o a seguir podem armazenar a revisão em um bucket do Amazon S3. • EC2/On-Premises • AWS Lambda • Amazon ECS

	quaisquer metadados que descrevam esse arquivo. Para armazenar um objeto no Amazon S3, carregue o arquivo em um bucket. Ao carregar um arquivo, você pode definir permissões e metadados no objeto. Saiba mais: • Criar um bucket no Amazon S3 • Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações) • Implemente automaticamente a partir do Amazon S3 usando CodeDeploy	
GitHub	Você pode armazenar as revisões do seu aplicativo em GitHubrepositórios. Você pode acionar uma implantação a partir de um GitHub repositório sempre que o código-fonte desse repositório for alterado. Saiba mais: • Integrando com CodeDeploy GitHub • Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub	Somente EC2/On-Premises implantações podem armazenar a revisão em um GitHub repositório.

Bitbucket	Você pode implantar código em grupos de implantação de instâncias do EC2 usando o CodeDeploy pipe no Bitbucket Pipelines. O Bitbucket Pipelines oferece recursos de integração contínua e implantação contínua (CI/CD), incluindo implantações do Bitbucket. O CodeDeploy pipe primeiro envia o artefato para um bucket do S3 que você especificou e, em seguida, implanta o artefato de código do bucket. Saiba mais: • Veja o CodeDeploy tubo do Bitbucket	Somente EC2/On-Premises implantações podem armazenar a revisão em um BitBucket repositório.

> **Note**
>
> Uma AWS Lambda implantação funciona somente com um repositório Amazon S3.

Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações)

Depois de planejar sua revisão conforme descrito em Planeje uma revisão para CodeDeploy e adicionar um AppSpec arquivo à revisão conforme descrito emAdicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy, você estará pronto para agrupar os arquivos de componentes e enviar a revisão para o Amazon S3. Para implantações em instâncias do Amazon EC2, depois de enviar a revisão, você pode CodeDeploy usá-la para implantar a revisão do Amazon S3 nas instâncias.

> **Note**
>
> CodeDeploy também pode ser usado para implantar revisões que foram enviadas para GitHub. Para obter mais informações, consulte sua GitHub documentação.

Supomos que você já tenha seguido as instruções em Começando com CodeDeploy para configurar a AWS CLI. Isso é especialmente importante para chamar o comando push descrito mais adiante.

Certifique-se de ter um bucket do Amazon S3. Siga as instruções em Criar um bucket.

Se sua implantação for para instâncias do Amazon EC2, o bucket de destino do Amazon S3 deverá ser criado ou existir na mesma região das instâncias de destino. Por exemplo, se você quiser implantar uma revisão em algumas instâncias na região Leste dos EUA (Norte da Virgínia) e outras na região Oeste dos EUA (Oregon), você deve ter um bucket na região Leste dos EUA (Norte da Virgínia) com uma cópia da revisão e outro bucket na região Oeste dos EUA (Oregon) com outra cópia da mesma revisão. Nesse cenário, você precisaria criar duas implantações separadas, uma na região Leste dos EUA (Norte da Virgínia) e outra na região Oeste dos EUA (Oregon), mesmo que a revisão seja a mesma nas regiões e nos buckets.

Você deve ter permissões para fazer upload no bucket do Amazon S3. Você pode especificar essas permissões por meio de uma política de bucket do Amazon S3. Por exemplo, na seguinte política de bucket do Amazon S3, o uso do caractere curinga (*) permite que `111122223333` a AWS conta faça upload de arquivos para qualquer diretório no bucket do Amazon S3 chamado: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
```

```
    ]
}
```

Para ver o ID AWS da sua conta, consulte Encontrando o ID AWS da sua conta.

Para saber mais sobre como gerar e anexar uma política de bucket do Amazon S3, consulte Exemplos de políticas de bucket.

O usuário que está chamando o comando push deve ter, no mínimo, permissões para fazer upload da revisão em cada bucket de destino do Amazon S3. Por exemplo, a seguinte política permite que o usuário faça upload de revisões em qualquer lugar no bucket do Amazon S3 denominado `amzn-s3-demo-bucket`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Para saber mais sobre como criar e anexar uma política do IAM, consulte Trabalhando com políticas.

Faça uma revisão usando o AWS CLI

> **Note**
>
> O `push` comando agrupa artefatos do aplicativo e um AppSpec arquivo em uma revisão. O formato de arquivo dessa revisão é um arquivo ZIP compactado. O comando não pode ser usado com uma implantação do AWS Lambda ou do Amazon ECS porque cada um espera uma revisão que seja um JSON-formatted arquivo ou. YAML-formatted AppSpec

Chame o comando push para agrupar e enviar a revisão por push para uma implantação. Os parâmetros são:

- --application-name: (string) necessário. O nome do aplicativo do CodeDeploy que será associado à revisão de aplicativo.
- --s3-location: (string) necessário. Informações sobre a localização da revisão de aplicativo que será carregada no Amazon S3. Você deve especificar um bucket e uma chave do Amazon S3. A chave é o nome da revisão. CodeDeploy compacta o conteúdo antes de ser carregado. Use o formato `s3://`*`amzn-s3-demo-bucket`*`/`*`your-key.zip`*.
- --ignore-hidden-files ou --no-ignore-hidden-files: (booliano) opcional. Use o sinalizador `--no-ignore-hidden-files` (o padrão) para agrupar e carregar arquivos ocultos para o Amazon S3. Use o sinalizador `--ignore-hidden-files` para não agrupar e carregar arquivos ocultos para o Amazon S3.
- --source (string) opcional. A localização do conteúdo a ser implantado e o AppSpec arquivo na máquina de desenvolvimento a ser compactado e carregado no Amazon S3. A localização é especificada como um caminho relativo ao diretório atual. Se o caminho relativo não for especificado, ou se um ponto final for usado para o caminho ("."), o diretório atual será usado.
- --description (string) opcional. Um comentário que resume a revisão de aplicativo. Se não for especificada, a string padrão “Carregado por AWS CLI 'hora' UTC” será usada, em que 'hora' é a hora atual do sistema no Horário Universal Coordenado (UTC).

Você pode usar o AWS CLI para enviar uma revisão para uma implantação do Amazon EC2. Um exemplo comando push é semelhante a:

No Linux, macOS ou Unix:

```
aws deploy push \
  --application-name WordPress_App \
  --description "This is a revision for the application WordPress_App" \
  --ignore-hidden-files \
  --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \
  --source .
```

No Windows:

```
aws deploy push --application-name WordPress_App --description "This is a revision for
 the application WordPress_App" --ignore-hidden-files --s3-location s3://amzn-s3-demo-
bucket/WordPressApp.zip --source .
```

Esse comando faz o seguinte:

- Associa o pacote de arquivos a um aplicativo chamado `WordPress_App`.
- Anexa uma descrição à revisão.
- Ignora arquivos ocultos.
- Nomeia a revisão `WordPressApp.zip` e a envia para um bucket chamado `amzn-s3-demo-bucket`.
- Agrupa todos os arquivos do diretório raiz na revisão.

Depois que o push for bem-sucedido, você poderá usar o console AWS CLI ou o CodeDeploy console para implantar a revisão a partir do Amazon S3. Para implantar essa revisão com o AWS CLI:

No Linux, macOS ou Unix:

```
aws deploy create-deployment \
  --application-name WordPress_App \
  --deployment-config-name your-deployment-config-name \
  --deployment-group-name your-deployment-group-name \
  --s3-location bucket=amzn-s3-demo-bucket,key=WordPressApp.zip,bundleType=zip
```

No Windows:

```
aws deploy create-deployment --application-name WordPress_App --deployment-config-
name your-deployment-config-name --deployment-group-name your-deployment-group-name --
s3-location bucket=amzn-s3-demo-bucket,key=WordPressApp.zip,bundleType=zip
```

Para obter mais informações, consulte Crie uma implantação com CodeDeploy.

Visualize os detalhes da revisão do aplicativo com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre todas as revisões do aplicativo que estão registradas na sua AWS conta para um aplicativo específico.

Para obter informações sobre como registrar uma revisão, consulte Registre uma revisão do aplicativo no Amazon S3 com CodeDeploy.

Tópicos

- Visualizar detalhes de revisões de aplicativo (console)
- Visualizar detalhes de revisões de aplicativo (CLI)

Visualizar detalhes de revisões de aplicativo (console)

Para visualizar detalhes de revisões de aplicativo:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Deploy (Implantar) e selecione Applications (Aplicativos).

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

3. Escolha o nome do aplicativo cuja revisão você deseja visualizar.
4. Na página Application details (Detalhes do aplicativo), selecione a guia Revisions (Revisões) e reveja a lista de revisões registradas para o aplicativo. Selecione uma revisão e, em seguida, selecione View details (Visualizar detalhes).

Visualizar detalhes de revisões de aplicativo (CLI)

Para usar o AWS CLI para visualizar uma revisão do aplicativo, chame o get-application-revisioncomando ou o list-application-revisionscomando.

> **Note**
>
> Referências a serem GitHub aplicadas a implantações somente em EC2/On-Premises implantações. As revisões para AWS Lambda implantações não funcionam com. GitHub

Para visualizar detalhes sobre uma única revisão de aplicativo, chame o comando get-application-revision, especificando:

- O nome do aplicativo. Para obter o nome do aplicativo, chame o comando list-applications.
- Para uma revisão armazenada em GitHub, o nome do GitHub repositório e o ID do commit que faz referência à revisão do aplicativo que foi enviada para o repositório.
- Para uma revisão armazenada no Amazon S3, o nome do bucket do Amazon S3 contendo a revisão, o nome e o tipo de arquivo do arquivamento carregado e, opcionalmente, o identificador de versão do Amazon S3 e a ETag desse arquivamento. Se o identificador de versão, a ETag ou ambos tiverem sido especificados durante uma chamada register-application-revision, eles deverão ser especificados aqui.

Para visualizar detalhes sobre várias revisões de aplicativo, chame o comando list-application-revisions, especificando:

- O nome do aplicativo. Para obter o nome do aplicativo, chame o comando list-applications.
- Opcionalmente, para visualizar detalhes apenas de revisões de aplicativo do Amazon S3, o nome do bucket do Amazon S3 contendo as revisões.
- Opcionalmente, para visualizar detalhes apenas de revisões de aplicativo do Amazon S3, uma string de prefixo para limitar a pesquisa a revisões de aplicativo do Amazon S3. (Se não for especificado, CodeDeploy listará todas as revisões correspondentes do aplicativo Amazon S3.)
- Opcionalmente, seja para listar detalhes de revisão com base em se cada revisão é a revisão de destino de um grupo de implantação. (Se não for especificado, CodeDeploy listará todas as revisões correspondentes.)
- Opcionalmente, o nome da coluna e a ordem com base na qual classificar a lista de detalhes da revisão. (Se não for especificado, CodeDeploy listará os resultados em uma ordem arbitrária.)

É possível listar todas as revisões ou apenas aquelas armazenadas no Amazon S3. Você não pode listar somente as revisões armazenadas em GitHub.

Registre uma revisão do aplicativo no Amazon S3 com CodeDeploy

Se você já tiver chamado o comando push para enviar uma revisão de aplicativo ao Amazon S3, não será necessário registrar a revisão. No entanto, se você fizer upload de uma revisão para o Amazon S3 por outros meios e quiser que a revisão apareça no CodeDeploy console ou por meio do AWS CLI, siga estas etapas para registrar a revisão primeiro.

Se você enviou uma revisão do aplicativo para um GitHub repositório e deseja que a revisão apareça no CodeDeploy console ou por meio do AWS CLI, você também deve seguir estas etapas.

Você pode usar somente as APIs AWS CLI ou as CodeDeploy APIs para registrar revisões de aplicativos no Amazon S3 ou. GitHub

Tópicos

- Registre uma revisão no Amazon S3 com (CodeDeploy CLI)
- Registre uma revisão GitHub com CodeDeploy (CLI)

Registre uma revisão no Amazon S3 com (CodeDeploy CLI)

1. Faça o upload do relatório no Amazon S3.
2. Chame o comando register-application-revision, especificando o seguinte:
   - O nome do aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
   - Informações sobre a revisão a ser registrada:
     - O nome do bucket do Amazon S3 que contém a revisão.
     - O nome e o tipo de arquivo da revisão carregada. Para AWS Lambda implantações, a revisão é um AppSpec arquivo escrito em JSON ou YAML. Para EC2/On-Premises implantações, a revisão contém uma versão dos arquivos de origem que CodeDeploy serão implantados em suas instâncias ou scripts que CodeDeploy serão executados em suas instâncias.

> **Note**
>
> Os formatos de arquivo tar e tar compactado (.tar e .tar.gz) não são compatíveis com instâncias do Windows Server.

- (Opcional) O identificador de versão do Amazon S3 da revisão. (Se o identificador da versão não for especificado, CodeDeploy usará a versão mais recente.)
- (Opcional) A ETag da revisão. (Se a ETag não for especificada, o CodeDeploy ignorará a validação do objeto.)

- (Opcional) Qualquer descrição que você queira associar à revisão.

As informações sobre uma revisão no Amazon S3 podem ser especificadas na linha de comando, usando essa sintaxe como parte da chamada de register-application-revision. (`version` e `eTag` são opcionais.)

Para um arquivo de revisão para uma EC2/On-Premises implantação:

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|
zip,version=string,eTag=string
```

Para um arquivo de revisão para uma implantação do AWS Lambda:

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

Registre uma revisão GitHub com CodeDeploy (CLI)

> **Note**
>
> AWS Lambda implantações não funcionam com GitHub.

1. Faça o upload da revisão para o seu GitHub repositório.
2. Chame o comando register-application-revision, especificando o seguinte:

   - O nome do aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.

- Informações sobre a revisão a ser registrada:
  - O nome GitHub do usuário ou grupo atribuído ao repositório que contém a revisão, seguido por uma barra (/), seguida pelo nome do repositório.
  - O ID da confirmação que faz referência à revisão no repositório.
- (Opcional) Qualquer descrição que você queira associar à revisão.

As informações sobre uma revisão em GitHub podem ser especificadas na linha de comando, usando essa sintaxe como parte da register-application-revision chamada:

```
--github-location repository=string,commitId=string
```

Trabalhando com implantações em CodeDeploy

Em CodeDeploy, uma implantação é o processo e os componentes envolvidos no processo de instalação de conteúdo em uma ou mais instâncias. Esse conteúdo pode consistir em código, arquivos web e de configuração, executáveis, pacotes, scripts e assim por diante. CodeDeploy implanta conteúdo armazenado em um repositório de origem, de acordo com as regras de configuração que você especificar.

Se você usa a plataforma de EC2/On-Premises computação, duas implantações no mesmo conjunto de instâncias podem ser executadas simultaneamente.

CodeDeploy fornece duas opções de tipo de implantação: implantações no local e blue/green implantações.

- In-place implantação: o aplicativo em cada instância do grupo de implantação é interrompido, a revisão mais recente do aplicativo é instalada e a nova versão do aplicativo é iniciada e validada. Você pode usar um balanceador de carga de forma que cada registro de instância é cancelado durante sua implantação e, em seguida, restaurado para o serviço após a conclusão da implantação. Somente implantações que usam a plataforma de EC2/On-Premises computação podem usar implantações no local. Para obter mais informações sobre implantações no local, consulte Visão geral de uma implantação no local.
- Blue/green implantação: o comportamento da sua implantação depende da plataforma de computação que você usa:
  - Blue/green em uma plataforma de EC2/On-Premises computação: as instâncias em um grupo de implantação (o ambiente original) são substituídas por um conjunto diferente de instâncias (o ambiente substituto) usando estas etapas:
    - As instâncias são provisionadas para o ambiente de substituição.
    - A revisão de aplicativo mais recente será instalada nas instâncias de substituição.
    - Um tempo de espera opcional ocorre para atividades como teste de aplicativos e verificação do sistema.
    - As instâncias no ambiente de substituição são registradas com um ou mais balanceadores de carga do Elastic Load Balancing, fazendo com que o tráfego seja redirecionado para elas. O registro das instâncias no ambiente original é cancelado e pode ser encerrado ou mantido em execução para outros usos.

> **Note**
>
> Se você usa uma plataforma EC2/On-Premises computacional, saiba que as blue/green implantações funcionam somente com instâncias do Amazon EC2.

- Blue/green em uma plataforma computacional AWS Lambda ou Amazon ECS: o tráfego é deslocado em incrementos de acordo com uma configuração de implantação canária, linear ou completa.
- Blue/green implantações por meio de CloudFormation: o tráfego é transferido de seus recursos atuais para seus recursos atualizados como parte de uma atualização de CloudFormation pilha. Atualmente, somente blue/green implantações do ECS são suportadas.

Para obter mais informações sobre blue/green implantações, consulteVisão geral de uma blue/green implantação.

Para obter informações sobre a implantação automática do Amazon S3, consulte Implantar automaticamente do Amazon S3 usando. CodeDeploy

Tópicos

- Crie uma implantação com CodeDeploy
- Exibir detalhes CodeDeploy da implantação
- Exibir dados de registro para CodeDeploy EC2/On-Premises implantações
- Interrompa uma implantação com CodeDeploy
- Reimplante e reverta uma implantação com CodeDeploy
- Implemente um aplicativo em outro AWS account
- Use o CodeDeploy agente para validar um pacote de implantação em uma máquina local

Crie uma implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para criar uma implantação que instale revisões de aplicativos que você já enviou para o Amazon S3 ou, se sua implantação for em uma plataforma EC2/On-Premises computacional GitHub, nas instâncias de um grupo de implantação.

O processo para criar uma implantação depende da plataforma de computação usada para a sua implantação.

Tópicos

- Pré-requisitos de implantação
- Criar uma implantação da plataforma de computação do Amazon ECS (console)
- Crie um AWS Implantação da Lambda Compute Platform (console)
- Crie uma implantação da EC2/On-Premises Compute Platform (console)
- Crie uma implantação da Amazon ECS Compute Platform (CLI)
- Crie um AWS Implantação da Lambda Compute Platform (CLI)
- Crie uma implantação da EC2/On-Premises Compute Platform (CLI)
- Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation

Pré-requisitos de implantação

Verifique se as etapas a seguir estão concluídas para iniciar uma implantação.

Pré-requisitos de implantação em um AWS Lambda Plataforma de computação

- Crie um aplicativo que inclua pelo menos um grupo de implantação. Para obter informações, consulte Crie um aplicativo com CodeDeploy e Crie um grupo de implantação com CodeDeploy.
- Prepare a revisão do aplicativo, também conhecida como AppSpec arquivo, que especifica a versão da função Lambda que você deseja implantar. O AppSpec arquivo também pode especificar funções do Lambda para validar sua implantação. Para obter mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.
- Para usar uma configuração de implantação personalizada para a sua implantação, crie-a antes de iniciar o processo de implantação. Para mais informações, consulte Create a Deployment Configuration.

Pré-requisitos de implantação em uma plataforma computacional EC2/on-premises

- Para uma implantação no local, crie ou configure as instâncias nas quais você deseja implantar. Para mais informações, consulte Trabalhando com instâncias para CodeDeploy. Para uma blue/green implantação, você tem um grupo existente do Amazon EC2 Auto Scaling para usar como modelo para seu ambiente substituto, ou você tem uma ou mais instâncias ou grupos do Amazon

EC2 Auto Scaling que você especifica como seu ambiente original. Para obter mais informações, consulte Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling e Integração CodeDeploy com o Amazon EC2 Auto Scaling.

- Crie um aplicativo que inclua pelo menos um grupo de implantação. Para obter informações, consulte Crie um aplicativo com CodeDeploy e Crie um grupo de implantação com CodeDeploy.
- Prepare a revisão do aplicativo que deseja implantar nas instâncias do seu grupo de implantação. Para mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.
- Para usar uma configuração de implantação personalizada para a sua implantação, crie-a antes de iniciar o processo de implantação. Para mais informações, consulte Create a Deployment Configuration.
- Se você estiver implantando a revisão do seu aplicativo a partir de um bucket do Amazon S3, o bucket estará na AWS mesma região das instâncias do seu grupo de implantação.
- Se você estiver implantando sua revisão de aplicativo de um bucket do Amazon S3, uma política de bucket do Amazon S3 foi aplicada a ele. Essa política concede às suas instâncias as permissões necessárias para baixar a revisão de aplicativo.

  Por exemplo, a seguinte política de bucket do Amazon S3 permite que qualquer instância do Amazon EC2 com um perfil de instância anexado do IAM contendo o ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` para fazer o download de qualquer lugar no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`.

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:role/CodeDeployDemo"
                ]
            }
        }
    ]
}
```

A seguinte política do bucket Amazon S3 permite que qualquer instância On-Premises com um usuário do IAM associado, contendo o ARN do `arn:aws:iam::444455556666:user/CodeDeployUser` para que seja feito o download de qualquer lugar no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`.

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:user/CodeDeployUser"
                ]
            }
        }
    ]
}
```

Para obter informações sobre como gerar e anexar uma política de bucket do Amazon S3, consulte Exemplos de políticas de bucket.

- Se você estiver criando uma blue/green implantação ou tiver especificado um Classic Load Balancer, Application Load Balancer ou Network Load Balancer opcional no grupo de implantação para uma implantação local, você criou uma VPC usando a Amazon VPC que contém pelo menos duas sub-redes. (CodeDeploy usa o Elastic Load Balancing, que exige que todas as instâncias em um grupo de balanceadores de carga estejam em uma única VPC.)

  Se você ainda não criou uma VPC, consulte o Guia de conceitos básicos da Amazon VPC.
- Se estiver criando uma blue/green implantação, você configurou pelo menos um Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing e o usou para registrar as instâncias que compõem seu ambiente original.

> **Note**
>
> As instâncias no seu ambiente de substituição serão registradas nesse balanceador de carga mais tarde.

Para obter mais informações sobre as opções de configuração do balanceador de carga, consulte Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2 e Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS.

Pré-requisitos de implantação para uma implantação por meio de blue/green CloudFormation

- Seu modelo não precisa modelar recursos para um CodeDeploy aplicativo ou grupo de implantação.
- Seu modelo deve incluir recursos para uma VPC usando a Amazon VPC que contenha pelo menos duas sub-redes.
- Seu modelo deve incluir recursos para um ou mais Classic Load Balancer, Application Load Balancer ou Network Load Balancer no Elastic Load Balancing que são usados para direcionar o tráfego para seus grupos-alvo.

Criar uma implantação da plataforma de computação do Amazon ECS (console)

Este tópico mostra como implantar um Amazon ECS usando o console. Para obter mais informações, consulte Tutorial: Implantar um aplicativo no Amazon ECS e Tutorial: Implantar um serviço do Amazon ECS com um teste de validação.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. Execute um destes procedimentos:

   - Se você desejar implantar um aplicativo, no painel de navegação, expanda Implantar e escolha Aplicativos. Escolha o nome do aplicativo que você deseja implantar. Certifique-se de que a coluna Plataforma de computação para o seu aplicativo seja Amazon ECS.
   - Se você deseja reimplantar uma implantação, no painel de navegação, expanda Implantar e escolha Implantações. Escolha a implantação que você deseja reimplantar e, na coluna Aplicação, escolha o nome do aplicativo. Certifique-se de que a coluna Plataforma de computação para a sua implantação seja Amazon ECS.

3. Na guia Implantações, selecione Criar implantação.

> **Note**
>
> Seu aplicativo deve ter um grupo de implantação antes de poder ser implantado. Se seu aplicativo não tiver um grupo de implantação, na guia Grupos de implantação, escolha Criar grupo de implantação. Para obter mais informações, consulte Crie um grupo de implantação com CodeDeploy.

4. Em Grupo de implantação, escolha um grupo de implantação para usar nesta implantação.
5. Ao lado de Local de revisão, escolha onde a sua revisão está localizada:

   - Meu aplicativo está armazenado no Amazon S3: para obter informações, consulte Especificar informações sobre uma revisão armazenada em um bucket do Amazon S3 e em seguida retorne à etapa 6.
   - Usar AppSpec editor — Selecione JSON ou YAML e, em seguida, insira seu AppSpec arquivo no editor. Você pode salvar o AppSpec arquivo escolhendo Salvar como arquivo de texto. Ao escolher Implantar no final dessas etapas, você receberá um erro se o JSON ou o YAML não for válido. Para obter mais informações sobre a criação de um AppSpec arquivo, consulteAdicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

6. (Opcional) Em Descrição da implantação, digite uma descrição para esta implantação.

7. (Opcional) Em Substituições de configurações de reversão, você pode especificar opções de reversão automática para essa implantação diferentes daquelas que foram especificadas para o grupo de implantação, se houver.

   Para obter informações sobre reversões em CodeDeploy, consulte e. Reimplantações e reversões de implantação Reimplante e reverta uma implantação com CodeDeploy

   Escolha uma das seguintes opções:

   - Reverta quando uma implantação falha — CodeDeploy reimplanta a última revisão válida conhecida como uma nova implantação.
   - Reverter quando os limites de alarme forem atingidos — Se os alarmes foram adicionados ao grupo de implantação, CodeDeploy reimplanta a última revisão válida quando um ou mais dos alarmes especificados forem ativados.
   - Desativar reversões: não execute reversões para esta implantação.
8. Escolha Criar implantação.

   Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

Crie um AWS Implantação da Lambda Compute Platform (console)

Este tópico mostra como implantar uma função do Lambda usando o console.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. Execute um destes procedimentos:

   - Se você desejar implantar um aplicativo, no painel de navegação, expanda Implantar e escolha Aplicativos. Escolha o nome do aplicativo que você deseja implantar. Certifique-se de que a coluna Plataforma de computação para o seu aplicativo seja AWS Lambda.

- Se você deseja reimplantar uma implantação, no painel de navegação, expanda Implantar e escolha Implantações. Escolha a implantação que você deseja reimplantar e, na coluna Aplicação, escolha o nome do aplicativo. Certifique-se de que a coluna da plataforma de computação para sua implantação seja AWS Lambda.

3. Na guia Implantações, selecione Criar implantação.

> **Note**
>
> Seu aplicativo deve ter um grupo de implantação antes de poder ser implantado. Se seu aplicativo não tiver um grupo de implantação, na guia Grupos de implantação, escolha Criar grupo de implantação. Para obter mais informações, consulte Crie um grupo de implantação com CodeDeploy.

4. Em Grupo de implantação, escolha um grupo de implantação para usar nesta implantação.
5. Ao lado de Local de revisão, escolha onde a sua revisão está localizada:

   - Meu aplicativo está armazenado no Amazon S3: para obter informações, consulte Especificar informações sobre uma revisão armazenada em um bucket do Amazon S3 e em seguida retorne à etapa 6.
   - Usar AppSpec editor — Selecione JSON ou YAML e, em seguida, insira seu AppSpec arquivo no editor. Você pode salvar o AppSpec arquivo escolhendo Salvar como arquivo de texto. Ao escolher Implantar no final dessas etapas, você receberá um erro se o JSON ou o YAML não for válido. Para obter mais informações sobre a criação de um AppSpec arquivo, consulteAdicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy.

6. (Opcional) Em Descrição da implantação, digite uma descrição para esta implantação.
7. (Opcional) Expanda Substituições de grupos de implantação para escolher uma configuração de implantação diferente para controlar como o tráfego é passado para a versão da função do Lambda diferente do que a especificada no grupo de implantação.

   Para obter mais informações, consulte Configurações de implantação em um AWS Lambda Plataforma de computação.
8. (Opcional) Em Substituições de configurações de reversão, você pode especificar opções de reversão automática para essa implantação diferentes daquelas que foram especificadas para o grupo de implantação, se houver.

   Para obter informações sobre reversões em CodeDeploy, consulte e. Reimplantações e reversões de implantação Reimplante e reverta uma implantação com CodeDeploy

Escolha uma das seguintes opções:

- Reverta quando uma implantação falha — CodeDeploy reimplanta a última revisão válida conhecida como uma nova implantação.
- Reverter quando os limites de alarme forem atingidos — Se os alarmes foram adicionados ao grupo de implantação, CodeDeploy reimplanta a última revisão válida quando um ou mais dos alarmes especificados forem ativados.
- Desativar reversões: não execute reversões para esta implantação.

9. Escolha Criar implantação.

   Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

Crie uma implantação da EC2/On-Premises Compute Platform (console)

Este tópico mostra como implantar um aplicativo em um servidor do Amazon EC2 ou on-premises usando o console.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. Execute um destes procedimentos:

   - Se você desejar implantar um aplicativo, no painel de navegação, expanda Implantar e escolha Aplicativos. Escolha o nome do aplicativo que você deseja implantar. Verifique se a coluna Plataforma de computação do aplicativo é EC2/On-Premises.
   - Se você deseja reimplantar uma implantação, no painel de navegação, expanda Implantar e escolha Implantações. Localize a implantação que você deseja reimplantar e escolha o nome do aplicativo na coluna Application (Aplicativos). Certifique-se de que a coluna Compute platform (Plataforma de computação) para a sua implantação seja EC2/On-Premises.

3. Na guia Implantações, selecione Criar implantação.

> **Note**
>
> Seu aplicativo deve ter um grupo de implantação antes de poder ser implantado. Se seu aplicativo não tiver um grupo de implantação, na guia Grupos de implantação, escolha Criar grupo de implantação. Para obter mais informações, consulte Crie um grupo de implantação com CodeDeploy.

4. Em Grupo de implantação, escolha um grupo de implantação para usar nesta implantação.
5. Ao lado de Repository type (Tipo de repositório), escolha o tipo de repositório no qual a sua revisão está armazenada:

   - Meu aplicativo está armazenado no Amazon S3: para obter informações, consulte Especificar informações sobre uma revisão armazenada em um bucket do Amazon S3 e em seguida retorne à etapa 6.
   - Meu aplicativo está armazenado em GitHub — Para obter informações, consulte eEspecificar informações sobre uma revisão armazenada em um GitHub repositório, em seguida, retorne à etapa 6.

6. (Opcional) Em Descrição da implantação, digite uma descrição para esta implantação.
7. (Opcional) Expanda Substituir a configuração de implantação para escolher uma configuração de para controlar a forma como o tráfego é deslocado para o servidor Amazon EC2 ou on-premises diferente daquele especificado no grupo de implantação.

   Para obter mais informações, consulte Trabalhando com configurações de implantação em CodeDeploy.

8. a. Selecione Não falhe na implantação se o evento do ApplicationStop ciclo de vida falhar se você quiser que a implantação em uma instância seja bem-sucedida se o evento do `ApplicationStop` ciclo de vida falhar.

   b. Expanda Configurações adicionais de comportamento de implantação para especificar como CodeDeploy manipula arquivos em um local de destino de implantação que não faziam parte da implantação anterior bem-sucedida.

      Escolha uma das seguintes opções:

      - Falha na implantação: um erro é informado, e o status da implantação é alterado para `Failed`.

- Substituir o conteúdo: se um arquivo com o mesmo nome existir na localização de destino, a versão da revisão do aplicativo o substituirá.
- Reter o conteúdo: se um arquivo com o mesmo nome existir no local de destino, ele será mantido e a versão na revisão do aplicativo não será copiada para a instância.

Para obter mais informações, consulte Comportamento de reversão com conteúdo existente.

9. (Opcional) Em Substituições de configurações de reversão, você pode especificar opções de reversão automática para essa implantação diferentes daquelas que foram especificadas para o grupo de implantação, se houver.

   Para obter informações sobre reversões em CodeDeploy, consulte e. Reimplantações e reversões de implantação Reimplante e reverta uma implantação com CodeDeploy

   Escolha uma das seguintes opções:

   - Reverta quando uma implantação falha — CodeDeploy reimplanta a última revisão válida conhecida como uma nova implantação.
   - Reverter quando os limites de alarme forem atingidos — Se os alarmes foram adicionados ao grupo de CodeDeploy implantação, implanta a última revisão válida quando um ou mais dos alarmes especificados forem ativados.
   - Desativar reversões: não execute reversões para esta implantação.
10. Selecione Iniciar implantação.

    Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

Tópicos

- Especificar informações sobre uma revisão armazenada em um bucket do Amazon S3
- Especificar informações sobre uma revisão armazenada em um GitHub repositório

Especificar informações sobre uma revisão armazenada em um bucket do Amazon S3

Se você estiver seguindo as etapas em Crie uma implantação da EC2/On-Premises Compute Platform (console), siga estas etapas para adicionar detalhes sobre uma revisão de aplicativo armazenada em um bucket do Amazon S3.

1. Copie o link para o Amazon S3 de sua revisão na caixa Local de revisão. Para encontrar o valor do link:

   a. Em uma guia separada do navegador:

      Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/

      Navegue até a sua revisão e escolha-a.

   b. Se o painel Propriedades não estiver visível, escolha o botão Propriedades.

   c. No painel Propriedades, copie o valor do campo Link na caixa Local da revisão no CodeDeploy console.

   Para especificar uma ETag (uma soma de verificação de arquivo) como parte da localização da revisão:

   - Se o valor do campo Link terminar em **`?versionId=`*`versionId`***, adicione **`&etag=`** e a ETag ao final do valor do campo Link.
   - Se o valor do campo Link não especificar um ID de versão, adicione **`?etag=`** e a ETag ao final do valor do campo Link.

   > **Note**
   >
   > Embora não seja tão fácil quanto copiar o valor do campo Link, você também pode digitar a localização da revisão em um dos seguintes formatos:
   > **`s3://`*`bucket-name`*`/`*`folders`*`/`*`objectName`***
   > **`s3://`*`bucket-name`*`/`*`folders`*`/`*`objectName`*`?versionId=`*`versionId`***
   > **`s3://`*`bucket-name`*`/`*`folders`*`/`*`objectName`*`?etag=`*`etag`***
   > **`s3://`*`bucket-name`*`/`*`folders`*`/`*`objectName`*`?versionId=`*`versionId`*`&etag=`*`etag`***
   > ***`bucket-name`*`.s3.amazonaws.com/`*`folders`*`/`*`objectName`***

2. Se for exibida uma mensagem na lista Tipo de arquivo especificando que o tipo de arquivo não pôde ser detectado, escolha o tipo de arquivo da revisão. Caso contrário, aceite o tipo de arquivo detectado.

Especificar informações sobre uma revisão armazenada em um GitHub repositório

Se você estiver seguindo as etapas emCrie uma implantação da EC2/On-Premises Compute Platform (console), siga estas etapas para adicionar detalhes sobre uma revisão do aplicativo armazenada em um GitHub repositório.

1. Em Connect to GitHub, faça o seguinte:

   - Para criar uma conexão de CodeDeploy aplicativos com uma GitHub conta, em uma guia diferente do navegador da Web, saia GitHub. Em GitHub Conta, insira um nome para identificar essa conexão e escolha Conectar GitHub a. A página da web solicita que você autorize CodeDeploy a interação com GitHub seu aplicativo. Continue na etapa 2.
   - Para usar uma conexão que você já criou, na GitHubconta, selecione seu nome e escolha Conectar GitHub a. Continue na etapa 4.
   - Para criar uma conexão com uma GitHub conta diferente, em uma guia diferente do navegador da Web, saia da GitHub. Escolha Conectar a uma GitHub conta diferente e, em seguida, escolha Conectar GitHub a. Continue na etapa 2.
2. Se você for solicitado a fazer login GitHub, siga as instruções na página de login. Faça login com seu nome de GitHub usuário ou e-mail e senha.
3. Se uma página para Autorizar aplicativo for exibida, escolha Autorizar aplicativo.
4. Na página Criar implantação, na caixa Nome do repositório, insira o nome GitHub do usuário ou da organização que contém a revisão, seguido por uma barra (/), seguida pelo nome do repositório que contém a revisão. Caso não tenha certeza do valor que deve ser digitado:

   a. Em uma guia diferente do navegador, acesse seu GitHubpainel.

   b. Em Seus repositórios, passe o ponteiro do mouse sobre o nome do repositório de destino. Uma dica de ferramenta aparece, exibindo o nome GitHub do usuário ou da organização, seguido por uma barra (/), seguida pelo nome do repositório. Insira esse valor exibido na caixa Repository name (Nome do repositório).

   > **Note**
   >
   > Se o nome do repositório de destino não estiver visível em Seus repositórios, use a GitHub caixa Pesquisar para encontrar o nome do repositório de destino e o nome GitHub do usuário ou da organização.

5. Na caixa Commit ID (ID de confirmação), insira o ID da confirmação que se refere à revisão no repositório. Caso não tenha certeza do valor que deve ser digitado:

   a. Em uma guia diferente do navegador, acesse seu GitHubpainel.

   b. Em Seus repositórios, escolha o nome do repositório que contém a confirmação de destino.

   c. Na lista de confirmações, localize e copie o ID de confirmação que se refere à revisão no repositório. Em geral, esse ID tem 40 caracteres de comprimento e é formado por letras e números. (Não use a versão mais curta do ID de confirmação, que consiste normalmente nos 10 primeiros caracteres da versão mais longa desse ID de confirmação.)

   d. Cole o ID de confirmação na caixa ID de confirmação.

Crie uma implantação da Amazon ECS Compute Platform (CLI)

Depois de criar o aplicativo e a revisão (nas implantações do Amazon ECS, este é o AppSpec arquivo):

Chame o comando create-deployment, especificando o ID da implantação, especificando:

- Um nome de aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- O nome do grupo da implantação. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.
- Informações sobre a revisão a ser implantada:

  Para revisões armazenadas no Amazon S3:

  - O nome do bucket do Amazon S3 que contém a revisão.
  - O nome da revisão carregada.
  - (Opcional) O identificador de versão do Amazon S3 da revisão. (Se o identificador da versão não for especificado, CodeDeploy usa a versão mais recente.)
  - (Opcional) A ETag da revisão. (Se a ETag não for especificada, o CodeDeploy ignorará a validação do objeto.)

  Para as revisões armazenadas em um arquivo que não está no Amazon S3, você precisa do nome do arquivo e o caminho. O seu arquivo de revisão está escrito usando JSON ou YAML, portanto, provavelmente possui uma extensão .json ou .yaml.
- (Opcional) Uma descrição da implantação.

O arquivo de revisão pode ser especificado como um arquivo carregado em um bucket do Amazon S3 ou como uma string. A sintaxe para cada um quando usado como parte do comando create-deployment é:

- Bucket do Amazon S3:

  Os `version` e `eTag` são opcionais.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- String:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> O comando create-deployment pode carregador uma revisão de um arquivo. Para obter mais informações, consulte Carregar parâmetros de um arquivo.

Para modelos AWS Lambda de revisão de implantação, consulteAdicionar um AppSpec arquivo para um AWS implantação do Lambda. Para obter uma revisão de exemplo, consulte AppSpec Exemplo de arquivo para um AWS implantação do Lambda.

Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

Crie um AWS Implantação da Lambda Compute Platform (CLI)

Depois de criar o aplicativo e a revisão (nas implantações do AWS Lambda, esse é o AppSpec arquivo):

Chame o comando create-deployment, especificando o ID da implantação, especificando:

- Um nome de aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- O nome do grupo da implantação. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.

- Informações sobre a revisão a ser implantada:

  Para revisões armazenadas no Amazon S3:

  - O nome do bucket do Amazon S3 que contém a revisão.
  - O nome da revisão carregada.
  - (Opcional) O identificador de versão do Amazon S3 da revisão. (Se o identificador da versão não for especificado, CodeDeploy usa a versão mais recente.)
  - (Opcional) A ETag da revisão. (Se a ETag não for especificada, o CodeDeploy ignorará a validação do objeto.)

  Para as revisões armazenadas em um arquivo que não está no Amazon S3, você precisa do nome do arquivo e o caminho. O seu arquivo de revisão está escrito usando JSON ou YAML, portanto, provavelmente possui uma extensão .json ou .yaml.
- (Opcional) O nome de uma configuração de implantação a ser usada. Para visualizar uma lista de configurações de implantação, chame o comando list-deployment-configs. (Se não for especificado, CodeDeploy usa uma configuração de implantação padrão específica.)
- (Opcional) Uma descrição da implantação.

O arquivo de revisão pode ser especificado como um arquivo carregado em um bucket do Amazon S3 ou como uma string. A sintaxe para cada um quando usado como parte do comando create-deployment é:

- Bucket do Amazon S3:

  Os `version` e `eTag` são opcionais.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- String:

```
--revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
```

> **Note**
>
> O comando create-deployment pode carregador uma revisão de um arquivo. Para obter mais informações, consulte Carregar parâmetros de um arquivo.

Para modelos AWS Lambda de revisão de implantação, consulteAdicionar um AppSpec arquivo para um AWS implantação do Lambda. Para obter uma revisão de exemplo, consulte AppSpec Exemplo de arquivo para um AWS implantação do Lambda.

Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

Crie uma implantação da EC2/On-Premises Compute Platform (CLI)

Para usar o AWS CLI para implantar uma revisão na plataforma de EC2/On-Premises computação:

1. Depois de ter preparado as instâncias, criado o aplicativo e enviado a revisão, siga um destes procedimentos:
   - Se você deseja implantar uma revisão de um bucket do Amazon S3, continue na etapa 2 agora.
   - Se você quiser implantar uma revisão a partir de um GitHub repositório, primeiro conclua as etapas em eConectar um CodeDeploy aplicativo a um GitHub repositório, em seguida, continue com a etapa 2.
2. Chame o comando create-deployment, especificando o ID da implantação, especificando:
   - `--application-name`: Um nome de aplicativo. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
   - `--deployment-group-name`: Um nome de grupo de implantação do Amazon EC2. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.
   - `--revision`: Informações sobre a revisão a ser implantada:

     Para revisões armazenadas no Amazon S3:

     - `s3Location`: O nome do bucket do Amazon S3 que contém a revisão.
     - `s3Location --> key`: O nome da revisão carregada.
     - `s3Location --> bundleType`: O nome e o tipo de arquivo da revisão carregada.

> **Note**
>
> Os formatos de arquivo tar e tar compactado (.tar e .tar.gz) não são compatíveis com instâncias do Windows Server.

  - `s3Location --> version`: (Opcional) O identificador de versão do Amazon S3 da revisão. (Se o identificador da versão não for especificado, CodeDeploy usa a versão mais recente.)
  - `s3Location --> eTag`: (Opcional) A ETag da revisão. (Se a ETag não for especificada, CodeDeploy ignora a validação do objeto.)

  Para revisões armazenadas em: GitHub

  - `gitHubLocation --> repository`: o nome GitHub do usuário ou grupo atribuído ao repositório que contém a revisão, seguido por uma barra (`/`), seguida pelo nome do repositório.
  - `gitHubLocation --> commitId`: O ID de confirmação da revisão.
- `--deployment-config-name`: (Opcional) O nome de uma configuração de implantação a ser usada. Para visualizar uma lista de configurações de implantação, chame o comando list-deployment-configs. (Se não for especificado, CodeDeploy usa uma configuração de implantação padrão específica.)
- `--ignore-application-stop-failures | --no-ignore-application-stop-failures`: (Opcional) Para que a implantação em uma instância continue para o evento de ciclo de vida da implantação `BeforeInstall` se o evento de ciclo de vida de implantação `ApplicationStop` falhar.
- `--description`: (Opcional) Uma descrição da implantação.
- `--file-exists-behavior`: (Opcional) Como parte do processo de implantação, o CodeDeploy agente remove de cada instância todos os arquivos instalados pela implantação mais recente. Escolha o que acontece quando arquivos que não faziam parte de uma implantação anterior aparecem nos locais de implantação de destino.
- `--target-instances`: Para blue/green implantações, informações sobre as instâncias que pertencem ao ambiente substituto em uma blue/green implantação, incluindo os nomes de um ou mais grupos do Amazon EC2 Auto Scaling ou a chave, o tipo e o valor do filtro de tags usados para identificar instâncias do Amazon EC2.

> **Note**
>
> Use essa sintaxe como parte da chamada create-deployment para especificar informações sobre uma revisão no Amazon S3 diretamente na linha de comando. (`version` e `eTag` são opcionais.)
>
> ```
> --s3-location bucket=string,key=string,bundleType=tar|tgz|
> zip,version=string,eTag=string
> ```
>
> Use essa sintaxe como parte da create-deployment chamada para especificar informações sobre uma revisão GitHub diretamente na linha de comando:
>
> ```
> --github-location repository=string,commitId=string
> ```
>
> Para obter informações sobre revisões que já foram enviadas, chame o comando list-application-revisions.

Para acompanhar o status da sua implantação, consulte Exibir detalhes CodeDeploy da implantação.

referência do comando create-deployment

A estrutura e as opções de comando a seguir estão disponíveis para o `create-deployment` comando. Para obter mais informações, consulte a referência de criação e implantação na Referência de comandos da AWS CLI .

```
create-deployment
--application-name <value>
[--deployment-group-name <value>]
[--revision <value>]
[--deployment-config-name <value>]
[--description <value>]
[--ignore-application-stop-failures | --no-ignore-application-stop-failures]
[--target-instances <value>]
[--auto-rollback-configuration <value>]
[--update-outdated-instances-only | --no-update-outdated-instances-only]
[--file-exists-behavior <value>]
[--s3-location <value>]
[--github-location <value>]
[--cli-input-json <value>]
```

```
[--generate-cli-skeleton <value>]
```

Conectar um CodeDeploy aplicativo a um GitHub repositório

Antes de implantar um aplicativo de um GitHub repositório pela primeira vez usando o AWS CLI, você deve primeiro dar CodeDeploy permissão para interagir com ele GitHub em nome da sua GitHub conta. Essa etapa deve ser concluída uma vez para cada aplicativo usando o CodeDeploy console.

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. Selecione Aplicações.
3. Em Aplicativos, escolha o aplicativo que você deseja vincular à sua conta de GitHub usuário e escolha Implantar aplicativo.

   > **Note**
   >
   > Você não está criando uma implantação. Atualmente, essa é a única maneira de dar CodeDeploy permissão para interagir GitHub em nome de sua conta de GitHub usuário.

4. Ao lado de Tipo de repositório, escolha Minha revisão do aplicativo está armazenada em GitHub.
5. Escolha Connect to GitHub.

   > **Note**
   >
   > Se você ver um link Connect to a uma GitHub conta diferente:
   > Talvez você já tenha autorizado CodeDeploy a interagir GitHub em nome de uma GitHub conta diferente para o aplicativo.
   > Você pode ter revogado a autorização CodeDeploy para interagir GitHub em nome da GitHub conta conectada para todos os aplicativos vinculados a ela. CodeDeploy

Para obter mais informações, consulte GitHub autenticação com aplicativos em CodeDeploy.

6. Se você ainda não estiver conectado GitHub, siga as instruções na página de login.
7. Na página para Autorizar aplicativo, escolha Autorizar aplicativo.
8. Agora que CodeDeploy tem permissão, escolha Cancelar e continue com as etapasCrie uma implantação da EC2/On-Premises Compute Platform (CLI).

Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation

Você pode usar AWS CloudFormation para gerenciar blue/green implantações do Amazon ECS por meio de. CodeDeploy Você gera a implantação definindo seus recursos verdes e azuis e especificando as configurações de roteamento e estabilização de tráfego a serem usadas no CloudFormation. Este tópico aborda as diferenças entre as blue/green implantações do Amazon ECS que são gerenciadas por CodeDeploy e as implantações que são gerenciadas por. CloudFormation

Para obter instruções passo a passo sobre como usar CloudFormation para gerenciar suas implantações do Amazon ECS, consulte Automatizar blue/green implantações do ECS por meio do uso no Guia do usuário blue/green. CodeDeploy AWS CloudFormationAWS CloudFormation

> **Note**
>
> O gerenciamento de blue/green implantações do Amazon ECS com não CloudFormation está disponível na região Ásia-Pacífico (Osaka).

Diferenças entre as blue/green implantações do Amazon ECS por meio de e CodeDeploy CloudFormation

O modelo de CloudFormation pilha modela os recursos e a infraestrutura relacionados às tarefas do Amazon ECS, bem como as opções de configuração para implantações. Portanto, há diferenças entre as implantações padrão do Amazon ECS e blue/green as blue/green implantações criadas por meio do. CloudFormation

Ao contrário das blue/green implantações padrão do Amazon ECS, você não modela nem cria manualmente o seguinte:

- Você não cria um AWS CodeDeploy aplicativo especificando um nome que represente de forma exclusiva o que você deseja implantar.
- Você não cria um grupo AWS CodeDeploy de implantação.
- Você não especifica um arquivo de especificação do aplicativo (AppSpec arquivo). As informações normalmente gerenciadas com o AppSpec arquivo, como as opções ponderadas de configuração ou os eventos do ciclo de vida, são gerenciadas pelo gancho. `AWS::CodeDeploy::BlueGreen`

Esta tabela resume as diferenças no fluxo de trabalho de alto nível entre os tipos de implantação.

Função	blue/green Implantações padrão	Blue/green implantações por meio de CloudFormation
Especifique o cluster do Amazon ECS, o serviço Amazon ECS, o Application Load Balancer ou o Network Load Balancer, o receptor de produção, o receptor de teste e dois grupos-alvo.	Crie um grupo CodeDeploy de implantação que especifique esses recursos.	Crie um CloudFormation modelo para modelar esses recursos.
Especifique a alteração a ser implantada.	Crie um CodeDeploy aplicativo.	Crie um CloudFormation modelo que especifique a imagem do contêiner.
Especifique a definição da tarefa do Amazon ECS, o nome do contêiner e a porta do contêiner.	Crie um AppSpec arquivo que especifique esses recursos.	Crie um CloudFormation modelo para modelar esses recursos.
Especifique as opções de deslocamento de tráfego de implantação e os ganchos de eventos de ciclo de vida.	Crie um AppSpec arquivo que especifique essas opções.	Crie um CloudFormation modelo que use os parâmetros do `AWS::CodeDeploy::BlueGreen` gancho para especificar essas opções.
CloudWatch alarmes.	Crie um CloudWatch alarme que acione uma reversão.	Configure um CloudWatch alarme no nível da CloudForm

Função	blue/green Implantações padrão	Blue/green implantações por meio de CloudFormation
		ation pilha que acione uma reversão.
Rollback/redeployment.	Especifique as opções de reversão e reimplantação.	Cancele a atualização da pilha em CloudFormation.

Monitorando blue/green implantações do Amazon ECS por meio de CloudFormation

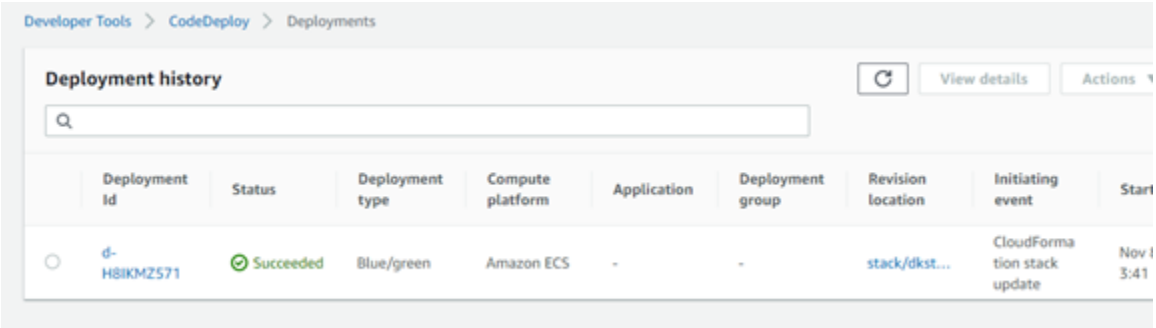
Você pode monitorar blue/green as implantações por meio de CloudFormation e. CodeDeploy Para obter informações sobre o monitoramento por meio de CloudFormation, consulte Monitoramento de blue/green eventos CloudFormation no Guia AWS CloudFormation do usuário.

Para ver o status de implantação das blue/green implantações em CodeDeploy

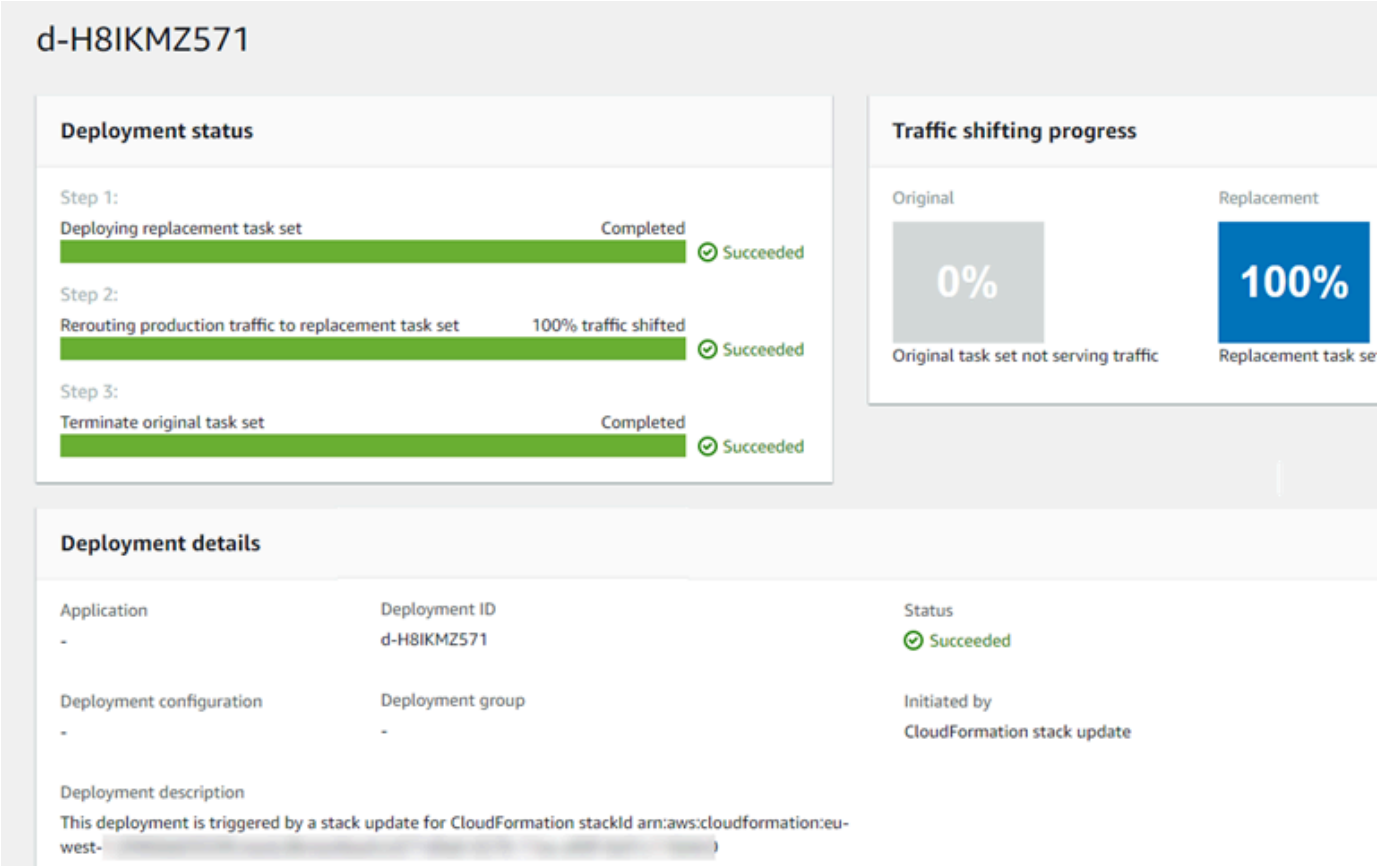
1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. Em Implantações, a implantação que foi acionada pela atualização da CloudFormation pilha é exibida. Escolha a implantação para visualizar o Histórico de implantação.



3. Escolha a implantação para visualizar o status do deslocamento do tráfego. Observe que o aplicativo e o grupo de implantação não são criados.



4. Os itens a seguir se aplicam para reverter ou interromper a implantação:

   - A implantação bem-sucedida aparece em CodeDeploy e mostra que a implantação foi iniciada por CloudFormation.
   - Se quiser interromper e reverter a implantação, você deve cancelar a atualização da pilha em CloudFormation.

Exibir detalhes CodeDeploy da implantação

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para ver detalhes sobre implantações associadas à sua AWS conta.

> **Note**
>
> Você pode ver os registros de EC2/On-Premises implantação em suas instâncias nos seguintes locais:

- Amazon Linux, RHEL e Ubuntu Server: `/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`
- Servidor Windows: C:\ProgramData\ Amazon\ CodeDeploy < DEPLOYMENT-GROUP-ID >< DEPLOYMENT-ID >\ logs\ scripts.log

Para obter mais informações, consulte Analisando arquivos de log para investigar falhas de implantação em instâncias.

Tópicos

- Visualizar detalhes de implantações (console)
- Visualizar detalhes de implantações (CLI)

Visualizar detalhes de implantações (console)

Para usar o CodeDeploy console para ver os detalhes da implantação:

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Implantações.

   > **Note**
   >
   > Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

3. Para ver mais detalhes de uma única implantação, em Deployment history (Histórico de implantação), escolha o ID da implantação ou escolha o botão próximo ao ID da implantação e View (Visualizar).

Visualizar detalhes de implantações (CLI)

Para usar o AWS CLI para visualizar os detalhes da implantação, chame o `get-deployment` comando ou o `batch-get-deployments` comando. É possível chamar o comando `list-deployments` para obter uma lista de IDs de implantação exclusivos a serem usados como entradas para o comando `get-deployment` e o comando `batch-get-deployments`.

Para ver detalhes sobre uma única implantação, chame o comando get-deployment, especificando o identificador de implantação exclusivo. Para obter o ID de implantação, chame o comando list-deployments.

Para visualizar detalhes sobre várias implantações, chame o comando batch-get-deployments, especificando vários identificadores de implantação exclusivos. Para obter os IDs de implantação, chame o comando list-deployments.

Para ver uma lista de IDs de implantação, chame o comando list-deployments especificando:

- O nome do aplicativo associado à implantação. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- O nome do grupo de implementação associado à implantação. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.
- Opcionalmente, se você deseja incluir detalhes sobre implantações de acordo com seu status de implantação. (Se não for especificado, todas as implantações correspondentes serão listadas, independentemente do status de implantação.)
- Opcionalmente, se você deseja incluir detalhes sobre implementações de acordo com suas horas de início e/ou de término de criação. (Se não for especificado, todas as implantações correspondentes serão listadas, independentemente da hora de criação.)

Exibir dados de registro para CodeDeploy EC2/On-Premises implantações

Você pode visualizar os dados de log criados por uma CodeDeploy implantação configurando o CloudWatch agente da Amazon para visualizar dados agregados no CloudWatch console ou fazendo login em uma instância individual para revisar o arquivo de log.

> **Note**
>
> Os registros não são compatíveis com implantações do AWS Lambda ou do Amazon ECS. Eles podem ser criados somente para EC2/On-Premises implantações.

Tópicos

- Visualize dados do arquivo de log no CloudWatch console da Amazon
- Visualizar arquivos de log em uma instância

Visualize dados do arquivo de log no CloudWatch console da Amazon

Quando o CloudWatch agente da Amazon é instalado em uma instância, os dados de implantação de todas as implantações dessa instância ficam disponíveis para visualização no CloudWatch console. Para simplificar, recomendamos usar CloudWatch para monitorar centralmente os arquivos de log em vez de visualizá-los instância por instância. Para obter mais informações, consulte Enviar registros CodeDeploy do agente para CloudWatch.

Visualizar arquivos de log em uma instância

Para visualizar dados de log de implantação para uma instância individual, você pode entrar nessa instância e procurar informações sobre erros ou outros eventos de implantação.

Tópicos

- Para visualizar arquivos de log de implantação em instâncias do Amazon Linux, RHEL e Ubuntu Server
- Para visualizar arquivos de logs de implantação em instâncias do Windows Server

Para visualizar arquivos de log de implantação em instâncias do Amazon Linux, RHEL e Ubuntu Server

Em instâncias do Amazon Linux, do RHEL e do Ubuntu Server, os logs de implantação são armazenados no seguinte local:

`/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`

Para visualizar ou analisar registros de implantação em instâncias do Amazon Linux, RHEL e Ubuntu Server, faça login na instância e digite o seguinte comando para abrir o arquivo de log do CodeDeploy agente:

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Para a versão 2.0.x e posterior, o log do agente gira diariamente ou quando o arquivo atinge 64 MB (configurável). Os arquivos rotacionados são nomeados `codedeploy-agent.YYYYMMDD.log` (com um sufixo numérico para rotações no mesmo dia) e são removidos após sete dias. O script log (`scripts.log`) tem um tamanho alternado de 64 MB com até 8 arquivos arquivados (configuráveis).

Digite os seguintes comandos para ver se há mensagens de erro no arquivo de log:

Command	Resultado
`& ERROR`	Mostrar apenas as mensagens de erro no arquivo de log. Use um único espaço antes e depois da palavra **`ERROR`**.
`/ ERROR`	Procurar a próxima mensagem de erro.[1]
`? ERROR`	Pesquise a mensagem de erro anterior.[2] Use um único espaço antes e depois da palavra. **`ERROR`**
`G`	Ir para o final do arquivo de log.
`g`	Ir para o início do arquivo de log.
`q`	Sair do arquivo de log.

Command	Resultado
h	Saber mais sobre comandos adicionais.

[1] Depois de digitar **/ ERROR** , digite **n** para a próxima mensagem de erro. Digite **N** para a mensagem de erro anterior.

[2] Depois de digitar **? ERROR** , digite **n** para a próxima mensagem de erro ou digite **N** para a mensagem de erro anterior.

Você também pode digitar o seguinte comando para abrir um arquivo de log de CodeDeploy scripts:

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/
scripts.log
```

Digite os seguintes comandos para ver se há mensagens de erro no arquivo de log:

Command	Resultado
&stderr	Mostrar apenas as mensagens de erro no arquivo de log.
/stderr	Procurar a próxima mensagem de erro.[1]
?stderr	Procurar a mensagem de erro anterior.
G	Ir para o final do arquivo de log.
g	Ir para o início do arquivo de log.
q	Sair do arquivo de log.
h	Saber mais sobre comandos adicionais.

[1]Depois de digitar **/stderr**, digite **n** para a próxima mensagem de erro adiante. Digite **N** para a mensagem de erro anterior retroativamente.

[2] Depois de digitar **?stderr**, digite **n** para a próxima mensagem de erro anterior. Digite **N** para a mensagem de erro anterior adiante.

Para visualizar arquivos de logs de implantação em instâncias do Windows Server

CodeDeploy arquivo de log do agente: nas instâncias do Windows Server, o arquivo de log do CodeDeploy agente é armazenado no seguinte local:

Para a versão 2.0.x e posterior:

`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log`

Para a versão 1.8.x e anteriores:

`C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt`

Para visualizar ou analisar o arquivo de log do CodeDeploy agente em uma instância do Windows Server, entre na instância e digite o seguinte comando para abrir o arquivo:

Para a versão 2.0.x e posterior:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Para a versão 1.8.x e anteriores:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Para procurar mensagens de erro no arquivo de log, pressione CTRL+F, digite **ERROR** e, em seguida, pressione Enter para encontrar o primeiro erro.

CodeDeploy arquivos de log de scripts: nas instâncias do Windows Server, os registros de implantação são armazenados no seguinte local:

`C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\logs\scripts.log`

Em que:

- *`deployment-group-id`*é uma string como `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *`deployment-id`*é um identificador como `d-12EXAMPLE`

Digite o comando a seguir para abrir um arquivo de log de CodeDeploy scripts:

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs
\scripts.log
```

Para a versão 2.0.x e posterior, o log do agente gira diariamente ou quando o arquivo atinge 64 MB (configurável). O registro de scripts é alternado em 64 MB com até 8 arquivos arquivados (configuráveis).

Para procurar mensagens de erro no arquivo de log, pressione CTRL+F, digite **`stderr`** e, em seguida, pressione Enter para encontrar o primeiro erro.

Interrompa uma implantação com CodeDeploy

Você pode usar o CodeDeploy console AWS CLI, o ou as CodeDeploy APIs para interromper as implantações associadas à sua AWS conta.

> **Warning**
>
> Interromper uma EC2/On-Premises implantação pode deixar algumas ou todas as instâncias em seus grupos de implantação em um estado de implantação indeterminado. Para obter mais informações, consulte Implantações interrompidas e com falha.

Você pode interromper uma implantação ou interromper e reverter uma implantação.

- Interromper uma implantação (console)
- Interromper uma implantação (CLI)

> **Note**
>
> Se sua implantação for uma blue/green implantação completa CloudFormation, você não poderá executar essa tarefa no CodeDeploy console. Vá até o CloudFormation console para realizar essa tarefa.

Interromper uma implantação (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Implantações.

> **Note**
>
> Se nenhuma entrada for exibida, verifique se a região correta está selecionada. Na barra de navegação, no seletor de região, escolha uma das regiões listadas em Região e Endpoints no. Referência geral da AWS CodeDeploy é suportado somente nessas regiões.

3. Escolha a implantação que você deseja interromper faça um dos seguintes:

   1. Escolha Stop deployment (Interromper implantação) para interromper a implantação sem uma reversão.
   2. Escolha Stop and roll back deployment (Interromper e reverter implantação) para interromper e reverter a implantação.

   Para obter mais informações, consulte Reimplante e reverta uma implantação com CodeDeploy.

> **Note**
>
> Se Stop deployment (Interromper uma implantação) e Stop and roll back deployment (Interromper e reverter implantação) não estiverem disponíveis, a implantação progrediu até um ponto em que não pode ser interrompida.

Interromper uma implantação (CLI)

Chame o comando stop-development, especificando o ID da implantação. Para ver uma lista de IDs de implantação, chame o comando list-deployments.

Reimplante e reverta uma implantação com CodeDeploy

CodeDeploy reverte as implantações reimplantando uma revisão implantada anteriormente de um aplicativo como uma nova implantação. Essas implantações revertidas são tecnicamente novas implantações, com novos ID de implantação, e não versões restauradas de uma implantação anterior.

Implantações podem ser revertidas automaticamente ou manualmente.

Tópicos

- Reversões automáticas
- Reversões manuais
- Fluxo de trabalho de reversão e reimplantação
- Comportamento de reversão com conteúdo existente

Reversões automáticas

Você pode configurar um grupo de implantação ou uma implantação para reversão automática quando uma implantação falhar ou quando um limite de monitoramento especificado for atendido. Nesse caso, a última versão conhecida em bom estado de uma revisão de aplicativo é implantada. Reversões automáticas são configuradas quando um aplicativo é criado ou quando um grupo de implantação é criado ou atualizado.

Ao criar uma nova implantação, você também pode optar por substituir a configuração de reversão automática que foi especificada para o grupo de implantação.

> **Note**
>
> Você pode usar o Amazon Simple Notification Service para receber uma notificação sempre que uma implantação for revertida automaticamente. Para mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications.

Para obter mais informações sobre como configurar reversões automáticas, consulte Configurar opções avançadas para um grupo de implantação.

Reversões manuais

Se você não tiver configurado reversões automáticas, poderá reverter manualmente uma implantação criando uma nova implantação que usa qualquer revisão de aplicativo implantada anteriormente e seguindo as etapas para reimplementar uma revisão. Isso pode ser feito quando um aplicativo entra em um estado desconhecido. Em vez de gastar muito tempo com a solução de problemas, você pode reimplantar o aplicativo para um bom estado de operação conhecido. Para obter mais informações, consulte Crie uma implantação com CodeDeploy.

> **Note**
>
> Se você remover uma instância de um grupo de implantação, CodeDeploy não desinstalará nada que já tenha sido instalado nessa instância.

Fluxo de trabalho de reversão e reimplantação

Quando a reversão automática é iniciada, ou quando você inicia manualmente uma reimplantação ou reversão manual, CodeDeploy primeiro tenta remover de cada instância participante todos os arquivos que foram instalados com sucesso pela última vez. CodeDeploy faz isso verificando o arquivo de limpeza:

`/opt/codedeploy-agent/deployment-root/deployment-instructions/`*`deployment-group-ID`*`-cleanup`arquivo (para instâncias Amazon Linux, etc.)

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`*`deployment-group-ID`*`-cleanup`arquivo (para instâncias do Windows Server)

Se ele existir, CodeDeploy usa o arquivo de limpeza para remover da instância todos os arquivos listados antes de iniciar a nova implantação.

Por exemplo, os dois primeiros arquivos de texto e os dois arquivos de script já foram implantados em uma instância do Amazon EC2 que executa o Windows Server, e os scripts criaram mais dois arquivos de texto durante os eventos de ciclo de vida de implantação:

```
c:\temp\a.txt (previously deployed by CodeDeploy)
c:\temp\b.txt (previously deployed by CodeDeploy)
c:\temp\c.bat (previously deployed by CodeDeploy)
c:\temp\d.bat (previously deployed by CodeDeploy)
c:\temp\e.txt (previously created by c.bat)
```

```
c:\temp\f.txt (previously created by d.bat)
```

O arquivo de limpeza listará apenas os dois primeiros arquivos de texto e dois arquivos de script:

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
c:\temp\d.bat
```

Antes da nova implantação, CodeDeploy removerá somente os dois primeiros arquivos de texto e os dois arquivos de script, deixando os dois últimos arquivos de texto intocados:

```
c:\temp\a.txt will be removed
c:\temp\b.txt will be removed
c:\temp\c.bat will be removed
c:\temp\d.bat will be removed
c:\temp\e.txt will remain
c:\temp\f.txt will remain
```

Como parte desse processo, não CodeDeploy tentará reverter ou reconciliar nenhuma ação realizada por nenhum script em implantações anteriores durante reimplantações subsequentes, sejam reversões manuais ou automáticas. Por exemplo, se os `d.bat` arquivos `c.bat` e contiverem lógica para não recriar os `f.txt` arquivos `e.txt` e se eles já existirem, as versões antigas de `e.txt` e `f.txt` permanecerão inalteradas sempre que forem CodeDeploy executadas `c.bat` e `d.bat` em implantações subsequentes. Você pode adicionar lógica a `c.bat` e `d.bat` para sempre verificar e excluir versões antigas de `e.txt` e `f.txt` antes de criar novas.

Comportamento de reversão com conteúdo existente

Como parte do processo de implantação, o CodeDeploy agente remove de cada instância todos os arquivos que a implantação mais recente instalou. Se os arquivos que não faziam parte de uma implantação anterior aparecerem nos locais de implantação de destino, você poderá escolher o CodeDeploy que fazer com eles durante a próxima implantação:

- Falha na implantação: um erro é informado, e o status da implantação é alterado para Failed.
- Substituir o conteúdo: a versão do arquivo da revisão de aplicativo substitui a versão que já está na instância.
- Reter o conteúdo: o arquivo no local de destino é mantido, e a versão na revisão do aplicativo não é copiada para a instância.

É possível escolher esse comportamento ao criar uma implantação. Criar uma configuração de implantação no console, consulte Crie uma implantação da EC2/On-Premises Compute Platform (console). Se estiver criando uma implantação com o AWS CLI, consulteCrie uma implantação da EC2/On-Premises Compute Platform (CLI).

Você pode optar por manter os arquivos que deseja que façam parte da próxima implantação sem precisar adicioná-los ao pacote de revisão de aplicativo. Por exemplo, você pode fazer upload de arquivos diretamente para a instância que são necessários para a implantação, mas que não foram adicionados ao pacote de revisão do aplicativo. Ou você pode fazer upload de arquivos para a instância se seus aplicativos já estiverem em seu ambiente de produção, mas você quiser CodeDeploy usá-los pela primeira vez para implantá-los.

No caso de reversões em que a mais recente revisão de aplicativo implantada com êxito for reimplantada devido a uma falha de implantação, a opção de tratamento de conteúdo para essa última implementação bem-sucedida será aplicada à implantação de reversão.

No entanto, se a implantação que falhou tiver sido configurada para substituir arquivos, em vez de mantê-los, um resultado inesperado poderá ocorrer durante a reversão. Especificamente, os arquivos que você esperava que fossem mantidos talvez sejam removidos pela implantação com falha. Os arquivos não estão na instância quando a implantação de reversão é executada.

No exemplo a seguir, existem três implementações. Nenhum arquivo substituído (excluído) durante a segunda implementação com falha estará mais disponível (não pode ser mantido) quando a revisão de aplicativo 1 for implantada novamente durante a implantação 3:

Implantação	Revisão de aplicativo	Opção de substituição de conteúdo	Estado da implantação	Comportamento e resultado
implantação 1	revisão de aplicativo 1	MANTER	Bem-sucedido	CodeDeploy detecta arquivos nos locais de destino que não foram implantados na implantação anterior. Esses arquivos podem ter sido

Implantação	Revisão de aplicativo	Opção de substituição de conteúdo	Estado da implantação	Comportamento e resultado
				colocados lá intencionalmente para se tornarem parte da implantação atual. Eles são mantidos e registrados como parte do pacote de implantação atual.
implantação 2	revisão de aplicativo 2	SUBSTITUIR	Failed	Durante o processo de implantação, CodeDeploy exclui todos os arquivos que fazem parte da implantação anterior bem-sucedida. Isso inclui os arquivos que foram mantidos durante a implantação 1. No entanto, a implantação falha por motivos não relacionados.

Implantação	Revisão de aplicativo	Opção de substituição de conteúdo	Estado da implantação	Comportamento e resultado
implantação 3	revisão de aplicativo 1	MANTER		Como a reversão automática está habilitada para a implantação ou o grupo de implantação, o CodeDeploy implanta a revisão de aplicativo em bom estado conhecida, a revisão de aplicativo 1. No entanto, os arquivos que você queria manter na implantação 1 foram excluídos antes que a implantação 2 falhasse e não podem ser recuperados por AWS CodeDeploy. Você poderá adicioná-los à instância se eles forem necessário para a revisão de

Implantação	Revisão de aplicativo	Opção de substituição de conteúdo	Estado da implantação	Comportamento e resultado
				aplicativo 1, ou poderá criar uma nova revisão de aplicativo.

Implemente um aplicativo em outro AWS account

As organizações geralmente têm várias AWS contas que usam para finalidades diferentes (por exemplo, uma para tarefas de administração do sistema e outra para tarefas de desenvolvimento, teste e produção ou uma associada a ambientes de desenvolvimento e teste e outra associada ao ambiente de produção).

Embora você possa realizar trabalhos relacionados em contas diferentes, os grupos de CodeDeploy implantação e as instâncias do Amazon EC2 nas quais eles são implantados estão estritamente vinculados às contas sob as quais foram criados. Por exemplo, você não pode adicionar uma instância iniciada em uma conta em um grupo de implantação em outra instância.

Suponha que você tenha duas AWS contas: sua conta de desenvolvimento e sua conta de produção. Você trabalha principalmente na conta de desenvolvimento, mas deseja poder iniciar implantações na conta de produção sem um conjunto completo de credenciais ou sem ter que se desconectar da conta de desenvolvimento e entrar na conta de produção.

Depois de seguir as etapas de configuração entre contas, você pode iniciar implantações que pertençam a outra conta da sua organização sem precisar de um conjunto completo de credenciais para essa outra conta. Isso é feito em parte usando uma capacidade fornecida pelo AWS Security Token Service (AWS STS), que concede acesso temporário a essa conta.

Etapa 1: criar um bucket do S3 em qualquer uma das contas

Na conta de desenvolvimento ou de produção:

- Se ainda não tiver feito isso, crie um bucket do Amazon S3 no qual as revisões de aplicativo para a conta de produção serão armazenadas. Para obter informações, consulte Criar um bucket no Amazon S3. Você pode até mesmo usar o mesmo bucket e as mesmas revisões de aplicativo para

ambas as contas, implantando no seu ambiente de produção os mesmos arquivos que você testou e verificou na sua conta de desenvolvimento.

Etapa 2: conceder permissões de bucket do Amazon S3; ao perfil de instância do IAM da conta de produção

Se o bucket do Amazon S3 criado na etapa 1 estiver na sua conta de produção, essa etapa não será necessária. A função que você assumir mais tarde já terá acesso a esse bucket, pois também está na conta de produção.

Se tiver criado o bucket do Amazon S3 na conta de desenvolvimento, faça o seguinte:

- Na conta de produção, crie um perfil de instância do IAM. Para mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.

  > **Note**
  >
  > Anote o ARN desse perfil de instância do IAM. Você precisará adicioná-lo à política entre buckets que será criada em seguida.

- Na conta de desenvolvimento, dê acesso ao bucket do Amazon S3 criado na conta de desenvolvimento para o perfil de instância do IAM que você acabou de criar na sua conta de produção. Para obter informações, consulte Exemplo 2: proprietário do bucket concedendo permissões para bucket entre contas.

  Observe o seguinte enquanto conclui o processo de concessão de permissões para bucket entre contas:

  - No passo a passo de amostra, a Conta A representa sua conta de desenvolvimento, enquanto a Conta B representa sua conta de produção.
  - Quando você executar as tarefas da Conta A (conta de desenvolvimento), modifique a seguinte política de bucket para conceder permissões entre contas em vez de usar a política de amostra fornecida no passo a passo.

    JSON

    ```
    {
        "Version":"2012-10-17",
        "Statement": [
    ```

```
        {
            "Sid": "Cross-account permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/role-name"
            },
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

*account-id*representa o número da conta de produção na qual você acabou de criar o perfil da instância do IAM.

*role-name*representa o nome do perfil da instância do IAM que você acabou de criar.

*bucket-name*representa o nome do bucket que você criou na etapa 1. Certifique-se de incluir o `/*` depois do nome do seu bucket, para fornecer acesso a cada um dos arquivos dentro desse bucket.

Etapa 3: criar recursos e um perfil entre contas na conta de produção

Na sua conta de produção:

- Crie seus CodeDeploy recursos — aplicativo, grupo de implantação, configuração de implantação, instâncias do Amazon EC2, perfil da instância do Amazon EC2, função de serviço etc. — usando as instruções deste guia.
- Crie uma função adicional, uma função do IAM entre contas, que um usuário em sua conta de desenvolvimento possa assumir para realizar CodeDeploy operações nessa conta de produção.

  Use o passo a passo: delegue o acesso entre AWS contas usando as funções do IAM como um guia para ajudá-lo a criar a função entre contas. Em vez de adicionar os exemplos de permissões na explicação passo a passo ao seu documento de política, você deve anexar, no mínimo, as duas políticas AWS fornecidas a seguir à função:

- `AmazonS3FullAccess`: necessária apenas se o bucket do S3 estiver na conta de desenvolvimento. Fornece ao perfil da conta de produção assumida acesso total aos serviços e recursos do Amazon S3 na conta de desenvolvimento, na qual a revisão está armazenada.
- `AWSCodeDeployDeployerAccess`: permite que um usuário registre e implante revisões.

Se quiser criar e gerenciar grupos de implantação e não apenas iniciar implantações, adicione a política `AWSCodeDeployFullAccess` em vez da política `AWSCodeDeployDeployerAccess`. Para obter mais informações sobre o uso de políticas gerenciadas do IAM para conceder permissões para CodeDeploy tarefas, consulteAWS políticas gerenciadas (predefinidas) para CodeDeploy.

Você poderá anexar políticas adicionais se quiser realizar tarefas em outros serviços da AWS enquanto usa essa função entre contas.

Important

À medida que você criar o perfil do IAM entre contas, anote os detalhes que serão necessários para obter acesso à conta de produção.
Para usar o Console de gerenciamento da AWS para alternar funções, você precisará fornecer um dos seguintes:

- Uma URL para acessar a conta de produção com as credenciais da função assumida. Você encontrará o URL na página Review (Revisão), que é exibida no final do processo de criação de função de conta cruzada.
- O nome da função entre contas e o alias ou o número de identificação da conta.

Para usar o AWS CLI para alternar funções, você precisará fornecer o seguinte:

- O ARN da função entre contas que você assumirá.

Etapa 4: carregar a revisão de aplicativo no bucket do Amazon S3

Na conta em que você criou o bucket do Amazon S3:

- Carregue sua revisão de aplicativo no bucket do Amazon S3. Para mais informações, consulte Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações).

Etapa 5: assumir o perfil entre contas e implantar aplicativos

Na conta de desenvolvimento, você pode usar o AWS CLI ou o Console de gerenciamento da AWS para assumir a função entre contas e iniciar a implantação na conta de produção.

Para obter instruções sobre como usar o Console de gerenciamento da AWS para alternar funções e iniciar implantações, consulte Mudando para uma função (Console de gerenciamento da AWS) e. Crie uma implantação da EC2/On-Premises Compute Platform (console)

Para obter instruções sobre como usar o AWS CLI para assumir a função entre contas e iniciar implantações, consulte Como mudar para uma função do IAM () e.AWS Command Line Interface Crie uma implantação da EC2/On-Premises Compute Platform (CLI)

Para obter mais informações sobre como assumir uma função por meio de AWS STS, consulte AssumeRoleo Guia do AWS Security Token Service Usuário e assume-role na Referência de Comandos.AWS CLI

Tópico relacionado:

- CodeDeploy: Implantação de uma conta de desenvolvimento para uma conta de produção

Use o CodeDeploy agente para validar um pacote de implantação em uma máquina local

Usando o CodeDeploy agente, você pode implantar conteúdo em uma instância na qual está conectado. Isso permite testar a integridade de um arquivo de especificação de aplicativo (AppSpec arquivo) que você pretende usar em uma implantação e o conteúdo que você pretende implantar.

Você não precisa criar um grupo de aplicativo e implantação. Se você quiser implantar conteúdo armazenado na instância local, você nem precisa de uma AWS conta. Para o teste mais simples, você pode executar o codedeploy-agent deploy-local comando, sem especificar nenhuma opção, em um diretório que contém o AppSpec arquivo e o conteúdo a ser implantado. Existem opções para outros casos de teste na ferramenta.

Ao validar um pacote de implantação em uma máquina local, é possível:

- Testar a integridade de uma revisão de aplicativo.
- Teste o conteúdo de um AppSpec arquivo.
- Experimente CodeDeploy pela primeira vez com seu código de aplicativo existente.

- Implantar o conteúdo rapidamente quando você já estiver conectado a uma instância.

Você pode usar conteúdo de implantação armazenado na instância local ou em um tipo de repositório remoto compatível (buckets do Amazon S3 ou GitHub repositórios públicos).

Pré-requisitos

Antes de iniciar uma implantação local, siga estas etapas:

- Crie ou use um tipo de instância compatível com o CodeDeploy agente. Para mais informações, consulte Sistemas operacionais suportados pelo CodeDeploy agente.
- Instale a versão 2.0.0 ou posterior do CodeDeploy agente. O `deploy-local` subcomando requer a versão 2.0.0 ou posterior. Para a versão 1.8.x e anteriores, use o comando autônomo `codedeploy-local` em vez disso. Para obter informações sobre versões, consulte Histórico de versões do CodeDeploy agente. Para obter informações sobre a instalação, consulte Instale o CodeDeploy agente.
- Se você estiver implantando seu conteúdo a partir de um bucket GitHub ou repositório do Amazon S3, provisione um usuário para usá-lo. CodeDeploy Para mais informações, consulte Etapa 1: configuração.
- Se você estiver implantando a revisão do seu aplicativo a partir de um bucket do Amazon S3, crie um bucket do Amazon S3 na região em que você está trabalhando e aplique uma política de bucket do Amazon S3 ao bucket. Essa política concede às suas instâncias as permissões necessárias para baixar a revisão de aplicativo.

  Por exemplo, a seguinte política de bucket do Amazon S3 permite que qualquer instância do Amazon EC2 com um perfil de instância anexado do IAM contendo o ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` para fazer o download de qualquer lugar no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`.

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
```

```
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:role/CodeDeployDemo"
                ]
            }
        }
    ]
}
```

A seguinte política do bucket Amazon S3 permite que qualquer instância On-Premises com um usuário do IAM associado, contendo o ARN do `arn:aws:iam::444455556666:user/CodeDeployUser` para que seja feito o download de qualquer lugar no bucket do Amazon S3 chamado de `amzn-s3-demo-bucket`.

```
{
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:user/CodeDeployUser"
                ]
            }
        }
    ]
}
```

Para obter informações sobre como gerar e anexar uma política de bucket do Amazon S3, consulte Exemplos de políticas de bucket.

- Se você estiver implantando a revisão do seu aplicativo a partir de um bucket GitHub ou repositório do Amazon S3, configure um perfil de instância do IAM e anexe-o à instância. Para obter informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2, Crie uma instância do Amazon EC2 para (CodeDeploy AWS CLI ou console Amazon EC2) e Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo).

- Se você estiver implantando seu conteúdo a partir de GitHub, crie uma GitHub conta e um repositório público. Para criar uma GitHub conta, consulte Entrar GitHub. Para criar um GitHub repositório, consulte Criar um repositório.

  > **Note**
  >
  > Atualmente, não há suporte para repositórios privados. Se seu conteúdo estiver armazenado em um GitHub repositório privado, você poderá baixá-lo para a instância e usar a `--bundle-location` opção para especificar seu caminho local.

- Prepare o conteúdo (incluindo um AppSpec arquivo) que você deseja implantar na instância e coloque-o na instância local, no bucket do Amazon S3 ou no seu GitHub repositório. Para mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.
- Se você quiser usar valores diferentes dos padrões para outras opções de configuração, crie o arquivo de configuração e coloque-o na instância (`/etc/codedeploy-agent/conf/codedeployagent.yml` para instâncias do Amazon Linux, RHEL, ou Ubuntu Server ou `C:\ProgramData\Amazon\CodeDeploy\conf.yml` para instâncias do Windows Server). Para mais informações, consulte CodeDeploy referência de configuração do agente.

  > **Note**
  >
  > Ao usar um arquivo de configuração nas instâncias do Amazon Linux, RHEL ou Ubuntu Server, você precisa realizar uma das seguintes ações:
  >
  > - Usar as variáveis `:root_dir:` e `:log_dir:` para especificar locais diferentes dos padrões para as pastas do diretório raiz e de log da implantação.
  > - Use `sudo` para executar comandos CodeDeploy do agente.

Criar uma implantação local

Na instância em que você quer criar a implantação local, abra uma sessão de terminal (instâncias do Amazon Linux, RHEL ou Ubuntu Server) ou um prompt de comando (Windows Server) para executar os comandos da ferramenta.

> **Note**
>
> O codedeploy-agent deploy-local comando usa o `codedeploy-agent` binário nos seguintes locais:
>
> - No Amazon Linux, RHEL ou Ubuntu Server:`/opt/codedeploy-agent/bin/codedeploy-agent`.
> - No Windows Server: `C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe`.
>
> Para compatibilidade com versões anteriores, o codedeploy-local comando também está disponível nos mesmos locais:
>
> - No Amazon Linux, RHEL ou Ubuntu Server, o comando está localizado em`/opt/codedeploy-agent/bin/codedeploy-local`.
> - No Windows Server, o comando está localizado em`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-local.exe`.
>
> A codedeploy-local [options] sintaxe continua funcionando e aceita as mesmas opções quecodedeploy-agent deploy-local.

Sintaxe básica de comando

```
codedeploy-agent deploy-local [options]
```

Resumo

```
codedeploy-agent deploy-local
[--bundle-location <value>]
[--type <value>]
[--file-exists-behavior <value>]
[--deployment-group <value>]
[--deployment-group-name <value>]
[--application-name <value>]
[--events <comma-separated values>]
[--agent-configuration-file <value>]
[--appspec-filename <value>]
```

Opções

-l, --bundle-location

O local do pacote de revisão do aplicativo. Se você não especificar um local, por padrão, a ferramenta usará o diretório no qual você está trabalhando no momento. Se você especificar um valor para `--bundle-location`, também deverá especificar um valor para `--type`.

Exemplos de formato de local do pacote:

- Amazon Linux, RHEL ou Ubuntu Server local instância: `/path/to/local/bundle.tgz`
- Instância local do Windows Server: `C:/path/to/local/bundle`
- Bucket do Amazon S3: `s3://amzn-s3-demo-bucket/bundle.tar`
- GitHub repositório: `https://github.com/`*`account-name`*`/`*`repository-name`*`/`

-t, --type

O formato do pacote de revisão do aplicativo. Os tipos compatíveis incluem `tgz`, `tar`, `zip` e `directory`. Se você não especificar um tipo, a ferramenta usará `directory` por padrão. Se você especificar um valor para `--type`, também deverá especificar um valor para `--bundle-location`.

-b, --file-exists-behavior

Indica como são processados os arquivos que já existem em um local de destino de implantação, mas não faziam parte de uma implantação anterior bem-sucedida. As opções incluem DISALLOW, OVERWRITE e RETAIN. Para obter mais informações, consulte o arquivo ExistsBehavior na Referência AWS CodeDeploy da API.

-g, --deployment-group

O caminho para a pasta que é o local de destino do conteúdo a ser implantado. Se você não especificar uma pasta, a ferramenta criará uma com o nome default-local-deployment-group no diretório raiz da sua implantação. Para cada implantação local que você cria, a ferramenta cria um subdiretório dentro dessa pasta com nomes como local-1234 (onde o número é o ID do processo).

-d, --deployment-group-name

O nome do grupo de implantação exposto aos scripts lifecycle-hook como variável de ambiente. `DEPLOYMENT_GROUP_NAME` O padrão é `LocalFleet`.

-a, --nome-do-aplicativo

O nome do aplicativo exposto aos scripts lifecycle-hook como variável de ambiente. `APPLICATION_NAME` O padrão é a localização do pacote.

-e, --events

Um conjunto de ganchos de eventos de ciclo de vida de substituição que você deseja executar, em ordem, em vez dos eventos listados no arquivo. AppSpec É possível especificar vários ganchos, separando-os por vírgulas. Você pode usar essa opção se:

- Você deseja executar um conjunto diferente de eventos sem precisar atualizar o AppSpec arquivo.
- Você deseja executar um único gancho de evento como uma exceção ao que está no AppSpec arquivo, como`ApplicationStop`.

Se você não especificar DownloadBundlee instalar eventos na lista de substituição, eles serão executados antes de todos os ganchos de eventos que você especificar. Se você incluir DownloadBundlee instalar na lista de `--events` opções, elas deverão ser precedidas somente por eventos que normalmente são executados antes delas nas CodeDeploy implantações. Para mais informações, consulte AppSpec seção 'ganchos'.

-c, --agent-configuration-file

O local de um arquivo de configuração a ser usado para a implantação, se você o armazenar em um local diferente do padrão. Um arquivo de configuração especifica alternativas para outros valores e comportamentos padrão de uma implantação.

Por padrão, os arquivos de configuração são armazenados em `/etc/codedeploy-agent/conf/codedeployagent.yml` (instâncias do Amazon Linux, RHEL ou Ubuntu Server) ou `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server). Para obter mais informações, consulte CodeDeploy referência de configuração do agente.

-A, --appspec-nome do arquivo

O nome do AppSpec arquivo. Para implantações locais, os valores aceitos são `appspec.yml` e `appspec.yaml`. Por padrão, o AppSpec arquivo é chamado`appspec.yml`.

-h, --help

Exibe um resumo do conteúdo de ajuda.

-v, --version

Exibe o número de versão da ferramenta.

Exemplos

Veja a seguir exemplos de formatos de comando válidos.

```
codedeploy-agent deploy-local
```

```
codedeploy-agent deploy-local --bundle-location /path/to/local/bundle/directory
```

```
codedeploy-agent deploy-local --bundle-location C:/path/to/local/bundle.zip --type zip
 --deployment-group my-deployment-group
```

```
codedeploy-agent deploy-local --bundle-location /path/to/local/directory --type
 directory --deployment-group my-deployment-group
```

Implantar um empacotamento a partir do Amazon S3:

```
codedeploy-agent deploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.tgz --
type tgz
```

```
codedeploy-agent deploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.zip?
versionId=1234&etag=47e8 --type zip --deployment-group my-deployment-group
```

Implante um pacote a partir de um GitHub repositório público:

```
codedeploy-agent deploy-local --bundle-location https://github.com/awslabs/aws-
codedeploy-sample-tomcat --type zip
```

```
codedeploy-agent deploy-local --bundle-location https://api.github.com/repos/awslabs/
aws-codedeploy-sample-tomcat/zipball/master --type zip
```

```
codedeploy-agent deploy-local --bundle-location https://api.github.com/repos/awslabs/
aws-codedeploy-sample-tomcat/zipball/HEAD --type zip
```

```
codedeploy-agent deploy-local --bundle-location https://api.github.com/repos/awslabs/
aws-codedeploy-sample-tomcat/zipball/1a2b3c4d --type zip
```

Implantar um pacote especificando vários eventos de ciclo de vida:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/
bundle.tar --type tar --deployment-group my-deployment-group --events
 DownloadBundle,Install,ApplicationStart,HealthCheck
```

Interrompa um aplicativo implantado anteriormente usando o evento de ApplicationStop ciclo de vida:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --
deployment-group my-deployment-group --events ApplicationStop
```

Implantar usando um ID de grupo específico para implantação:

```
codedeploy-agent deploy-local --bundle-location C:/path/to/local/bundle/directory --
deployment-group 1234abcd-5dd1-4774-89c6-30b107ac5dca
```

```
codedeploy-agent deploy-local --bundle-location C:/path/to/local/bundle.zip --type zip
 --deployment-group 1234abcd-5dd1-4774-89c6-30b107ac5dca
```

Monitorando implantações em CodeDeploy

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de CodeDeploy suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra. Antes de começar a monitorar CodeDeploy, no entanto, você deve criar um plano de monitoramento que inclua respostas às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

A próxima etapa é estabelecer uma linha de base para o CodeDeploy desempenho normal em seu ambiente, medindo o desempenho em vários momentos e sob diferentes condições de carga. Ao monitorar CodeDeploy, armazene dados históricos de monitoramento para que você possa compará-los com os dados de desempenho atuais, identificar padrões normais de desempenho e anomalias de desempenho e criar métodos para resolver problemas.

Por exemplo, se você estiver usando CodeDeploy, poderá monitorar o status das implantações e das instâncias de destino. Quando as implantações ou instâncias falham, talvez seja necessário reconfigurar um arquivo de especificação do aplicativo, reinstalar ou atualizar o CodeDeploy agente, atualizar as configurações em um aplicativo ou grupo de implantação ou fazer alterações nas configurações da instância ou em um arquivo. AppSpec

Para estabelecer uma linha de base, você deve, no mínimo, monitorar os seguintes itens:

- Eventos e status de implantações
- Eventos e status de instâncias

Ferramentas de monitoramento automatizadas

AWS fornece várias ferramentas que você pode usar para monitorar CodeDeploy. É possível configurar algumas dessas ferramentas para fazer o monitoramento em seu lugar, e, ao mesmo

tempo, algumas das ferramentas exigem intervenção manual. Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

Você pode usar as seguintes ferramentas de monitoramento automatizado para observar CodeDeploy e relatar quando algo está errado:

- Amazon CloudWatch Alarms — Observe uma única métrica durante um período de tempo especificado por você e execute uma ou mais ações com base no valor da métrica em relação a um determinado limite em vários períodos. A ação é uma notificação enviada para um tópico do Amazon Simple Notification Service (Amazon SNS) ou para uma política do Amazon EC2 Auto Scaling. CloudWatch os alarmes não invocam ações simplesmente porque estão em um determinado estado; o estado deve ter sido alterado e mantido por um determinado número de períodos. Para obter mais informações, consulte Monitoring Deployments with Amazon CloudWatch Tools.

  Para obter informações sobre como atualizar sua função de serviço para funcionar com o monitoramento de alarmes do CloudWatch , consulte Conceder CloudWatch permissões para uma função CodeDeploy de serviço. Para obter informações sobre como adicionar monitoramento de CloudWatch alarmes às suas CodeDeploy operaçõesCrie um aplicativo com CodeDeploy, consulteCrie um grupo de implantação com CodeDeploy, ouAltere as configurações do grupo de implantação com CodeDeploy.

- Amazon CloudWatch Logs — Monitore, armazene e acesse seus arquivos de log de AWS CloudTrail ou de outras fontes. Para obter mais informações, consulte Monitoramento de arquivos de log no Guia CloudWatch do usuário da Amazon.

  Para obter informações sobre como usar o CloudWatch console para visualizar CodeDeploy registros, consulte Exibir CodeDeploy registros no console de CloudWatch registros.

- Amazon CloudWatch Events — Combine eventos e encaminhe-os para uma ou mais funções ou streams de destino para fazer alterações, capturar informações de estado e tomar medidas corretivas. Para obter mais informações, consulte O que é Amazon CloudWatch Events no Guia CloudWatch do usuário da Amazon.

  Para obter informações sobre o uso de CloudWatch Eventos em suas CodeDeploy operações, consulteMonitoramento de implantações com Amazon Events CloudWatch.

- AWS CloudTrail Monitoramento de registros — compartilhe arquivos de log entre contas, monitore arquivos de CloudTrail log em tempo real enviando-os para o CloudWatch Logs, grave aplicativos de processamento de log em Java e valide se seus arquivos de log não foram alterados após

a entrega. CloudTrail Para obter mais informações, consulte Trabalhando com arquivos de CloudTrail log no Guia AWS CloudTrail do usuário.

Para obter informações sobre como usar CloudTrail com CodeDeploy, consulteMonitoring Deployments.

- Amazon Simple Notification Service: configure gatilhos com base em eventos para receber notificações por SMS ou e-mail sobre eventos de implantação e de instâncias, como sucesso ou falha. Para obter mais informações, consulte Criar um tópico e O que é o Amazon Simple Notification Service.

  Para obter informações sobre como configurar as notificações do Amazon SNS para CodeDeploy, consulte. Monitoring Deployments with Amazon SNS Event Notifications

Ferramentas de monitoramento manual

Outra parte importante do monitoramento CodeDeploy envolve o monitoramento manual dos itens que os CloudWatch alarmes não cobrem. O CodeDeploy, CloudWatch, e outros painéis AWS do console fornecem uma visão rápida do estado do seu ambiente. AWS Recomendamos que você também verifique os arquivos de log nas CodeDeploy implantações.

- CodeDeploy o console mostra:
  - O status das implantações
  - A data e a hora de cada última tentativa de implantação e de cada última implementação bem-sucedida de uma revisão
  - O número de instâncias que tiveram êxito, falharam, foram ignoradas ou estão em andamento em uma implantação
  - O status das instâncias locais
  - A data e a hora em que as instâncias locais foram registradas ou tiveram seu registro cancelado
- CloudWatch a página inicial mostra:
  - Alertas e status atual
  - Gráficos de alertas e recursos
  - Estado de integridade do serviço

  Além disso, você pode usar CloudWatch para fazer o seguinte:
  - Crie painéis personalizados para monitorar os serviços com os quais você se preocupa.

- Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências
- Pesquise e navegue em todas as suas métricas AWS de recursos
- Criar e editar alertas para ser notificado sobre problemas

Tópicos

- Monitoring Deployments with Amazon CloudWatch Tools
- Monitoring Deployments
- Monitoring Deployments with Amazon SNS Event Notifications

Monitoramento de implantações com ferramentas da Amazon CloudWatch

Você pode monitorar CodeDeploy implantações usando as seguintes CloudWatch ferramentas: Amazon CloudWatch Events, CloudWatch alarmes e Amazon CloudWatch Logs.

A análise dos registros criados pelo CodeDeploy agente e pelas implantações pode ajudá-lo a solucionar as causas das falhas de implantação. Como alternativa à análise de CodeDeploy registros em uma instância por vez, você pode usar o CloudWatch Logs para monitorar todos os registros em um local central.

Para obter informações sobre o uso de CloudWatch alarmes e CloudWatch eventos para monitorar suas CodeDeploy implantações, consulte os tópicos a seguir.

Tópicos

- Monitorando implantações com CloudWatch alarmes em CodeDeploy
- Monitoramento de implantações com Amazon Events CloudWatch

Monitorando implantações com CloudWatch alarmes em CodeDeploy

Você pode criar um CloudWatch alarme para uma instância ou grupo do Amazon EC2 Auto Scaling que você está usando CodeDeploy em suas operações. Um alarme monitora uma única métrica durante um período que você especifica e executa uma ou mais ações com base no valor da métrica em relação a um determinado limite em vários períodos. CloudWatch os alarmes invocam ações quando seu estado muda (por exemplo, de `OK` para`ALARM`).

Usando a funcionalidade de CloudWatch alarme nativa, você pode especificar qualquer uma das ações suportadas CloudWatch quando uma instância que você está usando em uma implantação falha, como enviar uma notificação do Amazon SNS ou interromper, encerrar, reinicializar ou recuperar uma instância. Para suas CodeDeploy operações, você pode configurar um grupo de implantação para interromper uma implantação sempre que qualquer CloudWatch alarme associado ao grupo de implantação for ativado.

Você pode associar até dez CloudWatch alarmes a um grupo de CodeDeploy implantação. Se qualquer um dos alarmes especificados for ativado, a implantação será interrompida, e o status será atualizado para Stopped. Para usar essa opção, você deve conceder CloudWatch permissões à sua função CodeDeploy de serviço.

Para obter informações sobre a configuração de CloudWatch alarmes no CloudWatch console, consulte Criação de CloudWatch alarmes da Amazon no Guia CloudWatch do usuário da Amazon.

Para obter informações sobre como associar um CloudWatch alarme a um grupo de implantação em CodeDeploy, consulte Crie um grupo de implantação com CodeDeploy e. Altere as configurações do grupo de implantação com CodeDeploy

Tópicos

- Conceder CloudWatch permissões para uma função CodeDeploy de serviço

Conceder CloudWatch permissões para uma função CodeDeploy de serviço

Antes de usar o monitoramento de CloudWatch alarmes com suas implantações, a função de serviço que você usa em suas CodeDeploy operações deve ter permissão para acessar os CloudWatch recursos.

Para conceder CloudWatch permissões a uma função de serviço

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No console do IAM, selecione Perfis no painel de navegação.
3. Escolha o nome da função de serviço que você usa em suas AWS CodeDeploy operações.
4. Na guia Permissões, na área de Políticas em linha, selecione Criar política da função.

   –ou–

Se o botão Criar política da função não estiver disponível, expanda a área de Políticas em linha e depois clique aqui.

5. Na página Definir permissões, selecione Política personalizada e, então, Selecionar.
6. Na página Revisar política, no campo Nome da política, digite um nome para identificar essa política, tal como `CWAlarms`.
7. Cole o seguinte no campo Documento da política:

   JSON

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

8. Selecione Aplicar política.

Monitoramento de implantações com Amazon Events CloudWatch

Você pode usar o Amazon CloudWatch Events para detectar e reagir às mudanças no estado de uma instância ou implantação (um “evento”) em suas CodeDeploy operações. Em seguida, com base nas regras criadas por você, CloudWatch os Eventos invocarão uma ou mais ações de destino quando uma implantação ou instância entrar no estado especificado em uma regra. Dependendo do tipo de alteração de estado, convém enviar notificações, capturar informações de estado, tomar medidas corretivas, iniciar eventos ou realizar outras ações. Você pode selecionar os seguintes tipos de alvos ao usar CloudWatch Eventos como parte de suas CodeDeploy operações:

- AWS Lambda funções
- Streams do Kinesis
- Filas do Amazon SQS

- Built-in alvos (`EC2 CreateSnapshot API call`EC2 `RebootInstances API call`,EC2 `StopInstances API call`, eEC2 `TerminateInstances API call`)
- Tópicos do Amazon SNS

A seguir, alguns casos de uso:

- Use uma função do Lambda para transmitir uma notificação a um canal Slack sempre que uma implantação falhar.
- Envie dados sobre implantações ou instâncias a um stream do Kinesis para oferecer suporte ao monitoramento abrangente do status em tempo real.
- Use ações de CloudWatch alarme para interromper, encerrar, reinicializar ou recuperar automaticamente instâncias do Amazon EC2 quando ocorrer uma implantação ou evento de instância especificado por você.

O restante deste tópico descreve o procedimento básico para criar uma regra de CloudWatch eventos para CodeDeploy. No entanto, antes de criar regras de eventos para uso em suas CodeDeploy operações, você deve fazer o seguinte:

- Conclua os pré-requisitos CloudWatch dos eventos. Para obter informações, consulte Pré-requisitos de CloudWatch eventos da Amazon.
- Familiarize-se com eventos, regras e metas em CloudWatch Eventos. Para obter mais informações, consulte O que é Amazon CloudWatch Events? e novos CloudWatch eventos — acompanhe e responda às mudanças em seus AWS recursos.
- Criar um ou mais destinos que você usará nas suas regras de evento.

Para criar uma regra de CloudWatch eventos para CodeDeploy:

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Eventos.
3. Escolha Create rule (Criar regra) e, em Event selector (Seletor de eventos), escolha AWS CodeDeploy.
4. Especifique um tipo de detalhe:
   - Para criar uma regra que se aplique a todas as alterações de estado das instâncias e das implantações, escolha Any detail type e vá para a etapa 6.

- Para criar uma regra que se aplique somente às instâncias, escolha Tipo de detalhe específico e, em seguida, escolha State-changeNotificação de CodeDeploy instância.
- Para criar uma regra que se aplique somente às implantações, escolha Tipo de detalhe específico e escolha State-changeNotificação de CodeDeploy implantação.

5. Especifique as alterações de estado às quais a regra se aplica:

   - Para criar uma regra que se aplique a todas as alterações de estado, escolha Any state.
   - Para criar uma regra que se aplique apenas a algumas alterações de estado, escolha Specific state(s) e escolha um ou mais valores de status na lista. A tabela a seguir lista os valores de status que você pode escolher:

Valores de status de implantação	Valores de status de instâncias
FALHA	FALHA
START	START
STOP	PRONTO
QUEUED	BEM-SUCEDIDA
PRONTO	
BEM-SUCEDIDA	

6. Especifique a quais CodeDeploy aplicativos a regra se aplica:

   - Para criar uma regra que se aplique a todos os aplicativos, escolha Any application e vá para a etapa 8.
   - Para criar uma regra que se aplique a apenas um aplicativo, escolha Specific application e escolha o nome do aplicativo na lista.

7. Especifique a quais grupos de implantação a regra se aplica:

   - Para criar uma regra que se aplique a todos os grupos de implantação associados ao aplicativo selecionado, escolha Any deployment group.
   - Para criar uma regra que se aplique a apenas um dos grupos de implantação associados ao aplicativo selecionado, escolha Specific deployment group(s) e escolha o nome do grupo de implantação na lista.

8. Analise a configuração da regra para garantir que ela atenda aos requisitos de monitoramento de evento.
9. Na área Targets, escolha Add target*.
10. Na lista Select target type (Selecionar tipo de destino), escolha o tipo de destino preparado para usar com essa regra e configure as opções adicionais necessárias a esse tipo.
11. Escolha Configure details (Configurar detalhes).
12. Na página Configurar detalhes da regra, digite um nome e uma descrição para a regra e escolha a caixa Estado para habilitar a regra agora.
13. Se você estiver satisfeito com a regra, escolha Create rule.

Monitorando implantações com AWS CloudTrail

CodeDeploy é integrado com CloudTrail, um serviço que captura chamadas de API feitas por ou em nome de CodeDeploy sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrailcaptura chamadas de API do CodeDeploy console, de CodeDeploy comandos por meio do ou AWS CLI diretamente das CodeDeploy APIs. Usando as informações coletadas por CloudTrail, você pode determinar para qual solicitação foi feita CodeDeploy, o endereço IP de origem a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e assim por diante. Para saber mais sobre CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

CodeDeploy informações em CloudTrail

Quando o CloudTrail registro está ativado em sua AWS conta, as chamadas de API feitas para CodeDeploy ações são rastreadas em arquivos de log. CodeDeploy os registros são gravados junto com outros registros AWS de serviço em um arquivo de log. CloudTrail determina quando criar e gravar em um novo arquivo com base no período e no tamanho do arquivo.

Todas as CodeDeploy ações são registradas e documentadas na Referência da Linha de AWS CodeDeploy Comando e na Referência da AWS CodeDeploy API. Por exemplo, chamadas para criar implantações, excluir aplicativos e registrar revisões de aplicativos geram entradas em arquivos de CloudTrail log.

Cada entrada de log contém informações sobre quem gerou a solicitação. As informações de identidade do usuário no registro ajudam a determinar se a solicitação foi feita com credenciais raiz ou de usuário, com credenciais de segurança temporárias para uma função ou usuário federado ou

por outro serviço. AWS Para obter mais informações, consulte o campo userIdentity na Referência de eventos do CloudTrail .

Você pode armazenar os arquivos de log no bucket pelo tempo que desejar, mas também pode definir regras do ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Por padrão, a criptografia do lado do servidor (SSE) do Amazon S3 é usada para criptografar seus arquivos de log.

Você pode CloudTrail publicar notificações do Amazon SNS quando novos arquivos de log forem entregues. Para obter mais informações, consulte Configuração de notificações do Amazon SNS para o CloudTrail.

Você também pode agregar arquivos de CodeDeploy log de várias AWS regiões e várias AWS contas em um único bucket do Amazon S3. Para obter mais informações, consulte Recebimento de arquivos de CloudTrail log de várias regiões.

Entendendo as entradas do arquivo de CodeDeploy log

CloudTrail os arquivos de log podem conter uma ou mais entradas de log, em que cada entrada é composta por vários JSON-formatted eventos. Uma entrada de log representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, quaisquer parâmetros, a data e hora da ação e assim por diante. Não é garantido que as entradas de log estejam em uma ordem específica. Ou seja, elas não são um rastreamento ordenado das chamadas de API públicas.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a ação de CodeDeploy criar grupo de implantação:

```
{
 "Records": [{
  "eventVersion": "1.02",
  "userIdentity": {
   "type": "AssumedRole",
   "principalId": "AKIAI44QH8DHBEXAMPLE:203.0.113.11",
   "arn": "arn:aws:sts::123456789012:assumed-role/example-role/203.0.113.11",
   "accountId": "123456789012",
   "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
   "sessionContext": {
    "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2014-11-27T03:57:36Z"
    },
    "sessionIssuer": {
```

```
     "type": "Role",
     "principalId": "AKIAI44QH8DHBEXAMPLE",
     "arn": "arn:aws:iam::123456789012:role/example-role",
     "accountId": "123456789012",
     "userName": "example-role"
    }
   }
  },
  "eventTime": "2014-11-27T03:57:36Z",
  "eventSource": "codedeploy.amazonaws.com",
  "eventName": "CreateDeploymentGroup",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.11",
  "userAgent": "example-user-agent-string",
  "requestParameters": {
   "applicationName": "ExampleApplication",
   "serviceRoleArn": "arn:aws:iam::123456789012:role/example-instance-group-role",
   "deploymentGroupName": "ExampleDeploymentGroup",
   "ec2TagFilters": [{
               "value": "CodeDeployDemo",
    "type": "KEY_AND_VALUE",
    "key": "Name"
           }],
           "deploymentConfigName": "CodeDeployDefault.HalfAtATime"
  },
  "responseElements": {
   "deploymentGroupId": "7d64e680-e6f4-4c07-b10a-9e117EXAMPLE"
  },
  "requestID": "86168559-75e9-11e4-8cf8-75d18EXAMPLE",
  "eventID": "832b82d5-d474-44e8-a51d-093ccEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
 },
    ... additional entries ...
    ]
}
```

Monitoramento de implantações com notificações de eventos do Amazon SNS

Você pode adicionar gatilhos a um grupo de CodeDeploy implantação para receber notificações sobre eventos relacionados a implantações ou instâncias nesse grupo de implantação. Essas

notificações são enviadas aos destinatários inscritos em um tópico do Amazon SNS que você incluiu como parte da ação do gatilho.

Você pode receber notificações de CodeDeploy eventos em mensagens SMS ou mensagens de e-mail. Também pode usar de outras maneiras os dados JSON que são criados quando um evento especificado ocorre, por exemplo, enviando mensagens a filas do Amazon SQS ou invocando uma função no AWS Lambda. Para ver a estrutura dos dados JSON fornecidos para gatilhos de implantação e instância, consulte Formatos de dados JSON para acionadores CodeDeploy.

Você poderá optar por usar gatilhos para receber notificações se:

- For um desenvolvedor que precisa saber quando uma implantação apresenta falhas ou é interrompida, para que você possa solucionar o problema.
- For um administrador do sistema que precisa saber quantas instâncias apresentam falhas para monitorar a integridade do seu Amazon EC2 Fleet.
- For um gerente que deseja uma contagem instantânea de eventos de implantação e instância, que possa ser obtida por meio de regras de filtragem que direcionam diferentes tipos de notificações para pastas no seu cliente de e-mail de desktop.

Você pode criar até 10 acionadores para cada grupo de CodeDeploy implantação, para qualquer um dos seguintes tipos de eventos.

Eventos de implantação	Eventos de instância
• Bem-sucedida • Falha • Iniciou • Interrompido • Reversão • Ready[1] • Todos os eventos de implantação	• Bem-sucedida • Falha • Iniciou • Ready[1] • Todos os eventos de instância

[1] Aplica-se somente a blue/green implantações. Indica que a última revisão do aplicativo foi instalada em instâncias em um ambiente de substituição e que o tráfego do ambiente original agora pode ser redirecionado atrás de um balanceador de carga. Para obter mais informações, consulte Trabalhando com implantações em CodeDeploy.

Tópicos

- Conceda permissões do Amazon SNS para uma função de serviço CodeDeploy
- Crie um gatilho para um CodeDeploy evento
- Editar um gatilho em um grupo CodeDeploy de implantação
- Excluir um gatilho de um grupo CodeDeploy de implantação
- Formatos de dados JSON para acionadores CodeDeploy

Conceda permissões do Amazon SNS para uma função de serviço CodeDeploy

Antes que seus gatilhos possam gerar notificações, a função de serviço que você usa em suas CodeDeploy operações deve ter permissão para acessar os recursos do Amazon SNS.

Para conceder permissões do Amazon SNS a um perfil de serviço

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No console do IAM, selecione Perfis no painel de navegação.
3. Escolha o nome do perfil de serviço utilizado em suas operações do AWS CodeDeploy .
4. Na guia Permissões, na área de Políticas em linha, selecione Criar política da função.

   –ou–

   Se o botão Criar política da função não estiver disponível, expanda a área de Políticas em linha e depois clique aqui.
5. Na página Definir permissões, selecione Política personalizada e, então, Selecionar.
6. Na página Review Policy (Revisar política), no campo Policy Name (Nome da política), insira um nome para identificar essa política, tal como `SNSPublish`.
7. Cole o seguinte no campo Documento da política:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "*"
        }
    ]
}
```

8. Selecione Aplicar política.

Crie um gatilho para um CodeDeploy evento

Você pode criar um gatilho que publique um tópico do Amazon Simple Notification Service (Amazon SNS) para um AWS CodeDeploy evento de implantação ou instância. Dessa forma, quando esse evento ocorrer, todos os assinantes do tópico associado receberão notificações através do endpoint especificado no tópico, como uma mensagem SMS ou uma mensagem de e-mail. O Amazon SNS oferece vários métodos para assinar tópicos.

Antes de criar um gatilho, é necessário configurar o tópico do Amazon SNS para o gatilho apontar. Para obter informações, consulte Criar um tópico. Ao criar um tópico, é recomendável dar a ele um nome que identifique sua finalidade, em formatos como `Topic-group-us-west-3-deploy-fail` ou `Topic-group-project-2-instance-stop`.

Você também deve conceder permissões ao Amazon SNS para uma função de CodeDeploy serviço antes que as notificações possam ser enviadas para seu gatilho. Para mais informações, consulte Conceda permissões do Amazon SNS para uma função de serviço CodeDeploy.

Depois de criar o tópico, você poderá adicionar assinantes. Para obter informações sobre como criar, gerenciar e assinar tópicos, consulte O que é o Amazon Simple Notification Service.

Crie um gatilho para enviar notificações de CodeDeploy eventos (console)

Você pode usar o CodeDeploy console para criar acionadores para um CodeDeploy evento. No final do processo de configuração, uma mensagem de notificação de teste é enviada para garantir que as permissões e os detalhes do gatilho estejam configurados corretamente.

Para criar um gatilho para um CodeDeploy evento

1. No Console de gerenciamento da AWS, abra o AWS CodeDeploy console.

2. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

3. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
4. Na página Applications (Aplicativos), escolha o nome do aplicativo associado ao grupo de implantação no qual você deseja adicionar um gatilho.
5. Na página Application details (Detalhes do aplicativo), escolha o grupo de implantação no qual você deseja adicionar um gatilho.
6. Escolha Editar.
7. Expanda Avançado - opcional.
8. Na área Triggers (Gatilhos), selecione Create trigger (Criar gatilho).
9. No painel Create deployment trigger (Criar gatilho de implantação), faça o seguinte:

   a. Em Trigger name (Nome de acionador), insira um nome para o acionador que facilite a identificação de sua finalidade. Recomendamos formatos como `Trigger-group-us-west-3-deploy-fail` ou `Trigger-group-eu-central-instance-stop`.

   b. Em Eventos, escolha um ou mais tipos de eventos que acionarão o envio de notificações do tópico do Amazon SNS.

   c. Em Tópicos do Amazon SNS, escolha o nome do tópico que você criou para enviar notificações para esse gatilho.

   d. Escolha Criar gatilho. CodeDeploy envia uma notificação de teste para confirmar que você configurou corretamente o acesso entre CodeDeploy e o tópico do Amazon SNS. Dependendo do tipo de endpoint selecionado para o tópico, e se você tiver assinado o tópico, será enviada uma confirmação em uma mensagem SMS ou de e-mail.

10. Escolha Salvar alterações.

Crie um gatilho para enviar notificações de CodeDeploy eventos (CLI)

Você pode usar a CLI para incluir gatilhos ao criar um grupo de implantação ou pode adicionar gatilhos a um grupo de implantação existente.

Para criar um gatilho para enviar notificações referentes um novo grupo de implantação

Crie um arquivo JSON para configurar o grupo de implantação e, em seguida, execute o comando create-deployment-group usando a opção `--cli-input-json`.

A maneira mais simples de criar o arquivo JSON é usar a opção `--generate-cli-skeleton` para obter uma cópia do formato JSON e, em seguida, fornecer os valores necessários em um editor de texto simples.

1. Execute o seguinte comando e depois copie os resultados em um editor de texto simples.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Adicione o nome de um CodeDeploy aplicativo existente à saída:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentGroupName": "",
    "deploymentConfigName": "",
    "ec2TagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "onPremisesInstanceTagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "autoScalingGroups": [
        ""
    ],
    "serviceRoleArn": "",
    "triggerConfigurations": [
        {
            "triggerName": "",
            "triggerTargetArn": "",
            "triggerEvents": [
                ""
```

```
            ]
        }
    ]
}
```

3. Forneça valores para os parâmetros que você deseja configurar.

   Ao usar o comando create-deployment-group, você deve fornecer pelo menos valores para os seguintes parâmetros:

   - `applicationName`: o nome de um aplicativo já criado na sua conta.
   - `deploymentGroupName`: um nome para o grupo de implantação que você está criando.
   - `serviceRoleArn`: o ARN de uma função de serviço existente configurada CodeDeploy em sua conta. Para mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

   Na seção `triggerConfigurations`, forneça valores para os seguintes parâmetros:

   - `triggerName`: o nome que você deseja dar ao gatilho, para poder identificá-lo facilmente. Recomendamos formatos como `Trigger-group-us-west-3-deploy-fail` ou `Trigger-group-eu-central-instance-stop`.
   - `triggerTargetArn`: o ARN do tópico do Amazon SNS que você criou para associar ao seu gatilho, neste formato: `arn:aws:sns:us-east-2:444455556666:NewTestTopic`.
   - `triggerEvents`: o tipo de eventos para os quais você deseja disparar notificações. É possível especificar um ou mais tipos de eventos, separando vários dos seus nomes com vírgulas (por exemplo, `"triggerEvents": ["DeploymentSuccess","DeploymentFailure","InstanceFailure"]`). Quando mais de um tipo de evento é adicionado, as notificações para todos esses tipos são enviadas ao tópico que você especificou, em vez de a um tópico diferente para cada um. É possível escolher entre os seguintes tipos de eventos:
     - DeploymentStart
     - DeploymentSuccess
     - DeploymentFailure
     - DeploymentStop
     - DeploymentRollback

- DeploymentReady (Aplica-se somente às instâncias de substituição em uma blue/green implantação)
- InstanceStart
- InstanceSuccess
- InstanceFailure
- InstanceReady (Aplica-se somente às instâncias de substituição em uma blue/green implantação)

O exemplo de configuração a seguir cria um grupo de implantação chamado `dep-group-ghi-789-2` para um aplicativo chamado `TestApp-us-east-2` e um gatilho que solicitará o envio de notificações sempre que uma implantação for iniciada, for bem-sucedida ou falhar:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "deploymentGroupName": "dep-group-ghi-789-2",
    "ec2TagFilters": [
        {
            "Key": "Name",
            "Value": "Project-ABC",
            "Type": "KEY_AND_VALUE"
        }
    ],
    "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
    "triggerConfigurations": [
        {
            "triggerName": "Trigger-group-us-east-2",
            "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-
deployments",
            "triggerEvents": [
                "DeploymentStart",
                "DeploymentSuccess",
                "DeploymentFailure"
            ]
        }
    ]
}
```

4. Salve suas atualizações como um arquivo JSON e, em seguida, chame esse arquivo usando a opção `--cli-input-json` ao executar o comando create-deployment-group:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

No final do processo de criação, você receberá uma mensagem de notificação de teste indicando que tanto as permissões quanto os detalhes do gatilho estão configurados corretamente.

Para criar um gatilho para enviar notificações referentes um grupo de implantação existente

Para usar o AWS CLI para adicionar gatilhos para CodeDeploy eventos a um grupo de implantação existente, crie um arquivo JSON para atualizar o grupo de implantação e, em seguida, execute o comando update-deployment-group usando a opção. `--cli-input-json`

A maneira mais simples de criar o arquivo JSON é executar o comando get-deployment-group para obter uma cópia da configuração do grupo de implantação, no formato JSON, e depois atualizar os valores dos parâmetros em um editor de texto simples.

1. Execute o seguinte comando e depois copie os resultados em um editor de texto simples.

```
aws deploy get-deployment-group --application-name application --deployment-group-
name deployment-group
```

2. Exclua o seguinte da saída:

   - No início da saída, exclua `{ "deploymentGroupInfo":`.
   - No final da saída, exclua `}`.
   - Exclua a linha que contém `deploymentGroupId`.
   - Exclua a linha que contém `deploymentGroupName`.

   O conteúdo do seu arquivo de texto agora deve ser semelhante ao seguinte:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "autoScalingGroups": [],
    "ec2TagFilters": [
        {
            "Type": "KEY_AND_VALUE",
            "Value": "Project-ABC",
            "Key": "Name"
        }
    ],
    "triggerConfigurations": [],
    "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
    "onPremisesInstanceTagFilters": []
}
```

3. Na seção `triggerConfigurations`, adicione dados para os parâmetros `triggerEvents`, `triggerTargetArn` e `triggerName`. Para obter informações sobre os parâmetros de configuração do acionador, consulte TriggerConfig.

   O conteúdo do seu arquivo de texto agora deve ser semelhante ao seguinte. Esse código solicitará que as notificações sejam enviadas sempre que uma implantação for iniciada, for bem-sucedida ou falhar.

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "autoScalingGroups": [],
    "ec2TagFilters": [
        {
            "Type": "KEY_AND_VALUE",
            "Value": "Project-ABC",
            "Key": "Name"
        }
    ],
    "triggerConfigurations": [
        {
            "triggerEvents": [
                "DeploymentStart",
                "DeploymentSuccess",
                "DeploymentFailure"
            ],
```

```
            "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-
deployments",
            "triggerName": "Trigger-group-us-east-2"
        }
    ],
    "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
    "onPremisesInstanceTagFilters": []
}
```

4. Salve suas atualizações como um arquivo JSON e execute o comando update-deployment-group usando a opção `--cli-input-json`. Certifique-se de incluir a `--current-deployment-group-name` opção e substituir o nome do seu arquivo JSON por*filename*:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-
group-name --cli-input-json file://filename.json
```

No final do processo de criação, você receberá uma mensagem de notificação de teste indicando que tanto as permissões quanto os detalhes do gatilho estão configurados corretamente.

Editar um gatilho em um grupo CodeDeploy de implantação

Se os requisitos de notificação forem alterados, você poderá modificar seu gatilho em vez de criar um novo.

Modificar um CodeDeploy gatilho (CLI)

Para usar os detalhes do acionador AWS CLI para alterar os CodeDeploy eventos ao atualizar um grupo de implantação, crie um arquivo JSON para definir as alterações nas propriedades do grupo de implantação e, em seguida, execute o comando update-deployment-group com a opção. `--cli-input-json`

A maneira mais simples de criar o arquivo JSON é executar o comando get-deployment-group para obter os detalhes do grupo de implantação atual no formato JSON e depois editar os valores necessários em um editor de texto simples.

1. Execute o comando a seguir, substituindo os nomes do seu aplicativo e do grupo de implantação por *application* e: *deployment-group*

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```

2. Copie os resultados do comando em um editor de texto simples e exclua o seguinte:

   - No início da saída, exclua `{ "deploymentGroupInfo":`.
   - No final da saída, exclua `}`.
   - Exclua a linha que contém `deploymentGroupId`.
   - Exclua a linha que contém `deploymentGroupName`.

   O conteúdo do seu arquivo de texto agora deve ser semelhante ao seguinte:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "autoScalingGroups": [],
    "ec2TagFilters": [
        {
            "Type": "KEY_AND_VALUE",
            "Value": "East-1-Instances",
            "Key": "Name"
        }
    ],
    "triggerConfigurations": [
        {
            "triggerEvents": [
                "DeploymentStart",
                "DeploymentSuccess",
                "DeploymentFailure",
                "DeploymentStop"
            ],
            "triggerTargetArn": "arn:aws:sns:us-east-2:111222333444:Trigger-group-us-east-2",
```

```
            "triggerName": "Trigger-group-us-east-2"
        }
    ],
    "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
    "onPremisesInstanceTagFilters": []
}
```

3. Altere quaisquer parâmetros, conforme necessário. Para obter informações sobre os parâmetros de configuração do acionador, consulte TriggerConfig.
4. Salve suas atualizações como um arquivo JSON e execute o comando update-deployment-group usando a opção `--cli-input-json`. Certifique-se de incluir a `--current-deployment-group-name` opção e substituir o nome do seu arquivo JSON por*`filename`*:

> **Important**
>
> Não se esqueça de incluir `file://` antes do nome de arquivo. Ele é obrigatório nesse comando.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
```

No final do processo de criação, você receberá uma mensagem de notificação de teste indicando que tanto as permissões quanto os detalhes do gatilho estão configurados corretamente.

Excluir um gatilho de um grupo CodeDeploy de implantação

Como há um limite de 10 gatilhos por grupo de implantação, talvez você queira excluir disparadores se eles não estiverem em uso. Não é possível desfazer a exclusão de um trigger, mas é possível recriar um.

Excluir um gatilho de um grupo de implantação (console)

1. Faça login no Console de gerenciamento da AWS e abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Faça login com o mesmo usuário que você configurou em Começando com CodeDeploy.

2. No painel de navegação, expanda a opção Implantar e escolha Aplicativos.
3. Na página Applications (Aplicativos), escolha o nome do aplicativo associado ao grupo de implantação do qual você deseja excluir um gatilho.
4. Na página Application details (Detalhes do aplicativo), escolha o grupo de implantação do qual você deseja excluir um gatilho.
5. Escolha Editar.
6. Expanda Avançado - opcional.
7. Na área Triggers (Gatilhos), escolha o gatilho que deseja excluir e selecione Delete trigger (Excluir gatilho).
8. Escolha Salvar alterações.

Excluir um gatilho de um grupo de implantação (CLI)

Para usar a CLI para excluir um gatilho, chame o comando update-deployment-group com parâmetros de configuração de gatilho vazios, especificando:

- O nome do aplicativo associado ao grupo de implantação. Para ver uma lista de nomes de aplicativos, chame o comando list-applications.
- O nome do grupo de implantação associado ao aplicativo. Para ver uma lista de nomes de grupos de implantação, chame o comando list-deployment-groups.

Por exemplo:

```
aws deploy update-deployment-group --application-name application-name --current-
deployment-group-name deployment-group-name --trigger-configurations
```

Formatos de dados JSON para acionadores CodeDeploy

Você pode usar a saída JSON que é criada quando um gatilho para uma implantação ou instância é ativado em um fluxo de trabalho de notificação personalizado, como enviar mensagens a filas do Amazon SQS ou invocar uma função no AWS Lambda.

> **Note**
>
> Este guia não aborda como configurar notificações usando o JSON. Para obter informações sobre o uso do Amazon SNS para enviar mensagens para filas do Amazon SQS, consulte Envio de mensagens do Amazon SNS para filas do Amazon SQS. Para obter informações sobre o uso do Amazon SNS para invocar uma função do Lambda, consulte Invocação de funções do Lambda usando notificações do Amazon SNS.

Os exemplos a seguir mostram a estrutura da saída JSON disponível com gatilhos do CodeDeploy.

Exemplo de saída JSON para acionadores Instance-Based

```
{
    "region": "us-east-2",
    "accountId": "111222333444",
    "eventTriggerName": "trigger-group-us-east-instance-succeeded",
    "deploymentId": "d-75I7MBT7C",
    "instanceId": "arn:aws:ec2:us-east-2:444455556666:instance/i-496589f7",
    "lastUpdatedAt": "1446744207.564",
    "instanceStatus": "Succeeded",
    "lifecycleEvents": [
        {
            "LifecycleEvent": "ApplicationStop",
            "LifecycleEventStatus": "Succeeded",
            "StartTime": "1446744188.595",
            "EndTime": "1446744188.711"
        },
        {
            "LifecycleEvent": "BeforeInstall",
            "LifecycleEventStatus": "Succeeded",
            "StartTime": "1446744189.827",
            "EndTime": "1446744190.402"
        }
//More lifecycle events might be listed here
```

```
    ]
}
```

Exemplo de saída JSON para acionadores Deployment-Based

```
{
    "region": "us-west-1",
    "accountId": "111222333444",
    "eventTriggerName": "Trigger-group-us-west-3-deploy-failed",
    "applicationName": "ProductionApp-us-west-3",
    "deploymentId": "d-75I7MBT7C",
    "deploymentGroupName": "dep-group-def-456",
    "createTime": "1446744188.595",
    "completeTime": "1446744190.402",
    "deploymentOverview": {
        "Failed": "10",
        "InProgress": "0",
        "Pending": "0",
        "Skipped": "0",
        "Succeeded": "0"
    },
    "status": "Failed",
    "errorInformation": {
        "ErrorCode": "IAM_ROLE_MISSING",
        "ErrorMessage": "IAM Role is missing for deployment group: dep-group-def-456"
    }
}
```

Segurança em AWS CodeDeploy

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem.

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Third-party auditores testam e verificam regularmente a eficácia de nossa segurança como parte dos programas de AWS conformidade. Para saber mais sobre os programas de conformidade aplicáveis AWS CodeDeploy, consulte AWS Serviços no escopo por programa de conformidade.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar CodeDeploy. Os tópicos a seguir mostram como configurar para atender CodeDeploy aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus CodeDeploy recursos.

Tópicos

- Proteção de dados em AWS CodeDeploy
- Gerenciamento de identidade e acesso para AWS CodeDeploy
- Registro e monitoramento em CodeDeploy
- Validação de conformidade AWS CodeDeploy
- Resiliência em AWS CodeDeploy
- Segurança da infraestrutura em AWS CodeDeploy

Proteção de dados em AWS CodeDeploy

O AWS modelo de responsabilidade compartilhada se aplica à proteção de dados no AWS CodeDeploy. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura

global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte Perguntas frequentes sobre privacidade de dados . Para obter informações sobre proteção de dados na Europa, consulte o Centro de Regulamento Geral sobre a Proteção de Dados (RGPD).

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com CodeDeploy ou Serviços da AWS usa o console, a API ou AWS os SDKs. AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Privacidade do tráfego entre redes

CodeDeploy é um serviço de implantação totalmente gerenciado que oferece suporte a instâncias EC2, funções Lambda, Amazon ECS e servidores locais. Para instâncias do EC2 e servidores locais, um agente baseado em host se comunica usando TLS. CodeDeploy

Atualmente, a comunicação do agente com o serviço requer uma conexão de saída com a Internet para que o agente possa se comunicar com o público CodeDeploy e com os endpoints do serviço Amazon S3. Em uma nuvem privada virtual, isso pode ser feito com um gateway da Internet, com uma conexão VPN de site a site com a rede corporativa ou com uma conexão direta.

O CodeDeploy agente oferece suporte a proxies HTTP.

Os endpoints Amazon VPC, desenvolvidos por AWS PrivateLink, estão disponíveis CodeDeploy em determinadas regiões. Para obter detalhes, consulte Use CodeDeploy com a Amazon Virtual Private Cloud.

> **Note**
>
> O CodeDeploy agente é necessário somente se você implantar em uma plataforma de EC2/ On-premises computação da Amazon. O agente não é necessário para implantações que usam o Amazon ECS ou a plataforma de AWS Lambda computação.

Criptografia em repouso

O código do cliente não está armazenado em CodeDeploy. Como serviço de implantação, CodeDeploy está enviando comandos para o CodeDeploy agente executado em instâncias do EC2 ou servidores locais. Em seguida, o CodeDeploy agente executa os comandos usando o TLS. Os dados do modelo de serviço para implantações, configuração de implantação, grupos de implantação, aplicativos e revisões de aplicativos são armazenados no Amazon DynamoDB e criptografados em repouso usando Chave pertencente à AWS um, de propriedade e gerenciado pela. CodeDeploy Para obter mais informações, consulte Chave pertencente à AWS s.

Criptografia em trânsito

O CodeDeploy agente inicia toda a comunicação com a CodeDeploy porta 443. O agente sonda o CodeDeploy e escuta um comando. O CodeDeploy agente é de código aberto. Toda a comunicação

de serviço para serviço e de cliente para serviço é criptografada em trânsito usando TLS. Isso protege os dados do cliente em trânsito entre outros CodeDeploy serviços, como o Amazon S3.

Gerenciamento de chave de criptografia

Não há chaves de criptografia para você gerenciar. Os dados do modelo de CodeDeploy serviço são criptografados usando um Chave pertencente à AWS, de propriedade e gerenciado por CodeDeploy. Para obter mais informações, consulte Chave pertencente à AWS s.

Gerenciamento de identidade e acesso para AWS CodeDeploy

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar CodeDeploy os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Público
- Autenticação com identidades
- Gerenciar o acesso usando políticas
- Como AWS CodeDeploy funciona com o IAM
- AWS políticas gerenciadas (predefinidas) para CodeDeploy
- CodeDeploy atualizações para AWS políticas gerenciadas
- AWS CodeDeploy exemplos de políticas baseadas em identidade
- Solução de problemas AWS CodeDeploy identidade e acesso
- CodeDeploy referência de permissões
- Cross-service prevenção delegada confusa

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas AWS CodeDeploy identidade e acesso).

- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como AWS CodeDeploy funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte AWS CodeDeploy exemplos de políticas baseadas em identidade)

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário-raiz

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Usuários e grupos

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Identity-based políticas

Identity-based políticas são documentos de políticas de permissões JSON que você anexa a uma identidade (usuário, grupo ou função). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Identity-based as políticas podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de Controle de Serviços (SCPs): as SCPs especificam o número máximo de permissões para uma organização ou uma unidade organizacional no AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs): definem o número máximo de permissões disponíveis para recursos em suas contas. Consulte mais informações em Resource control policies (RCPs) no Guia do usuário do AWS Organizations .
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como AWS CodeDeploy funciona com o IAM

Antes de usar o IAM para gerenciar o acesso CodeDeploy, você deve entender quais recursos do IAM estão disponíveis para uso CodeDeploy. Para obter mais informações, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- CodeDeploy políticas baseadas em identidade
- CodeDeploy políticas baseadas em recursos
- Autorização baseada em CodeDeploy tags
- CodeDeploy Funções do IAM

CodeDeploy políticas baseadas em identidade

Com as políticas baseadas em identidade do IAM, você pode especificar ações e recursos permitidos ou negados e as condições sob as quais as ações são permitidas ou negadas. CodeDeploy suporta ações, recursos e chaves de condição. Para obter informações sobre os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações de política CodeDeploy usam o `codedeploy:` prefixo antes da ação. Por exemplo, a permissão `codedeploy:GetApplication` concede ao usuário permissões para executar a operação `GetApplication`. As declarações de política devem incluir um `NotAction` elemento `Action` ou. CodeDeploy define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

```
"Action": [
      "codedeploy:action1",
      "codedeploy:action2"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, inclua a seguinte ação para especificar todas as ações que começam com a palavra `Describe`:

```
"Action": "ec2:Describe*"
```

Para ver uma lista de CodeDeploy ações, consulte Ações definidas por AWS CodeDeploy no Guia do usuário do IAM.

Para ver uma tabela que lista todas as ações da CodeDeploy API e os recursos aos quais elas se aplicam, consulteCodeDeploy referência de permissões.

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Por exemplo, você pode indicar um grupo de implantação (*myDeploymentGroup*) em sua declaração usando seu ARN da seguinte forma:

```
"Resource": "arn:aws:codedeploy:us-
west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

Também é possível especificar todos os grupos de implantação que pertencem a uma conta usando o caractere curinga (*) da seguinte maneira:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Para especificar todos os recursos, ou se uma ação de API não oferece suporte aos ARNs, use o caractere curinga (*) no elemento `Resource`, como se segue:

```
"Resource": "*"
```

Algumas ações CodeDeploy da API aceitam vários recursos (por exemplo,`BatchGetDeploymentGroups`). Para especificar vários recursos em uma única declaração, separe seus ARNs com vírgulas, como se segue:

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy fornece um conjunto de operações para trabalhar com os CodeDeploy recursos. Para ver uma lista das operações disponíveis, consulte CodeDeploy referência de permissões.

Para ver uma lista dos tipos de CodeDeploy recursos e seus ARNs, consulte Resources Defined by AWS CodeDeploy no Guia do usuário do IAM. Para obter informações sobre as ações nas quais é possível especificar o ARN de cada recurso, consulte Ações definidas pelo AWS CodeDeploy.

CodeDeploy recursos e operações

Em CodeDeploy, o recurso principal é um grupo de implantação. Em uma política, você usa um Amazon Resource Name (ARN) para identificar o recurso ao qual a política se aplica. CodeDeploy oferece suporte a outros recursos que podem ser usados com grupos de implantação, incluindo aplicativos, configurações de implantação e instâncias. Estes são chamados de sub-recursos. Esses recursos e sub-recursos têm ARNs exclusivos associados a eles. Para obter mais informações, consulte Nomes dos recursos da Amazon (ARNs) no Referência geral da Amazon Web Services.

Tipo de atributo	Formato ARN
Grupo de implantação	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentgroup:` *`application-name`* `/`*`deployment-group-name`*
Aplicação	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:application:` *`application-name`*
Configuração de implantação	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentconfig:` *`deployment-configuration-name`*
Instância	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:instance/` *`instance-ID`*
Todos os CodeDeploy recursos	`arn:aws:codedeploy:*`
Todos os CodeDeploy recursos pertencentes à conta especificada na região especificada	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:*`

> **Note**
>
> A maioria dos serviços em AWS trata dois pontos (:) ou uma barra invertida (/) como o mesmo caractere em ARNs. No entanto, CodeDeploy usa uma correspondência exata nos padrões e regras dos recursos. Use os caracteres corretos do ARN ao criar padrões de evento, de modo que eles correspondam à sintaxe do ARN no recurso.

Chaves de condição

CodeDeploy não fornece nenhuma chave de condição específica do serviço, mas oferece suporte ao uso de algumas chaves de condição globais. Para obter mais informações, consulte Chaves de contexto de condição global da AWS no Guia do usuário do IAM.

Exemplos

Para ver exemplos de políticas CodeDeploy baseadas em identidade, consulte. AWS CodeDeploy exemplos de políticas baseadas em identidade

CodeDeploy políticas baseadas em recursos

CodeDeploy não oferece suporte a políticas baseadas em recursos. Para ver um exemplo de uma página detalhada de políticas baseadas em recursos, consulte Usando políticas baseadas em recursos para. AWS Lambda

Autorização baseada em CodeDeploy tags

CodeDeploy não oferece suporte à marcação de recursos ou ao controle de acesso com base em tags.

CodeDeploy Funções do IAM

Uma função do IAM é uma entidade na sua AWS conta que tem permissões específicas.

Usando credenciais temporárias com CodeDeploy

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

CodeDeploy suporta o uso de credenciais temporárias.

Service-linked funções

CodeDeploy não oferece suporte a funções vinculadas a serviços.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. As funções de serviço aparecem na sua AWS conta e são de propriedade da conta. Isso significa que um usuário pode alterar as permissões para esse perfil. Porém, fazer isso pode alterar a funcionalidade do serviço.

CodeDeploy suporta funções de serviço.

Escolha de uma função do IAM em CodeDeploy

Ao criar um recurso de grupo de implantação em CodeDeploy, você deve escolher uma função para permitir o acesso CodeDeploy ao Amazon EC2 em seu nome. Caso já tenha criado uma função de serviço ou uma função vinculada ao serviço, o CodeDeploy fornecerá uma lista das funções para que você escolha. É importante escolher uma função que conceda acesso para iniciar e interromper instâncias do EC2.

AWS políticas gerenciadas (predefinidas) para CodeDeploy

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas AWS gerenciadas concedem permissões para casos de uso comuns para que você não precise investigar quais permissões são necessárias. Para saber mais, consulte AWS Políticas gerenciadas pela no Guia do usuário do IAM.

Tópicos

- Lista de AWS políticas gerenciadas para CodeDeploy
- CodeDeploy políticas e notificações gerenciadas

Lista de AWS políticas gerenciadas para CodeDeploy

As seguintes políticas AWS gerenciadas, que você pode anexar aos usuários em sua conta, são específicas para CodeDeploy:

- `AWSCodeDeployFullAccess`: concede acesso total ao CodeDeploy.

> **Note**
>
> AWSCodeDeployFullAccess não fornece permissões para operações em outros serviços necessários para implantar seus aplicativos, como Amazon EC2 e Amazon S3, somente para operações específicas de. CodeDeploy

- `AWSCodeDeployDeployerAccess`: concede permissão para registrar e implantar revisões.

- `AWSCodeDeployReadOnlyAccess`: concede acesso somente leitura ao CodeDeploy.

- `AWSCodeDeployRole`: CodeDeploy Permite:
  - leia as tags nas instâncias ou identifique as instâncias do Amazon EC2 pelos nomes de grupo do Amazon EC2 Auto Scaling.
  - leia, crie, atualize e exclua grupos, ganchos do ciclo de vida, políticas de escalabilidade e recursos de grupos de alta atividade do Amazon EC2 Auto Scaling
  - publique informações nos tópicos do Amazon SNS.
  - recuperar informações sobre os alarmes da Amazon CloudWatch
  - leia e atualize recursos no serviço Elastic Load Balancing

  A política contém o código a seguir:

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling:DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling:CreateAutoScalingGroup",
```

```
        "autoscaling:CreateOrUpdateTags",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DeleteAutoScalingGroup",
        "autoscaling:PutWarmPool",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:TerminateInstances",
        "tag:GetResources",
        "sns:Publish",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeLoadBalancerAttributes",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetGroupAttributes",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

- `AWSCodeDeployRoleForLambda`: concede CodeDeploy permissão para acessar AWS Lambda e qualquer outro recurso necessário para uma implantação.

- `AWSCodeDeployRoleForECS`: concede CodeDeploy permissão para acessar o Amazon ECS e qualquer outro recurso necessário para uma implantação.

- `AWSCodeDeployRoleForECSLimited`: concede CodeDeploy permissão para acessar o Amazon ECS e qualquer outro recurso necessário para uma implantação, com as seguintes exceções:
  - Na `hooks` seção do AppSpec arquivo, somente funções Lambda com nomes que começam com `CodeDeployHook_` podem ser usadas. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.
  - O acesso ao bucket do S3 é limitado a buckets do S3 com uma tag de registro, `UseWithCodeDeploy`, com um valor de `true`. Para obter mais informações, consulte Marcação de objetos.
- `AmazonEC2RoleforAWSCodeDeployLimited`: concede CodeDeploy permissão para obter e listar objetos em um bucket CodeDeploy do Amazon S3. A política contém o código a seguir:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*/CodeDeploy/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
                }
            }
        }
    ]
}
```

As permissões para alguns aspectos do processo de implantação são concedidas a dois outros tipos de função que atuam em nome de CodeDeploy:

- Um perfil de instância do IAM é um perfil do IAM que você anexa às suas instâncias do Amazon EC2. Esse perfil inclui as permissões necessárias para acessar os buckets ou GitHub repositórios do Amazon S3 onde os aplicativos são armazenados. Para obter mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.
- Uma função de serviço é uma função do IAM que concede permissões a um AWS serviço para que ele possa acessar AWS recursos. As políticas que você anexa à função de serviço determinam quais AWS recursos o serviço pode acessar e as ações que ele pode realizar com esses recursos. Para CodeDeploy, uma função de serviço é usada para o seguinte:
  - Para ler as tags aplicadas às instâncias ou aos nomes de grupo do Amazon EC2 Auto Scaling associados a essas instâncias. Isso permite CodeDeploy identificar instâncias nas quais ele pode implantar aplicativos.
  - Para realizar operações em instâncias, grupos do Amazon EC2 Auto Scaling e balanceadores de carga do Elastic Load Balancing.
  - Para publicar informações nos tópicos do Amazon SNS para que as notificações possam ser enviadas quando ocorrerem eventos específicos de implantação ou instância.
  - Para recuperar informações sobre CloudWatch alarmes para configurar o monitoramento de alarmes para implantações.

  Para obter mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.

Você também pode criar políticas personalizadas do IAM para conceder permissões para CodeDeploy ações e recursos. Você anexa essas políticas personalizadas aos perfis do IAM e, em seguida, atribui os perfis aos usuários ou grupos que precisam das permissões.

CodeDeploy políticas e notificações gerenciadas

CodeDeploy oferece suporte a notificações para informar os usuários sobre mudanças importantes nas implantações. Políticas gerenciadas para CodeDeploy incluir declarações de política para funcionalidade de notificação. Para obter mais informações, consulte O que são notificações?.

Permissões para notificações em políticas gerenciadas de acesso total

A política gerenciada `AWSCodeDeployFullAccess` inclui as declarações a seguir para permitir acesso total às notificações. Os usuários com essa política gerenciada aplicada também podem criar e gerenciar tópicos do Amazon SNS para notificações, inscrever e cancelar a assinatura de usuários em tópicos e listar tópicos para escolher como destinos para regras de notificação.

```
    {
        "Sid": "CodeStarNotificationsReadWriteAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:DeleteNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe"
        ],
        "Resource": "*",
        "Condition" : {
            "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codedeploy:*:*:application:*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
        "Effect": "Allow",
```

```
        "Action": [
            "sns:CreateTopic",
            "sns:SetTopicAttributes"
        ],
        "Resource": "arn:aws:sns:*:*:codestar-notifications*"
    },
    {
      "Sid" : "CodeStarNotificationsChatbotAccess",
      "Effect" : "Allow",
      "Action" : [
        "chatbot:DescribeSlackChannelConfigurations"
      ],
      "Resource" : "*"
    },
    {
        "Sid": "SNSTopicListAccess",
        "Effect": "Allow",
        "Action": [
            "sns:ListTopics"
        ],
        "Resource": "*"
    }
```

Permissões para notificações em políticas gerenciadas somente leitura

A política gerenciada `AWSCodeDeployReadOnlyAccess` inclui as instruções a seguir para permitir acesso somente leitura às notificações. Os usuários com essa política gerenciada aplicada podem visualizar notificações de recursos, mas não podem criá-los, gerenciá-los ou assiná-los.

```
   {
        "Sid": "CodeStarNotificationsPowerUserAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:DescribeNotificationRule"
        ],
        "Resource": "*",
        "Condition" : {
            "ArnLike" : {"codestar-notifications:NotificationsForResource" :
 "arn:aws:codedeploy:*:*:application:*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
```

```
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules"
        ],
        "Resource": "*"
    }
```

Permissões relacionadas a notificações em outras políticas gerenciadas

A política gerenciada `AWSCodeDeployDeployerAccess` inclui as seguintes declarações para permitir que os usuários criem, atualizem, assinem e visualizem notificações de recursos, mas não podem excluí-las. Os usuários com essa política gerenciada aplicada também podem criar e gerenciar tópicos do Amazon SNS para notificações.

Esta política inclui permissões para fazer o seguinte:

- `codestar-notifications:CreateNotificationRule`: permite que as entidades principais criem notificações.
- `codestar-notifications:DescribeNotificationRule`: permite que as entidades principais recuperem informações sobre notificações.
- `codestar-notifications:UpdateNotificationRule`: permite que as entidades principais atualizem notificações.
- `codestar-notifications:Subscribe`: permite que as entidades principais assinem atualizações de notificações.
- `codestar-notifications:Unsubscribe`: permite que as entidades principais cancelem a assinatura de atualizações de notificações.
- `codestar-notifications:ListNotificationRules`: permite que as entidades principais recuperem a lista de regras de notificação.
- `codestar-notifications:ListTargets`: permite que as entidades principais recuperem a lista de destinos.
- `codestar-notifications:ListTagsforResource`: permite que as entidades principais recuperem a lista de tags.
- `codestar-notifications:ListEventTypes`: permite que as entidades principais recuperem a lista de tipos de evento.
- `chatbot:DescribeSlackChannelConfiguration`: permite que as entidades principais recuperem informações sobre as configurações do canal do Slack.

- `sns:ListTopics`: permite que as entidades principais recuperem a lista de tópicos do Amazon SNS para notificações.

```
  {
     "Sid" : "CodeStarNotificationsReadWriteAccess",
     "Effect" : "Allow",
     "Action" : [
       "codestar-notifications:CreateNotificationRule",
       "codestar-notifications:DescribeNotificationRule",
       "codestar-notifications:UpdateNotificationRule",
       "codestar-notifications:Subscribe",
       "codestar-notifications:Unsubscribe"
     ],
     "Resource" : "*",
     "Condition" : {
       "ArnLike" : {
         "codestar-notifications:NotificationsForResource" :
"arn:aws:codedeploy:*:*:application:*"
       }
     }
   },
   {
     "Sid" : "CodeStarNotificationsListAccess",
     "Effect" : "Allow",
     "Action" : [
       "codestar-notifications:ListNotificationRules",
       "codestar-notifications:ListTargets",
       "codestar-notifications:ListTagsforResource",
       "codestar-notifications:ListEventTypes"
     ],
     "Resource" : "*"
   },
   {
     "Sid" : "CodeStarNotificationsChatbotAccess",
     "Effect" : "Allow",
     "Action" : [
       "chatbot:DescribeSlackChannelConfigurations"
     ],
     "Resource" : "*"
   },
   {
     "Sid" : "SNSTopicListAccess",
```

```
      "Effect" : "Allow",
      "Action" : [
        "sns:ListTopics"
      ],
      "Resource" : "*"
    }
```

Para obter mais informações, consulte Gerenciamento de identidade e acesso para CodeStar notificações da AWS.

CodeDeploy atualizações para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas CodeDeploy desde que esse serviço começou a rastrear essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed de RSS em CodeDeploy Histórico do documento.

Alteração	Descrição	Data
Política gerenciada pelo `AWSCodeDeployDeployerAccess` : atualização para uma política existente	Atualização da ação `codestar-notifications:NotificationsForResource` para comportar alterações de validação da política do IAM. O recurso original `arn:aws:codedeploy:*` foi atualizado para `arn:aws:codedeploy:*:*:application:*` . Para obter mais informações sobre essa política, consulte Permissões relacionadas a notificações em outras políticas gerenciadas.	16 de dezembro de 2024
Política gerenciada pelo `AWSCodeDeployReadO`	Atualização da ação `codestar-notifications:Notifications`	16 de dezembro de 2024

Alteração	Descrição	Data
`nlyAccess` : atualização para uma política existente	`ForResource` para comportar alterações de validação da política do IAM. O recurso original `arn:aws:codedeploy:*` foi atualizado para `arn:aws:codedeploy:*:*:application:*` . Para obter mais informações sobre essa política, consulte Permissões para notificações em políticas gerenciadas somente leitura.	
Política gerenciada pelo `AWSCodeDeployFullAccess` : atualização para uma política existente	Atualização da ação `codestar-notifications:NotificationsForResource` para comportar alterações de validação da política do IAM. O recurso original `arn:aws:codedeploy:*` foi atualizado para `arn:aws:codedeploy:*:*:application:*` . Para obter mais informações sobre essa política, consulte Permissões para notificações em políticas gerenciadas de acesso total.	16 de dezembro de 2024

Alteração	Descrição	Data
Política gerenciada pelo `AWSCodeDeployRole` : atualização para uma política existente	As `elasticloadbalancing:DescribeLoadBalancerAttributes` ações e `elasticloadbalancing:DescribeTargetGroupAttributes` foram adicionadas à declaração de política para apoiar as alterações do Elastic Load Balancing. Para obter mais informações sobre essa política, consulte AWSCodeDeployRole.	16 de agosto de 2023
Política gerenciada pelo `AWSCodeDeployFullAccess` : atualização para uma política existente	A ação `chatbot:ListMicrosoftTeamsChannelConfigurations` foi adicionada à declaração de política para apoiar as alterações nas notificações. Para obter mais informações sobre essa política, consulte AWSCodeDeployRole.	11 de maio de 2023

Alteração	Descrição	Data
Política gerenciada pelo `AWSCodeDeployRole` : atualização para uma política existente	A ação `autoscaling:CreateOrUpdateTags` foi adicionada à declaração de política para apoiar as alterações de autorização do Amazon EC2 Auto Scaling. Para obter mais informações sobre essa política, consulte AWSCodeDeployRole.	3 de fevereiro de 2023
Política gerenciada pelo `AmazonEC2RoleforAWSCodeDeployLimited` : atualização para uma política existente	A ação `s3:ListBucket` foi removida da declaração de política que inclui a condição `s3:ExistingObjectTag/UseWithCodeDeploy` . Para obter mais informações sobre essa política, consulte AmazonEC2RoleforAWSCodeDeployLimited.	22 de novembro de 2021
Política gerenciada pelo `AWSCodeDeployRole` : atualização para uma política existente	Foi adicionada a `autoscaling:PutWarmPool` ação para oferecer suporte à adição de pools quentes aos grupos do Amazon EC2 Auto Scaling blue/green para implantações. Ações duplicadas desnecessárias foram removidas.	18 de maio de 2021

Alteração	Descrição	Data
CodeDeploy começou a rastrear alterações	CodeDeploy começou a rastrear as mudanças em suas políticas AWS gerenciadas.	18 de maio de 2021

AWS CodeDeploy exemplos de políticas baseadas em identidade

Por padrão, os usuários não têm permissão para criar ou modificar CodeDeploy recursos. Eles também não podem realizar tarefas usando a AWS API Console de gerenciamento da AWS AWS CLI, ou. Você deve criar políticas do IAM que concedam aos perfis do IAM permissão para executar operações da API nos recursos necessários especificados. Em seguida, você precisa anexar esses perfis do IAM aos usuários ou grupos que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM utilizando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Em CodeDeploy, políticas baseadas em identidade são usadas para gerenciar permissões para os vários recursos relacionados ao processo de implantação. É possível controlar o acesso aos seguintes tipos de recursos:

- Aplicativos e revisões de aplicativo.
- Implantações.
- Configurações de implantação.
- Instâncias e instâncias no local.

As capacidades controladas por políticas de recursos variam de acordo com o tipo de recurso, conforme descrito na tabela a seguir:

Resource types	Capacidades
Todos	Visualizar e listar detalhes sobre recursos
Aplicativos Configurações de implantação	Criar recursos do Excluir recursos.

Resource types	Capacidades
Grupos de implantação	
Implantações	Criar implantações Parar implantações
Revisões de aplicativo	Registrar revisões de aplicativo
Aplicativos Grupos de implantação	Atualizar recursos
On-premises instâncias	Adicionar tags a instâncias Remover tags de instâncias Registrar instâncias Cancelar o registro de instâncias

O exemplo a seguir mostra uma política de permissões que permite que um usuário exclua o grupo de implantação chamado **`WordPress_DepGroup`** associado ao aplicativo chamado **`WordPress_App`** na região **`us-west-2`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:DeleteDeploymentGroup"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
      ]
    }
```

```
  ]
}
```

Tópicos

- Customer-managed exemplos de políticas
- Práticas recomendadas de política
- Usando o CodeDeploy console
- Permitir que os usuários visualizem suas próprias permissões

Customer-managed exemplos de políticas

Nesta seção, você pode encontrar exemplos de políticas que concedem permissões para várias CodeDeploy ações. Essas políticas funcionam quando você usa a CodeDeploy API, AWS os SDKs ou o. AWS CLIÉ necessário conceder permissões adicionais para ações executadas no console. Para saber mais sobre a concessão de permissões do console, consulte Usando o CodeDeploy console.

> **Note**
>
> Todos os exemplos usam a Região do Oeste dos EUA (Oregon) (`us-west-2`) e contêm IDs de conta fictícios.

Exemplos

- Exemplo 1: Permitir permissão para realizar CodeDeploy operações em uma única região
- Exemplo 2: permitir o registro de revisões para um único aplicativo
- Exemplo 3: permitir a criação de implantações para um único grupo de implantação

Exemplo 1: Permitir permissão para realizar CodeDeploy operações em uma única região

O exemplo a seguir concede permissões para realizar CodeDeploy operações somente na **`us-west-2`** região:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:*"
      ]
    }
  ]
}
```

Exemplo 2: permitir o registro de revisões para um único aplicativo

O exemplo a seguir concede permissões para registrar revisões de aplicativo para todos os aplicativos que começam com **`Test`** na região **`us-west-2`**:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:RegisterApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:Test*"
      ]
    }
  ]
}
```

Exemplo 3: permitir a criação de implantações para um único grupo de implantação

O exemplo a seguir permite a criação de implantações para o grupo de implantação denominado **WordPress_DepGroup**, associado ao aplicativo denominado **WordPress_App**, a configuração de implantação personalizada denominada **ThreeQuartersHealthy** e qualquer revisão de aplicativos associada ao aplicativo denominado **WordPress_App**. Todos esses recursos estão na região **us-west-2**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:CreateDeployment"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
      ]
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:GetDeploymentConfig"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-
west-2:111122223333:deploymentconfig:ThreeQuartersHealthy"
      ]
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:GetApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:WordPress_App"
      ]
    }
```

```
  ]
}
```

Práticas recomendadas de política

Identity-based as políticas determinam se alguém pode criar, acessar ou excluir CodeDeploy recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando

as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o CodeDeploy console

Se você usa o CodeDeploy console, deve ter um conjunto mínimo de permissões que permita descrever outros AWS recursos para sua AWS conta. Para usar CodeDeploy no CodeDeploy console, você deve ter permissões dos seguintes serviços:

- Amazon EC2 Auto Scaling
- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- AWS Identity and Access Management
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon CloudWatch

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para os usuários que têm um perfil com essa política do IAM. Para garantir que esses usuários ainda possam usar o CodeDeploy console, anexe também a política `AWSCodeDeployReadOnlyAccess` gerenciada à função atribuída ao usuário, conforme descrito emAWS políticas gerenciadas (predefinidas) para CodeDeploy.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a CodeDeploy API.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solução de problemas AWS CodeDeploy identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com CodeDeploy um IAM.

Tópicos

- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta para acessar meus CodeDeploy recursos

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o CodeDeploy.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no CodeDeploy. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta para acessar meus CodeDeploy recursos

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), é possível usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é CodeDeploy compatível com esses recursos, consulteComo AWS CodeDeploy funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

CodeDeploy referência de permissões

Use a tabela a seguir ao configurar o acesso e escrever políticas de permissões que podem ser anexadas a uma identidade do IAM (políticas baseadas em identidade). A tabela lista cada operação da CodeDeploy API, as ações para as quais você pode conceder permissões para realizar a ação e o formato do ARN do recurso a ser usado para conceder permissões. Você especifica as ações no campo `Action` das políticas. Você especifica um ARN, com ou sem um caractere curinga (*), como o valor do recurso no campo `Resource` das políticas.

Você pode usar chaves AWS de condição abrangentes em suas CodeDeploy políticas para expressar condições. Para obter uma lista completa AWS de chaves gerais, consulte Chaves disponíveis no Guia do usuário do IAM.

Para especificar uma ação, use o prefixo `codedeploy:` seguido pelo nome da operação API (por exemplo, `codedeploy:GetApplication` e `codedeploy:CreateApplication`). Para especificar várias ações em uma única declaração, separe-as com vírgulas (por exemplo, `"Action": ["codedeploy:action1", "codedeploy:action2"]`).

Usando caracteres curinga

Você pode usar um caractere curinga (*) no seu ARN para especificar várias ações ou recursos. Por exemplo, `codedeploy:*` especifica todas as CodeDeploy ações e `codedeploy:Get*` especifica todas as CodeDeploy ações que começam com a palavra. `Get` O exemplo a seguir concede acesso a todos os grupos de implantação com nomes que começam com `West` e estão associados a aplicativos com nomes que começam com `Test`.

```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

É possível usar curingas com os seguintes recursos listados na tabela:

- *application-name*
- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

Os curingas não podem ser usados com *region* ou*account-id*. Para obter mais informações sobre curingas, consulte Identificadores do IAM, no Manual do usuário do IAM.

> **Note**
>
> No ARN de cada ação, dois-pontos (:) vêm logo após o recurso. Você também pode usar uma barra (/) para vir logo após o recurso. Para obter mais informações, consulte CodeDeployexemplos de ARNs.

CodeDeploy Operações de API e permissões necessárias para ações

AddTagsToOnPremisesInstances

Ação: `codedeploy:AddTagsToOnPremisesInstances`

Necessário para adicionar tags a uma ou mais instâncias on-premises.

Recurso: `arn:aws:codedeploy:`*region*`:`*account-id*`:instance/`*instance-ID*

BatchGetApplicationRevisions

Ação: `codedeploy:BatchGetApplicationRevisions`

Necessário para obter informações sobre várias revisões de aplicativo associadas ao usuário do .

Recurso: `arn:aws:codedeploy:`*region*`:`*account-id*`:application:`*application-name*

BatchGetApplications

Ação: `codedeploy:BatchGetApplications`

Necessário para obter informações sobre vários aplicativos associados ao usuário do .

Recurso: `arn:aws:codedeploy:`*region*`:`*account-id*`:application:*`

BatchGetDeploymentGroups

Ação: `codedeploy:BatchGetDeploymentGroups`

Necessário para obter informações sobre vários grupos de implantação associados ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetDeploymentInstances

Ação: `codedeploy:BatchGetDeploymentInstances`

Necessário para obter informações sobre uma ou mais instâncias que fazem parte de um grupo de implantação.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetDeployments

Ação: `codedeploy:BatchGetDeployments`

Necessário para obter informações sobre várias implantações associadas ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetOnPremisesInstances

Ação: `codedeploy:BatchGetOnPremisesInstances`

Necessário para obter informações sobre uma ou mais instâncias on-premises.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:*`

ContinueDeployment

Ação: `codedeploy:ContinueDeployment`

Necessário durante uma blue/green implantação para começar a registrar instâncias em um ambiente substituto com um balanceador de carga do Elastic Load Balancing.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateApplication

Ação: `codedeploy:CreateApplication`

Necessário para criar um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

CreateDeployment[1]

Ação: `codedeploy:CreateDeployment`

Necessário para criar uma implantação para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateDeploymentConfig

Ação: `codedeploy:CreateDeploymentConfig`

Necessário para criar uma configuração de implantação personalizada associada ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

CreateDeploymentGroup

Ação: `codedeploy:CreateDeploymentGroup`

Necessário para criar um grupo de implantação para um aplicativo associado ao usuário.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeleteApplication

Ação: `codedeploy:DeleteApplication`

Necessário para excluir um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

DeleteDeploymentConfig

Ação: `codedeploy:DeleteDeploymentConfig`

Necessário para excluir uma configuração de implantação personalizada associada ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

DeleteDeploymentGroup

Ação: `codedeploy:DeleteDeploymentGroup`

Necessário para excluir um grupo de implantação para um aplicativo associado ao usuário.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeregisterOnPremisesInstance

Ação: `codedeploy:DeregisterOnPremisesInstance`

Necessário para cancelar o registro de uma instância on-premises.

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

GetApplication

Ação: `codedeploy:GetApplication`

Necessário para obter informações sobre um único aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

GetApplicationRevision

Ação: `codedeploy:GetApplicationRevision`

Necessário para obter informações sobre uma única revisão de aplicativo para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

GetDeployment

Ação: `codedeploy:GetDeployment`

Necessário para obter informações sobre uma única implantação em um grupo de implantação para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetDeploymentConfig

Ação: `codedeploy:GetDeploymentConfig`

Necessário para obter informações sobre uma única configuração de implantação associada ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

GetDeploymentGroup

Ação: `codedeploy:GetDeploymentGroup`

Necessário para obter informações sobre um único grupo de implantação para um aplicativo associado ao usuário.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetDeploymentInstance

Ação: `codedeploy:GetDeploymentInstance`

Necessário para obter informações sobre uma única instância em uma implantação associada ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetOnPremisesInstance

Ação: `codedeploy:GetOnPremisesInstance`

Necessário para obter informações sobre uma única instância on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

ListApplicationRevisions

Ação: `codedeploy:ListApplicationRevisions`

Necessário para obter informações sobre todas as revisões de aplicativo para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:application:*`

ListApplications

Ação: `codedeploy:ListApplications`

Necessário para obter informações sobre todos os aplicativos associados ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:application:*`

ListDeploymentConfigs

Ação: `codedeploy:ListDeploymentConfigs`

Necessário para obter informações sobre todas as configurações de implantação associadas ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

ListDeploymentGroups

Ação: `codedeploy:ListDeploymentGroups`

Necessário para obter informações sobre todos os grupos de implantação de um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

ListDeploymentInstances

Ação: `codedeploy:ListDeploymentInstances`

Obrigatório para obter informações sobre todas as instâncias em uma implantação associada ao usuário.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListDeployments

Ação: `codedeploy:ListDeployments`

Necessário para obter informações sobre todas as implantações em um grupo de implantação associado ao usuário, ou para obter todas as implantações associadas ao usuário.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListGitHubAccountTokenNames

Ação: `codedeploy:ListGitHubAccountTokenNames`

Necessário para obter uma lista dos nomes das conexões armazenadas GitHub nas contas.

Recurso: `arn:aws:codedeploy:region:account-id:*`

ListOnPremisesInstances

Ação: `codedeploy:ListOnPremisesInstances`

Necessário para obter uma lista de um ou mais nomes de instâncias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:*`

RegisterApplicationRevision

Ação: `codedeploy:RegisterApplicationRevision`

Necessário para registrar informações sobre uma revisão de aplicativo para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

RegisterOnPremisesInstance

Ação: `codedeploy:RegisterOnPremisesInstance`

Necessário para registrar uma instância local com CodeDeploy.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

RemoveTagsFromOnPremisesInstances

Ação: `codedeploy:RemoveTagsFromOnPremisesInstances`

Necessário para remover tags de uma ou mais instâncias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

SkipWaitTimeForInstanceTermination

Ação: `codedeploy:SkipWaitTimeForInstanceTermination`

Necessário para substituir um tempo de espera especificado e começar a encerrar instâncias no ambiente original imediatamente após o tráfego ser roteado com sucesso em uma implantação. blue/green

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

StopDeployment

Ação: `codedeploy:StopDeployment`

Necessário para interromper uma implantação em andamento em um grupo de implantação para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

UpdateApplication[3]

Ação: `codedeploy:UpdateApplication`

Necessário para alterar informações sobre um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

UpdateDeploymentGroup[3]

Ação: `codedeploy:UpdateDeploymentGroup`

Necessário para alterar informações sobre um único grupo de implantação para um aplicativo associado ao usuário do .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

[1] Ao especificar permissões `CreateDeployment`, também é necessário especificar permissões `GetDeploymentConfig` para a configuração de implantação e permissões `GetApplicationRevision` ou `RegisterApplicationRevision` para a revisão do aplicativo.

[2] Válido para `ListDeployments` ao fornecer um grupo de implantação, mas não ao listar todas as implantações associadas ao usuário.

[3] Para `UpdateApplication`, é necessário ter permissões `UpdateApplication` para o nome do aplicativo novo e antigo. Para ações `UpdateDeploymentGroup` que envolvem a alteração do nome

de um grupo de implantação, é necessário ter permissões `UpdateDeploymentGroup` para o nome do grupo de implantação novo e antigo.

Cross-service prevenção delegada confusa

O problema do “confused deputy” é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executar a ação. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. Cross-service a representação 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, AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com diretores de serviços que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição SourceAccount global aws: SourceArn e aws: nas políticas de recursos para limitar as permissões que CodeDeploy concedem outro serviço ao recurso. Se você utilizar ambas as chaves de contexto de condição global e o valor de `aws:SourceArn` contiver o ID da conta, o valor de `aws:SourceAccount` e a conta no valor de `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma declaração da política. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser que qualquer recurso nessa conta seja associado ao uso entre serviços.

Para EC2/On-Premises implantações do AWS Lambda e do Amazon ECS regulares, o valor de `aws:SourceArn` deve incluir o ARN do CodeDeploy grupo de implantação com o qual pode assumir a função do CodeDeploy IAM.

Para blue/greenimplantações do Amazon ECS criadas por meio de CloudFormation, o valor de `aws:SourceArn` deve incluir o ARN da CloudFormation pilha com o CodeDeploy qual é permitido assumir a função do IAM.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo ou estiver especificando vários recursos, use caracteres curingas (*) para as partes desconhecidas.

Por exemplo, você pode usar a seguinte política de confiança com uma implantação EC2/On-Premises AWS Lambda ou regular do Amazon ECS:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "codedeploy.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:codedeploy:us-
east-1:111122223333:deploymentgroup:myApplication/*"
                }
            }
        }
    ]
}
```

Para uma blue/greenimplantação do Amazon ECS criada por meio de CloudFormation, você pode usar:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "codedeploy.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cloudformation:us-
east-1:111122223333:stack/MyCloudFormationStackName/*"
                }
            }
        }
    ]
}
```

Registro e monitoramento em CodeDeploy

Esta seção fornece uma visão geral do monitoramento, registro em log e resposta a incidentes no CodeDeploy.

Auditoria de todas as interações com CodeDeploy

CodeDeploy é integrado com AWS CloudTrail, um serviço que captura chamadas de API feitas por ou em nome de CodeDeploy sua AWS conta e entrega os arquivos de log para um bucket do S3 que você especificar. CloudTrail captura chamadas de API do CodeDeploy console, de CodeDeploy comandos por meio do ou AWS CLI diretamente das CodeDeploy APIs. Usando as informações coletadas por CloudTrail, você pode determinar para qual solicitação foi feita CodeDeploy, o endereço IP de origem a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e assim por diante. Para saber mais CloudTrail, consulte Como trabalhar com arquivos de CloudTrail log no Guia do AWS CloudTrail usuário.

Você pode visualizar os dados de log criados por uma CodeDeploy implantação configurando o CloudWatch agente da Amazon para visualizar dados agregados no CloudWatch console ou fazendo login em uma instância para revisar o arquivo de log. Para obter mais informações, consulte Enviar registros CodeDeploy do agente para CloudWatch.

Gerenciamento de incidentes e alertas

Você pode usar o Amazon CloudWatch Events para detectar e reagir às mudanças no estado de uma instância ou de uma implantação (um evento) em suas CodeDeploy operações. Em seguida,

com base nas regras que você cria, o CloudWatch Events invoca uma ou mais ações de destino quando uma implantação ou instância entra no estado especificado em uma regra. Dependendo do tipo de alteração de estado, convém enviar notificações, capturar informações de estado, tomar medidas corretivas, iniciar eventos ou realizar outras ações. Você pode selecionar os seguintes tipos de alvos ao usar CloudWatch Eventos como parte de suas CodeDeploy operações:

- AWS Lambda funções
- Streams do Kinesis
- Filas do Amazon SQS
- Built-in alvos (ações CloudWatch de alarme)
- Tópicos do Amazon SNS

A seguir, alguns casos de uso:

- Use uma função do Lambda para transmitir uma notificação a um canal Slack sempre que uma implantação falhar.
- Envie dados sobre implantações ou instâncias a um stream do Kinesis para oferecer suporte ao monitoramento abrangente do status em tempo real.
- Use ações de CloudWatch alarme para interromper, encerrar, reinicializar ou recuperar automaticamente instâncias do EC2 quando ocorrer uma implantação ou evento de instância especificado por você.

Para obter mais informações, consulte O que é Amazon CloudWatch Events no Guia CloudWatch do usuário da Amazon.

Validação de conformidade AWS CodeDeploy

Nossa nova experiência AWS de inscrição não foi projetada para cargas de trabalho regulamentadas. Se você estiver usando nossa nova experiência de AWS inscrição, mas quiser usá-la AWS para cargas de trabalho regulamentadas, você pode se inscrever AWS (avançado) ou ativar recursos avançados para seu AWS ambiente.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a documentação AWS de segurança.

Resiliência em AWS CodeDeploy

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que 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 AWS regiões e zonas de disponibilidade, consulte infraestrutura AWS global.

Segurança da infraestrutura em AWS CodeDeploy

Como serviço gerenciado, AWS CodeDeploy é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa chamadas de API AWS publicadas para acessar CodeDeploy pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (PFS), como Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

As solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Referência

Referência.

Tópicos

- CodeDeploy AppSpec referência de arquivo
- CodeDeploy referência de configuração do agente
- CloudFormation modelos para CodeDeploy referência
- Use CodeDeploy com a Amazon Virtual Private Cloud
- CodeDeploy referência do kit de recursos
- CodeDeploy cotas

CodeDeploy AppSpec referência de arquivo

Esta seção é apenas uma referência. Para obter uma visão geral conceitual do AppSpec arquivo, consulteApplication Specification Files.

O arquivo de especificação do aplicativo (AppSpec arquivo) é um JSON-formatted arquivo formatado em YAML ou usado CodeDeploy para gerenciar uma implantação.

> **Note**
>
> O AppSpec arquivo de uma EC2/On-Premises implantação deve ser nomeado`appspec.yml`, a menos que você esteja executando uma implantação local. Para obter mais informações, consulte Criar uma implantação local.

Tópicos

- AppSpec arquivos em uma plataforma computacional Amazon ECS
- AppSpec arquivos em um AWS Lambda Plataforma de computação
- AppSpec arquivos em uma plataforma de EC2/on-premises computação
- AppSpec Estrutura do arquivo
- AppSpec Exemplo de arquivo

- AppSpec Espaçamento de arquivos
- Valide seu AppSpec arquivo e a localização do arquivo

AppSpec arquivos em uma plataforma computacional Amazon ECS

Para aplicativos da plataforma de computação Amazon ECS, o AppSpec arquivo é usado CodeDeploy para determinar:

- Seu arquivo de definição de tarefa do Amazon ECS. Isso é especificado com seu ARN na `TaskDefinition` instrução no AppSpec arquivo.
- O contêiner e a porta de seu conjunto de tarefas de substituição em que o Application Load Balancer ou o Network Load Balancer redireciona o tráfego durante uma implantação. Isso é especificado com a `LoadBalancerInfo` instrução no AppSpec arquivo.
- Informações opcionais sobre o serviço do Amazon ECS, como a versão da plataforma em que ele é executado, suas sub-redes e seus grupos de segurança.
- Funções do Lambda opcionais que serão executadas durante hooks que correspondam aos eventos de ciclo de vida durante uma implantação do Amazon ECS. Para obter mais informações, consulte AppSpec seção 'hooks' para uma implantação do Amazon ECS.

AppSpec arquivos em um AWS Lambda Plataforma de computação

Para aplicativos AWS da plataforma de computação Lambda, o AppSpec arquivo é usado por para determinar: CodeDeploy

- Qual versão da função do Lambda a ser implantada.
- As funções do Lambda a serem usadas como testes de validação.

Um AppSpec arquivo pode ser YAML-formatted ou JSON-formatted. Você também pode inserir o conteúdo de um AppSpec arquivo diretamente no CodeDeploy console ao criar uma implantação.

AppSpec arquivos em uma plataforma de EC2/on-premises computação

Se seu aplicativo usa a plataforma de EC2/On-Premises computação, o AppSpec arquivo deve ser um YAML-formatted arquivo chamado `appspec.yml` e deve ser colocado na raiz da estrutura de diretórios do código-fonte de um aplicativo. Caso contrário, haverá falha nas implantações. É usado por CodeDeploy para determinar:

- O que ele deve instalar em suas instâncias a partir da revisão do aplicativo no Amazon S3 ou. GitHub
- quais ganchos de evento de ciclo de vida devem ser executados em resposta a eventos de ciclo de vida de implantação.

Depois de concluir um AppSpec arquivo, você o agrupa, junto com o conteúdo a ser implantado, em um arquivo compactado (zip, tar ou tar compactado). Para obter mais informações, consulte Trabalhando com revisões de aplicativos para CodeDeploy.

> **Note**
>
> Os formatos de arquivo tar e tar compactado (.tar e .tar.gz) não são compatíveis com instâncias do Windows Server.

Depois de ter um arquivo compactado (conhecido CodeDeploy como revisão), você o carrega em um bucket do Amazon S3 ou repositório Git. Em seguida, você usa CodeDeploy para implantar a revisão. Para instruções, consulte Crie uma implantação com CodeDeploy.

O appspec.yml para a implantação de uma plataforma de EC2/On-Premises computação é salvo no diretório raiz da sua revisão. Para obter mais informações, consulte Adicionar um AppSpec arquivo para uma EC2/On-Premises implantação e Planeje uma revisão para CodeDeploy.

AppSpec Estrutura do arquivo

A seguir está a estrutura de alto nível de um AppSpec arquivo usado para implantações em plataformas AWS Lambda e de computação. EC2/On-Premises

Um valor em um YAML-formatted AppSpec arquivo que seja uma string não deve ser colocado entre aspas (“”), a menos que seja especificado de outra forma.

AppSpec estrutura de arquivos para implantações do Amazon ECS

> **Note**
>
> Esse AppSpec arquivo é escrito em YAML, mas você pode usar a mesma estrutura para escrever um em JSON. Uma string em um JSON-formatted AppSpec arquivo está sempre entre aspas (“”).

```
version: 0.0
resources:
  ecs-service-specifications
hooks:
  deployment-lifecycle-event-mappings
```

Nessa estrutura:

version

Esta seção especifica a versão do AppSpec arquivo. Não mude esse valor. Ele é necessário. Atualmente, o único valor permitido é **0.0**. É reservado CodeDeploy para uso futuro.

Especifique a versão com uma string.

recursos

Esta seção especifica informações sobre o aplicativo do Amazon ECS que será implantado.

Para obter mais informações, consulte AppSpec seção 'recursos' para implantações do Amazon ECS.

hooks

Esta seção especifica as funções do Lambda que serão executadas em hooks de eventos específicos de ciclo de vida de implantação para validar a implantação.

Para obter mais informações, consulte Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS.

AppSpec estrutura de arquivos para AWS Implantações do Lambda

> **Note**
>
> Esse AppSpec arquivo é escrito em YAML, mas você pode usar a mesma estrutura para gravar um AppSpec arquivo para uma implantação do Lambda em JSON. Uma string em um JSON-formatted AppSpec arquivo está sempre entre aspas (“”).

```
version: 0.0
resources:
  lambda-function-specifications
```

```
hooks:
  deployment-lifecycle-event-mappings
```

Nessa estrutura:

version

Esta seção especifica a versão do AppSpec arquivo. Não mude esse valor. Ele é necessário. Atualmente, o único valor permitido é **`0.0`**. É reservado CodeDeploy para uso futuro.

Especifique a versão com uma string.

recursos

Esta seção especifica informações sobre a função do Lambda a ser implantada.

Para obter mais informações, consulte AppSpec seção 'recursos' (Amazon ECS e AWS Lambda (somente implantações).

hooks

Esta seção especifica as funções do Lambda a serem executadas em eventos específicos de ciclo de vida de implantação para validar a implantação.

Para obter mais informações, consulte AppSpec seção 'ganchos'.

AppSpec estrutura de arquivos para EC2/On-Premises implantações

```
version: 0.0
os: operating-system-name
files:
  source-destination-files-mappings
permissions:
  permissions-specifications
hooks:
  deployment-lifecycle-event-mappings
```

Nessa estrutura:

version

Esta seção especifica a versão do AppSpec arquivo. Não mude esse valor. Ele é necessário. Atualmente, o único valor permitido é **`0.0`**. É reservado CodeDeploy para uso futuro.

Especifique a versão com uma string.

os

Esta seção especifica o valor do sistema operacional da instância na qual você fará a implantação. Ele é necessário. Os seguintes valores podem ser especificados:

- linux — A instância é uma instância Amazon Linux, Ubuntu Server, RHEL ou macOS.
- windows: A instância é uma instância do Windows Server.

Especifique o SO com uma string.

files

Esta seção especifica o nome dos arquivos que devem ser copiados para a instância durante o evento Install da implantação.

Para obter mais informações, consulte AppSpec seção 'arquivos' (somente EC2/On-Premises implantações).

permissions

Esta seção especifica como permissões especiais, se houver, devem ser aplicadas aos arquivos na seção `files` à medida que estes são copiados para a instância. Esta seção se aplica somente às instâncias do Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) e macOS.

Para obter mais informações, consulte, AppSpec seção 'permissões' (somente EC2/On-Premises implantações).

hooks

Esta seção especifica scripts a serem executados em eventos específicos de ciclo de vida de implantação durante a implantação.

Para obter mais informações, consulte AppSpec seção 'ganchos'.

Tópicos

- AppSpec seção 'arquivos' (somente EC2/On-Premises implantações)
- AppSpec seção 'recursos' (Amazon ECS e AWS Lambda (somente implantações)
- AppSpec seção 'permissões' (somente EC2/On-Premises implantações)
- AppSpec seção 'ganchos'

AppSpec seção 'arquivos' (somente EC2/On-Premises implantações)

Fornece informações CodeDeploy sobre quais arquivos da revisão do seu aplicativo devem ser instalados na instância durante o evento de instalação da implantação. Esta seção é necessária apenas quando você copia arquivos da sua revisão em locais na instância durante a implantação.

Esta seção tem a seguinte estrutura:

```
files:
  - source: source-file-location-1
    destination: destination-file-location-1
file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

Vários pares `destination` e `source` podem ser definidos.

A instrução `source` identifica um arquivo ou diretório de sua revisão para copiar na instância:

- Se a `source` se referir a um arquivo, somente os arquivos especificados serão copiados na instância.
- Se a `source` se referir a um diretório, todos os arquivos no diretório serão copiados na instância.
- Se `source` for uma única barra ("/" para as instâncias do Amazon Linux, RHEL e Ubuntu Server, ou "\" para instâncias do Windows Server), todos os arquivos da sua revisão serão copiados na instância.

Os caminhos usados no `source` são relativos ao arquivo `appspec.yml`, que deve estar na raiz da sua revisão. Para obter detalhes sobre a estrutura de arquivos de uma revisão, consulte Planeje uma revisão para CodeDeploy.

A instrução `destination` identifica a localização na instância em que os arquivos devem ser copiados. Esse deve ser um caminho totalmente qualificado, como `/root/destination/directory` (no Linux, RHEL e Ubuntu) ou `c:\destination\folder` (no Windows).

`source` e `destination` são especificados com uma string.

A `file_exists_behavior` instrução é opcional e especifica como CodeDeploy manipula arquivos que já existem em um local de destino de implantação, mas não faziam parte da implantação anterior bem-sucedida. Essa configuração ser um dos valores a seguir:

- DISALLOW: a implantação falha. Esse também é o comportamento padrão se nenhuma opção for especificada.
- OVERWRITE: a versão do arquivo da revisão de aplicativo substitui a versão sendo implantada no momento que já está na instância.
- RETAIN: A versão do arquivo que já está na instância é mantida e usada como parte da nova implantação.

Ao usar a configuração `file_exists_behavior`, entenda que essa configuração:

- só pode ser especificada uma vez e se aplica a todos os arquivos e diretórios listados sob `files:`.
- tem precedência sobre a `--file-exists-behavior` AWS CLI opção e a opção `fileExistsBehavior` API (ambas também são opcionais).

Aqui está uma seção `files` de exemplo para uma instância Amazon Linux, Ubuntu Server ou RHEL.

```
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
```

Neste exemplo, as duas operações a seguir são realizadas durante o evento Install:

1. Copie o arquivo `Config/config.txt` em sua revisão no caminho `/webapps/Config/config.txt` na instância.
2. Copie recursivamente todos os arquivos no diretório `source` da revisão para o diretório `/webapps/myApp` na instância.

Exemplos da seção “files”

Os exemplos a seguir mostram como especificar a seção `files`. Embora esses exemplos descrevam estruturas de arquivos e diretórios (pasta) do Windows Server, eles podem ser facilmente adaptados para instâncias do Amazon Linux, Ubuntu Server e RHEL.

> **Note**
>
> Somente EC2/On-Premises implantações usam a `files` seção. Ela não se aplica às implantações do AWS Lambda.

Para os exemplos a seguir, supomos que esses arquivos apareçam no pacote na raiz de `source`:

- `appspec.yml`
- `my-file.txt`
- `my-file-2.txt`
- `my-file-3.txt`

```
# 1) Copy only my-file.txt to the destination folder c:\temp.
#
files:
  - source: .\my-file.txt
    destination: c:\temp
#
# Result:
#   c:\temp\my-file.txt
#
# ---------------------
#
# 2) Copy only my-file-2.txt and my-file-3.txt to the destination folder c:\temp.
#
files:
  - source: my-file-2.txt
    destination: c:\temp
  - source: my-file-3.txt
    destination: c:\temp
#
# Result:
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
#
# ---------------------
#
# 3) Copy my-file.txt, my-file-2.txt, and my-file-3.txt (along with the appspec.yml
 file) to the destination folder c:\temp.
```

```
#
files:
  - source: \
    destination: c:\temp
#
# Result:
#   c:\temp\appspec.yml
#   c:\temp\my-file.txt
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
```

Para os seguintes exemplos, supomos que `appspec.yml` apareça no pacote na raiz de `source`, juntamente com uma pasta chamada `my-folder` que contém três arquivos:

- `appspec.yml`
- `my-folder\my-file.txt`
- `my-folder\my-file-2.txt`
- `my-folder\my-file-3.txt`

```
# 4) Copy the 3 files in my-folder (but do not copy my-folder itself) to the
 destination folder c:\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp
#
# Result:
#   c:\temp\my-file.txt
#   c:\temp\my-file-2.txt
#   c:\temp\my-file-3.txt
#
# ---------------------
#
# 5) Copy my-folder and its 3 files to my-folder within the destination folder c:\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp\my-folder
#
# Result:
#   c:\temp\my-folder\my-file.txt
```

```
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
#
# ---------------------
#
# 6) Copy the 3 files in my-folder to other-folder within the destination folder c:
\temp.
#
files:
  - source: .\my-folder
    destination: c:\temp\other-folder
#
# Result:
#   c:\temp\other-folder\my-file.txt
#   c:\temp\other-folder\my-file-2.txt
#   c:\temp\other-folder\my-file-3.txt
#
# ---------------------
#
# 7) Copy only my-file-2.txt and my-file-3.txt to my-folder within the destination
 folder c:\temp.
#
files:
  - source: .\my-folder\my-file-2.txt
    destination: c:\temp\my-folder
  - source: .\my-folder\my-file-3.txt
    destination: c:\temp\my-folder
#
# Result:
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
#
# ---------------------
#
# 8) Copy only my-file-2.txt and my-file-3.txt to other-folder within the destination
 folder c:\temp.
#
files:
  - source: .\my-folder\my-file-2.txt
    destination: c:\temp\other-folder
  - source: .\my-folder\my-file-3.txt
    destination: c:\temp\other-folder
#
# Result:
```

```
#   c:\temp\other-folder\my-file-2.txt
#   c:\temp\other-folder\my-file-3.txt
#
# --------------------
#
# 9) Copy my-folder and its 3 files (along with the appspec.yml file) to the
 destination folder c:\temp. If any of the files already exist on the instance,
 overwrite them.
#
files:
  - source: \
    destination: c:\temp
file_exists_behavior: OVERWRITE
#
# Result:
#   c:\temp\appspec.yml
#   c:\temp\my-folder\my-file.txt
#   c:\temp\my-folder\my-file-2.txt
#   c:\temp\my-folder\my-file-3.txt
```

AppSpec seção 'recursos' (Amazon ECS e AWS Lambda (somente implantações)

O conteúdo na `'resources'` seção do AppSpec arquivo varia, dependendo da plataforma computacional da sua implantação. A seção `'resources'` para uma implantação do Amazon ECS contém a definição de tarefa do Amazon ECS, o contêiner e a porta para rotear o tráfego para o seu conjunto de tarefas do Amazon ECS atualizado e outras informações opcionais. A `'resources'` seção de uma AWS Lambda implantação contém o nome, o alias, a versão atual e a versão de destino de uma função Lambda.

Tópicos

- AppSpec seção 'recursos' para AWS Implantações do Lambda
- AppSpec seção 'recursos' para implantações do Amazon ECS

AppSpec seção 'recursos' para AWS Implantações do Lambda

A seção `'resources'` especifica a função do Lambda que será implantada e tem a seguinte estrutura:

YAML:

```
resources:
```

```
  - name-of-function-to-deploy:
      type: "AWS::Lambda::Function"
      properties:
        name: name-of-lambda-function-to-deploy
        alias: alias-of-lambda-function-to-deploy
        currentversion: version-of-the-lambda-function-traffic-currently-points-to
        targetversion: version-of-the-lambda-function-to-shift-traffic-to
```

JSON:

```
"resources": [
    {
        "name-of-function-to-deploy" {
            "type": "AWS::Lambda::Function",
            "properties": {
                "name": "name-of-lambda-function-to-deploy",
                "alias": "alias-of-lambda-function-to-deploy",
                "currentversion": "version-of-the-lambda-function-traffic-currently-
points-to",
                "targetversion": "version-of-the-lambda-function-to-shift-traffic-to"
            }
        }
    }
]
```

Cada propriedade é especificada com uma string.

- `name` – obrigatório. Esse é o nome da função do Lambda a ser implantada.
- `alias` – obrigatório. Esse é o nome do alias da função do Lambda.
- `currentversion` – obrigatório. Essa é a versão para a qual o tráfego da função do Lambda aponta. Este valor deve ser um inteiro positivo válido.
- `targetversion` – obrigatório. Essa é a versão para a qual o tráfego da função do Lambda é deslocado. Este valor deve ser um inteiro positivo válido.

AppSpec seção 'recursos' para implantações do Amazon ECS

A seção `'resources'` especifica o serviço do Amazon ECS que será implantado e tem a seguinte estrutura:

YAML:

```
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "task-definition-arn"
        LoadBalancerInfo:
          ContainerName: "ecs-container-name"
          ContainerPort: "ecs-application-port"
# Optional properties
        PlatformVersion: "ecs-service-platform-version"
        NetworkConfiguration:
          AwsvpcConfiguration:
            Subnets: ["ecs-subnet-1","ecs-subnet-n"]
            SecurityGroups: ["ecs-security-group-1","ecs-security-group-n"]
            AssignPublicIp: "ENABLED | DISABLED"
        CapacityProviderStrategy:
          - Base: integer
            CapacityProvider: "capacityProviderA"
            Weight: integer
          - Base: integer
            CapacityProvider: "capacityProviderB"
            Weight: integer
```

JSON:

```
"Resources": [
    {
        "TargetService": {
            "Type": "AWS::ECS::Service",
            "Properties": {
                "TaskDefinition": "task-definition-arn",
                "LoadBalancerInfo": {
                    "ContainerName": "ecs-container-name",
                    "ContainerPort": "ecs-application-port"
                },
                "PlatformVersion": "ecs-service-platform-version",
                "NetworkConfiguration": {
                    "AwsvpcConfiguration": {
                        "Subnets": [
                            "ecs-subnet-1",
                            "ecs-subnet-n"
                        ],
                        "SecurityGroups": [
```

```
                        "ecs-security-group-1",
                        "ecs-security-group-n"
                    ],
                    "AssignPublicIp": "ENABLED | DISABLED"
                }
            },
            "CapacityProviderStrategy": [
                {
                    "Base": integer,
                    "CapacityProvider": "capacityProviderA",
                    "Weight": integer
                },
                {
                    "Base": integer,
                    "CapacityProvider": "capacityProviderB",
                    "Weight": integer
                }
            ]
        }
    }
]
```

Cada propriedade é especificada com uma string, exceto `ContainerPort`, que é um número.

- `TaskDefinition` – obrigatório. Esta é a definição de tarefa para o serviço do Amazon ECS que será implantado. Ela é especificada com o ARN da definição de tarefa. O formato do ARN é `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Para obter mais informações, consulte Amazon Resource Names (ARNs) e namespaces AWS de serviços.

  > **Note**
  >
  > A porção `:`*`task-definition-revision`* do ARN é opcional. Se for omitido, o Amazon ECS usa a última revisão ATIVA da definição da tarefa.

- `ContainerName` – obrigatório. Esse é o nome do contêiner do Amazon ECS que contém seu aplicativo do Amazon ECS. Ele deve ser um contêiner especificado em sua definição de tarefa do Amazon ECS.
- `ContainerPort` – obrigatório. Essa é a porta do contêiner para a qual o tráfego será roteado.

- `PlatformVersion`: opcional. A versão da plataforma das tarefas do Fargate no serviço do Amazon ECS implantado. Para obter mais informações, consulte Versões da plataforma do AWS Fargate. Se não for especificado, `LATEST` é usado como padrão.
- `NetworkConfiguration`: opcional. Em `AwsvpcConfiguration`, especifique o seguinte: Para obter mais informações, consulte AwsVpcConfigurationa Referência da API do Amazon ECS Container Service.
  - `Subnets`: opcional. Uma lista separada por vírgulas de uma ou mais sub-redes em seu serviço do Amazon ECS.
  - `SecurityGroups`: opcional. Uma lista separada por vírgulas de um ou mais grupos de segurança em seu serviço do Amazon Elastic Container Service.
  - `AssignPublicIp`: opcional. Uma string que especifica se a interface de rede elástica do serviço do Amazon ECS recebe um endereço IP público. Os valores válidos são `ENABLED` e `DISABLED`.

  > **Note**
  >
  > Todas ou nenhuma das configurações em `NetworkConfiguration` devem ser especificadas. Por exemplo, se você deseja especificar `Subnets`, também deverá especificar `SecurityGroups` e `AssignPublicIp`. Se nenhuma for especificada, CodeDeploy usa as configurações atuais da rede Amazon ECS.

- `CapacityProviderStrategy`: opcional. Uma lista dos provedores de capacidade do Amazon ECS que você deseja usar para sua implantação. Para obter mais informações, consulte Provedores de capacidade do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service. Para cada provedor de capacidade, é possível especificar as configurações a seguir. Para obter detalhes sobre essas configurações, consulte AWS: :ECS:: ServiceCapacityProviderStrategyItem no Guia do usuário AWS CloudFormation
  - `Base`: opcional. O valor da base designa o número mínimo de tarefas que serão executadas no provedor de capacidade especificado. Somente um provedor de capacidade em uma estratégia de provedor de capacidade pode ter uma base definida. Se nenhum valor for especificado, será usado o valor padrão 0.
  - `CapacityProvider`: opcional. O nome curto do provedor de capacidade. Exemplo: capacityProviderA
  - `Weight`: opcional.

O valor do peso designa a porcentagem relativa do número total de tarefas executadas que devem usar o provedor de capacidade especificado. O valor `weight` é levado em consideração depois que o valor `base`, se definido, for satisfeito.

Se nenhum valor de `weight` for especificado, será usado o valor padrão de `0`. Quando vários fornecedores de capacidade são especificados dentro de uma estratégia de provedor de capacidade, pelo menos um dos provedores de capacidade deve ter um valor ponderal superior a zero e quaisquer prestadores de capacidade com um peso de não `0` serão utilizados para realizar tarefas. Se você especificar vários provedores de capacidade em uma estratégia em que todos tenham um peso de `0`, quaisquer ações `CreateService` ou `RunTask` que usarem a estratégia de provedor de capacidade apresentarão falha.

Um exemplo de situação para o uso de pesos é definir uma estratégia que contenha dois provedores de capacidade e ambos tiverem um peso de 1. Quando o `base` for atendido, as tarefas serão divididas uniformemente entre os dois provedores de capacidade. Usando essa mesma lógica, se você especificar um peso de 1 para CapacityProviderA e um peso de 4 para CapacityProviderB, então, para cada tarefa executada que usar o CapacityProviderA, quatro tarefas usariam o CapacityProviderB.

AppSpec seção 'permissões' (somente EC2/On-Premises implantações)

A `'permissions'` seção especifica como as permissões especiais, se houver, devem ser aplicadas aos arquivos e directories/folders na `'files'` seção após serem copiadas para a instância. É possível especificar várias instruções `object`. Esta seção é opcional. Ela se aplica somente às instâncias Amazon Linux, Ubuntu Server, RHEL e macOS.

> **Note**
>
> Nas instâncias do macOS, as `context` instruções `acls` e não são suportadas. Uma implantação cujo AppSpec arquivo usa qualquer um deles no macOS falha antes que qualquer arquivo seja instalado. As outras instruções funcionam conforme descrito nesta seção.

> **Note**
>
> A seção `'permissions'` é usada somente para implantações EC2/On-Premises. Ele não é usado para implantações do AWS Lambda ou do Amazon ECS.

Esta seção tem a seguinte estrutura:

```
permissions:
  - object: object-specification
    pattern: pattern-specification
    except: exception-specification
    owner: owner-account-name
    group: group-name
    mode: mode-specification
    acls:
      - acls-specification
    context:
      user: user-specification
      type: type-specification
      range: range-specification
    type:
      - object-type
```

As instruções são as seguintes:

- `object` – obrigatório. Esse é um conjunto de objetos do sistema de arquivos (arquivos ou directories/folders) aos quais as permissões especificadas são aplicadas depois que os objetos do sistema de arquivos são copiados para a instância.

  Especifique `object` com uma string.

- `pattern`: opcional. Especifica um padrão para aplicar permissões. Se não forem especificadas ou se forem especificadas com os caracteres especiais **"**"**, as permissões serão aplicadas em todos os arquivos ou diretórios correspondentes, de acordo com o `type`.

  Especifique `pattern` com uma string entre aspas ("").

- `except`: opcional. Especifica os arquivos ou diretórios que são exceções ao `pattern`.

  Especifique `except` com uma lista de strings separada por vírgulas entre colchetes.

- `owner`: opcional. O nome do proprietário do `object`. Se não for especificado, todos os proprietários existentes aplicados ao arquivo ou directory/folder estrutura original permanecerão inalterados após a operação de cópia.

  Especifique `owner` com uma string.

- `group`: opcional. O nome do grupo para `object`. Se não for especificado, todos os grupos existentes aplicados ao arquivo ou directory/folder estrutura original permanecerão inalterados após a operação de cópia.

  Especifique `group` com uma string.

- `mode`: opcional. Um valor numérico que especifica as permissões a serem aplicadas ao `object`. A configuração do modo segue a sintaxe do comando chmod do Linux.

  > **Important**
  >
  > Se o valor incluir um zero inicial, você deverá colocá-lo entre aspas duplas ou remover o zero inicial para que restem apenas três dígitos.

  > **Note**
  >
  > Notação simbólica, como **u+x**, não é suportada para a configuração `mode`.

  Exemplos:

  - `mode: "0644"` concede permissões de leitura e gravação ao proprietário do objeto (6), permissões somente leitura ao grupo (4) e permissões somente leitura a todos os outros usuários (4).
  - `mode: 644` concede as mesmas permissões que `mode: "0644"`.
  - `mode: 4755` define o atributo setuid (4), concede permissões de controle total ao proprietário (7), concede permissões de leitura e execução ao grupo (5) e concede permissões de leitura e execução a todos os outros usuários (5).

    Para obter mais exemplos, consulte a documentação do comando chmod do Linux.

    Se o modo não for especificado, todos os modos existentes aplicados à estrutura original de arquivos ou pastas permanecerão inalterados após a operação de cópia.

- `acls`: opcional. Uma lista de strings de caracteres que representam uma ou mais entradas da lista de controle de acesso (ACL) aplicadas ao `object`. Por exemplo, **`u:bob:rw`** representa permissões de leitura e gravação para o usuário **bob**. (Para obter mais exemplos, consulte exemplos de formatos de entradas de ACL na documentação do comando `setfacl` do Linux). É possível especificar várias entradas de ACL. Se não `acls` for especificado, todas as ACLs existentes aplicadas ao arquivo ou directory/folder estrutura original permanecerão inalteradas após a operação de cópia. Elas substituirão as ACLs existentes.

  Especifique uma `acls` com um traço (-), seguido por um espaço e uma string (por exemplo, `- u:jane:rw`). Se você tiver mais de uma ACL, cada uma será especificada em uma linha separada.

  > **Note**
  >
  > A configuração de usuários sem nome, grupos sem nome ou outras entradas de ACL semelhantes faz com que o AppSpec arquivo falhe. Use `mode` para especificar esses tipos de permissões em vez disso.

- `context`: opcional. Para instâncias habilitadas para Security-Enhanced Linux (SELinux), uma lista de rótulos de contexto relevantes para segurança a serem aplicados aos objetos copiados. Os rótulos são especificados como chaves contendo `user`, `type`, e `range`. (Para obter mais informações, consulte a documentação do SELinux). Cada chave é inserida com uma string. Se não for especificado, todos os rótulos existentes aplicados ao arquivo ou directory/folder estrutura original permanecerão inalterados após a operação de cópia.
  - `user`: opcional. O usuário do SELinux.
  - `type`: opcional. O nome do tipo SELinux.
  - `range`: opcional. O especificador do intervalo SELinux. Isso não tem efeito, a menos que Multi-Level Segurança (MLS) e Multi-Category Segurança (MCS) estejam habilitadas na máquina. Se não estiver habilitado, `range` assumirá **`s0`** como padrão.

  Especifique `context` com uma string (por exemplo, `user: unconfined_u`). Cada `context` é especificado em uma linha separada.

- `type`: opcional. Os tipos de objetos aos quais aplicar as permissões especificadas. `type` é uma string que pode ser definida como **`file`** ou **`directory`**. Se **`file`** for especificado, as permissões serão aplicadas somente nos arquivos que estiverem imediatamente contidos no `object` após a operação de cópia (e não no `object` em si). Se **`directory`** for especificado, as permissões

serão aplicadas recursivamente a tudo o directories/folders que estiver em qualquer lugar `object` após a operação de cópia (mas não a `object` si mesma).

Especifique o `type` com um traço (-), seguido por um espaço e uma string (por exemplo, `- file`).

Exemplo da seção 'permissions'

O exemplo a seguir mostra como especificar a seção `'permissions'` com as instruções `object`, `pattern`, `except`, `owner`, `mode` e `type`. Esse exemplo se aplica somente às instâncias Amazon Linux, Ubuntu Server e RHEL. Neste exemplo, suponha que os seguintes arquivos e pastas sejam copiados para a instância nesta hierarquia:

```
/tmp
  `-- my-app
       |-- my-file-1.txt
       |-- my-file-2.txt
       |-- my-file-3.txt
       |-- my-folder-1
       |     |-- my-file-4.txt
       |     |-- my-file-5.txt
       |     `-- my-file-6.txt
       `-- my-folder-2
             |-- my-file-7.txt
             |-- my-file-8.txt
             |-- my-file-9.txt
            `-- my-folder-3
```

O AppSpec arquivo a seguir mostra como definir permissões nesses arquivos e pastas após serem copiados:

```
version: 0.0
os: linux
# Copy over all of the folders and files with the permissions they
#  were originally assigned.
files:
  - source: ./my-file-1.txt
    destination: /tmp/my-app
  - source: ./my-file-2.txt
    destination: /tmp/my-app
  - source: ./my-file-3.txt
    destination: /tmp/my-app
```

```
  - source: ./my-folder-1
    destination: /tmp/my-app/my-folder-1
  - source: ./my-folder-2
    destination: /tmp/my-app/my-folder-2
# 1) For all of the files in the /tmp/my-app folder ending in -3.txt
#  (for example, just my-file-3.txt), owner = adm, group = wheel, and
#  mode = 464 (-r--rw-r--).
permissions:
  - object: /tmp/my-app
    pattern: "*-3.txt"
    owner: adm
    group: wheel
    mode: 464
    type:
      - file
# 2) For all of the files ending in .txt in the /tmp/my-app
#  folder, but not for the file my-file-3.txt (for example,
#  just my-file-1.txt and my-file-2.txt),
#  owner = ec2-user and mode = 444 (-r--r--r--).
  - object: /tmp/my-app
    pattern: "*.txt"
    except: [my-file-3.txt]
    owner: ec2-user
    mode: 444
    type:
      - file
# 3) For all the files in the /tmp/my-app/my-folder-1 folder except
#  for my-file-4.txt and my-file-5.txt, (for example,
#  just my-file-6.txt), owner = operator and mode = 646 (-rw-r--rw-).
  - object: /tmp/my-app/my-folder-1
    pattern: "**"
    except: [my-file-4.txt, my-file-5.txt]
    owner: operator
    mode: 646
    type:
      - file
# 4) For all of the files that are immediately under
#  the /tmp/my-app/my-folder-2 folder except for my-file-8.txt,
#  (for example, just my-file-7.txt and
#  my-file-9.txt), owner = ec2-user and mode = 777 (-rwxrwxrwx).
  - object: /tmp/my-app/my-folder-2
    pattern: "**"
    except: [my-file-8.txt]
    owner: ec2-user
```

```
    mode: 777
    type:
      - file
# 5) For all folders at any level under /tmp/my-app that contain
#  the name my-folder but not
#  /tmp/my-app/my-folder-2/my-folder-3 (for example, just
#  /tmp/my-app/my-folder-1 and /tmp/my-app/my-folder-2),
#  owner = ec2-user and mode = 555 (dr-xr-xr-x).
  - object: /tmp/my-app
    pattern: "*my-folder*"
    except: [tmp/my-app/my-folder-2/my-folder-3]
    owner: ec2-user
    mode: 555
    type:
      - directory
# 6) For the folder /tmp/my-app/my-folder-2/my-folder-3,
#  group = wheel and mode = 564 (dr-xrw-r--).
  - object: /tmp/my-app/my-folder-2/my-folder-3
    group: wheel
    mode: 564
    type:
      - directory
```

As permissões resultantes são as seguintes:

```
-r--r--r-- ec2-user root  my-file-1.txt
-r--r--r-- ec2-user root  my-file-2.txt
-r--rw-r-- adm      wheel my-file-3.txt

dr-xr-xr-x ec2-user root  my-folder-1
-rw-r--r-- root     root  my-file-4.txt
-rw-r--r-- root     root  my-file-5.txt
-rw-r--rw- operator root  my-file-6.txt

dr-xr-xr-x ec2-user root  my-folder-2
-rwxrwxrwx ec2-user root  my-file-7.txt
-rw-r--r-- root     root  my-file-8.txt
-rwxrwxrwx ec2-user root  my-file-9.txt

dr-xrw-r-- root     wheel my-folder-3
```

O exemplo a seguir mostra como especificar a seção `'permissions'` com a adição das instruções `acls` e `context`. Esse exemplo se aplica somente às instâncias Amazon Linux, Ubuntu Server e RHEL.

```
permissions:
  - object: /var/www/html/WordPress
    pattern: "**"
    except: [/var/www/html/WordPress/ReadMe.txt]
    owner: bob
    group: writers
    mode: 644
    acls:
      - u:mary:rw
      - u:sam:rw
      - m::rw
    context:
      user: unconfined_u
      type: httpd_sys_content_t
      range: s0
    type:
      - file
```

AppSpec seção 'ganchos'

O conteúdo na `'hooks'` seção do AppSpec arquivo varia, dependendo da plataforma de computação para sua implantação. A seção `'hooks'` para uma implantação da EC2/On-Premises contém mapeamentos que vinculam os ganchos de evento do ciclo de vida de implantação a um ou mais scripts. A seção `'hooks'` para uma implantação do Lambda ou Amazon ECS especifica as funções de validação Lambda que serão executadas durante um evento de ciclo de vida de implantação. Se um gancho de evento não estiver presente, nenhuma operação será executada para esse evento. Esta seção é necessária somente se você está executando scripts ou funções de validação do Lambda como parte da implantação.

Tópicos

- AppSpec seção 'hooks' para uma implantação do Amazon ECS
- AppSpec seção 'ganchos' para um AWS implantação do Lambda
- AppSpec seção 'ganchos' para uma implantação EC2/On-Premises

AppSpec seção 'hooks' para uma implantação do Amazon ECS

Tópicos

- Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS
- Ordem de execução de hooks em uma implantação do Amazon ECS.
- Estrutura da seção 'hooks'
- Exemplo da função do Lambda 'hooks'

Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS

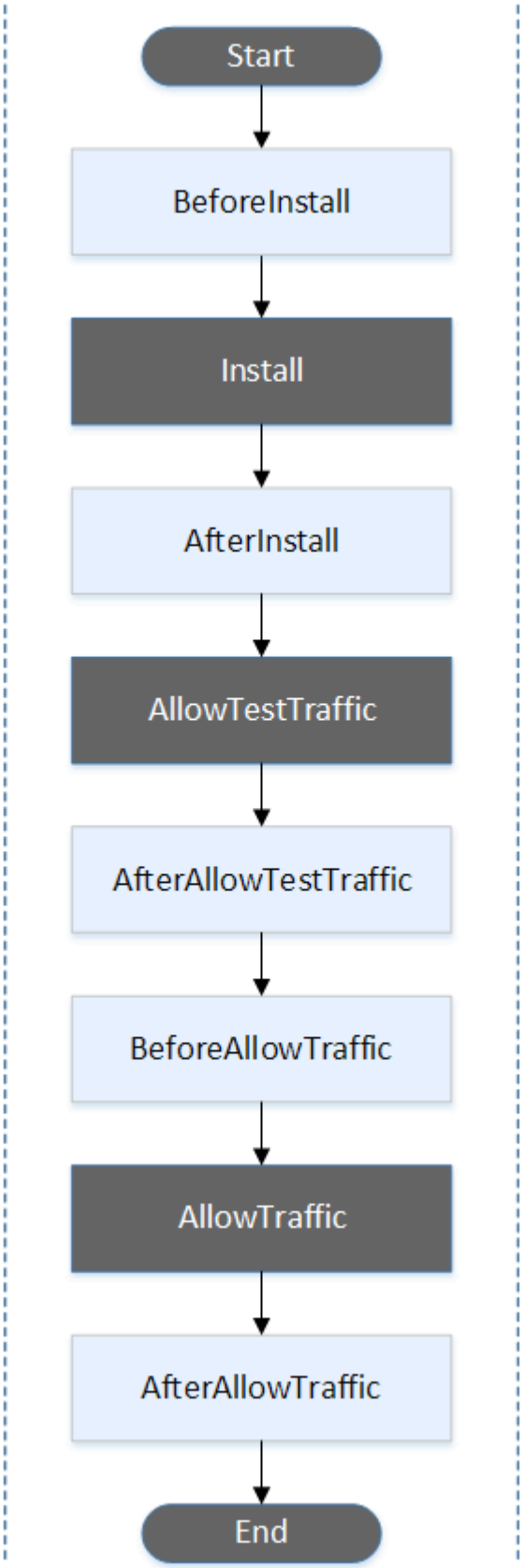
Um gancho AWS Lambda é uma função Lambda especificada com uma string em uma nova linha após o nome do evento do ciclo de vida. Cada gancho é executado uma vez por implantação. Veja a seguir as descrições dos eventos de ciclo de vida em que você pode executar um hook durante uma implantação do Amazon ECS.

- `BeforeInstall`: use para executar tarefas antes que o conjunto de tarefas de substituição seja criado. Um grupo de destino é associado ao conjunto de tarefas original. Se um listener de teste opcional for especificado, ele será associado ao conjunto de tarefas original. Não é possível fazer uma reversão nesse momento.
- `AfterInstall`: use para executar tarefas depois que o conjunto de tarefas de substituição for criado e um dos grupos de destino for associado a ele. Se um listener de teste opcional for especificado, ele será associado ao conjunto de tarefas original. Os resultados de uma função de gancho nesse evento de ciclo de vida podem acionar uma reversão.
- `AfterAllowTestTraffic`: use para executar tarefas depois que o receptor de teste distribuir o tráfego para o conjunto de tarefas de substituição. Os resultados de uma função de gancho, nesse momento, podem acionar uma reversão.
- `BeforeAllowTraffic`: use 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. Os resultados de uma função de gancho nesse evento de ciclo de vida podem acionar uma reversão.
- `AfterAllowTraffic`: use para executar tarefas depois que o segundo grupo de destino distribuir o tráfego para o conjunto de tarefas de substituição. Os resultados de uma função de gancho nesse evento de ciclo de vida podem acionar uma reversão.

Para obter mais informações, consulte O que acontece durante uma implantação do e Tutorial: Implantar um serviço do Amazon ECS com um teste de validação.

Ordem de execução de hooks em uma implantação do Amazon ECS.

Em uma implantação do Amazon ECS, hooks de eventos são executados na seguinte ordem:


Start
BeforeInstall
Install
AfterInstall
AllowTestTraffic
AfterAllowTestTraffic
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
End

> **Note**
>
> Os eventos Start, Install TestTraffic, AllowTraffic, e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama.

Estrutura da seção 'hooks'

Os seguintes são exemplos da estrutura da seção `'hooks'`.

Uso de YAML:

```
Hooks:
  - BeforeInstall: "BeforeInstallHookFunctionName"
  - AfterInstall: "AfterInstallHookFunctionName"
  - AfterAllowTestTraffic: "AfterAllowTestTrafficHookFunctionName"
  - BeforeAllowTraffic: "BeforeAllowTrafficHookFunctionName"
  - AfterAllowTraffic: "AfterAllowTrafficHookFunctionName"
```

Uso de JSON:

```
"Hooks": [
  {
   "BeforeInstall": "BeforeInstallHookFunctionName"
  },
  {
   "AfterInstall": "AfterInstallHookFunctionName"
  },
  {
   "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
  },
  {
   "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
  },
  {
   "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
  }
 ]
}
```

Exemplo da função do Lambda 'hooks'

Use a `'hooks'` seção para especificar uma função Lambda que CodeDeploy pode ser chamada para validar uma implantação do Amazon ECS. É possível usar a mesma função ou uma diferente para os eventos de ciclo de vida de implantação `BeforeInstall`, `AfterInstall`, `AfterAllowTestTraffic`, `BeforeAllowTraffic` e `AfterAllowTraffic`. Após a conclusão dos testes de validação, a `AfterAllowTraffic` função Lambda retorna CodeDeploy e fornece um resultado de `Succeeded` ou. `Failed`

> **Important**
>
> A implantação é considerada falhada se não CodeDeploy for notificada pela função de validação do Lambda em uma hora.

Antes de invocar uma função de hook do Lambda, o servidor deve ser notificado sobre o ID de implantação e o ID de execução do hook de evento de ciclo de vida usando o comando `putLifecycleEventHookExecutionStatus`.

A seguir está um exemplo da função de gancho Lambda escrita em. Node.js

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
    //Read the DeploymentId from the event payload.
    var deploymentId = event.DeploymentId;

    //Read the LifecycleEventHookExecutionId from the event payload
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    /*
     Enter validation tests here.
    */

    // Prepare the validation test results with the deploymentId and
    // the lifecycleEventHookExecutionId for CodeDeploy.
    var params = {
        deploymentId: deploymentId,
```

```
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
        if (err) {
            // Validation failed.
            callback('Validation test failed');
        } else {
            // Validation succeeded.
            callback(null, 'Validation test succeeded');
        }
    });
};
```

AppSpec seção 'ganchos' para um AWS implantação do Lambda

Tópicos

- Lista de ganchos de eventos de ciclo de vida para um AWS implantação do Lambda
- Ordem de execução de hooks em uma implantação da versão da função do Lambda
- Estrutura da seção 'hooks'
- Exemplo da função do Lambda 'hooks'

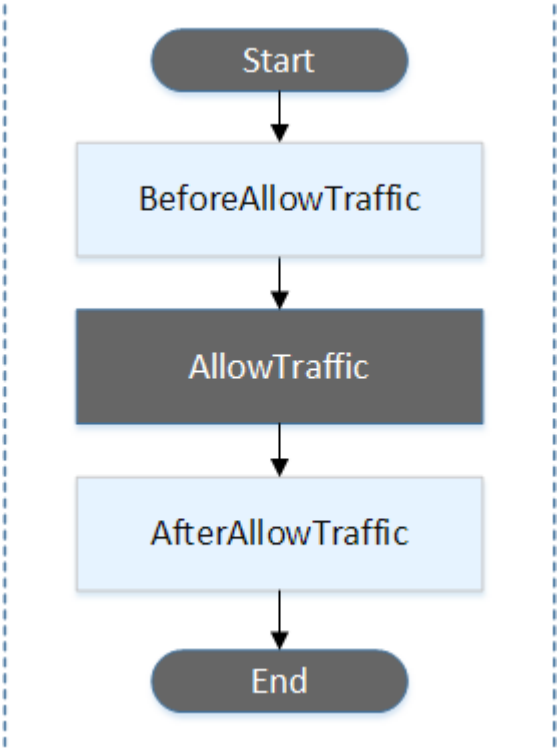
Lista de ganchos de eventos de ciclo de vida para um AWS implantação do Lambda

Um gancho AWS Lambda é uma função Lambda especificada com uma string em uma nova linha após o nome do evento do ciclo de vida. Cada gancho é executado uma vez por implantação. Aqui estão as descrições dos ganchos disponíveis para uso em seu AppSpec arquivo.

- BeforeAllowTraffic— Use para executar tarefas antes que o tráfego seja transferido para a versão implantada da função Lambda.
- AfterAllowTraffic— Use para executar tarefas depois que todo o tráfego for transferido para a versão implantada da função Lambda.

Ordem de execução de hooks em uma implantação da versão da função do Lambda

Em uma implantação da versão da função do Lambda com tecnologia sem servidor, os hooks de eventos são executados na seguinte ordem:



> **Note**
>
> Os eventos Start e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama. AllowTraffic

Estrutura da seção 'hooks'

Os seguintes são exemplos da estrutura da seção 'hooks'.

Uso de YAML:

```
hooks:
   - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
   - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

Uso de JSON:

```
"hooks": [{
    "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
    },
    {
```

```
    "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

Exemplo da função do Lambda 'hooks'

Use a seção 'hooks' para especificar uma função Lambda CodeDeploy que pode ser chamada para validar uma implantação do Lambda. É possível usar a mesma função ou uma diferente para os eventos de ciclo de vida de implantação `BeforeAllowTraffic` e `AfterAllowTraffic`. Após a conclusão dos testes de validação, a função de validação do Lambda retorna CodeDeploy e fornece um resultado de `Succeeded` ou. `Failed`

> **Important**
>
> A implantação é considerada falhada se não CodeDeploy for notificada pela função de validação do Lambda em uma hora.

Antes de invocar uma função de hook do Lambda, o servidor deve ser notificado sobre o ID de implantação e o ID de execução do hook de evento de ciclo de vida usando o comando `putLifecycleEventHookExecutionStatus`.

A seguir está um exemplo da função de gancho Lambda escrita em. Node.js

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
    //Read the DeploymentId from the event payload.
    var deploymentId = event.DeploymentId;

    //Read the LifecycleEventHookExecutionId from the event payload
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

    /*
     Enter validation tests here.
    */

    // Prepare the validation test results with the deploymentId and
    // the lifecycleEventHookExecutionId for CodeDeploy.
```

```
    var params = {
        deploymentId: deploymentId,
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
        if (err) {
            // Validation failed.
            callback('Validation test failed');
        } else {
            // Validation succeeded.
            callback(null, 'Validation test succeeded');
        }
    });
};
```

AppSpec seção 'ganchos' para uma implantação EC2/On-Premises

Tópicos

- Lista de hooks de eventos de ciclo de vida
- Disponibilidade de hooks de eventos de ciclo de vida
- Ordem de execução de hooks em uma implantação
- Estrutura da seção 'hooks'
- Referenciar arquivos em scripts de hook
- Disponibilidade variáveis de ambientes para hooks
- Exemplo de hooks

Lista de hooks de eventos de ciclo de vida

Um gancho de EC2/On-Premises implantação é executado uma vez por implantação em uma instância. Você pode especificar um ou mais scripts a serem executados em um gancho. Cada gancho para um evento de ciclo de vida é especificado com uma string em uma linha separada. Aqui estão as descrições dos ganchos disponíveis para uso em seu AppSpec arquivo.

Para obter informações sobre quais ganchos de eventos de ciclo de vida são válidos para quais tipos de implantação e reversão, consulte Disponibilidade de hooks de eventos de ciclo de vida.

- `ApplicationStop`: esse evento de ciclo de vida de implantação ocorre mesmo antes do download da revisão de aplicativo. É possível especificar scripts para esse evento de forma a interromper normalmente o aplicativo ou remover pacotes atualmente instalados na preparação para uma implantação. O AppSpec arquivo e os scripts usados para esse evento do ciclo de vida da implantação são da revisão anterior do aplicativo implantado com sucesso.

  > **Note**
  >
  > Um AppSpec arquivo não existe em uma instância antes de você implantá-la. Por esse motivo, o gancho `ApplicationStop` não é executado da primeira vez em que você implanta na instância. Você poderá usar o gancho `ApplicationStop` na segunda vez que implantar em uma instância.

  Para determinar o local da última revisão do aplicativo implantada com sucesso, o CodeDeploy agente consulta o local listado no *`deployment-group-id`*`_last_successful_install` arquivo. Esse arquivo está localizado em:

  Pasta `/opt/codedeploy-agent/deployment-root/deployment-instructions` em instâncias do Amazon Linux, do Ubuntu Server e do RHEL do Amazon EC2.

  Pasta `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions` em instâncias do Windows Server Amazon EC2

  Para solucionar um problema de falha na implantação durante o evento de ciclo de vida de implantação `ApplicationStop`, consulte Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação.

- `DownloadBundle`— Durante esse evento do ciclo de vida da implantação, o CodeDeploy agente copia os arquivos de revisão do aplicativo em um local temporário:

  Pasta `/opt/codedeploy-agent/deployment-root/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive` em instâncias do Amazon Linux, do Ubuntu Server e do RHEL do Amazon EC2.

  Pasta `C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\deployment-archive` em instâncias do Windows Server Amazon EC2

  Esse evento é reservado para o CodeDeploy agente e não pode ser usado para executar scripts.

Para solucionar um problema de falha na implantação durante o evento de ciclo de vida de implantação `DownloadBundle`, consulte Solução de problemas em um evento DownloadBundle de ciclo de vida de implantação com falha com UnknownError: não aberto para leitura.

- `BeforeInstall`: use esse evento de ciclo de vida de implantação para tarefas de pré-instalação, como descriptografar arquivos e criar um backup da versão atual.
- `Install`— Durante esse evento do ciclo de vida da implantação, o CodeDeploy agente copia os arquivos de revisão do local temporário para a pasta de destino final. Esse evento é reservado para o CodeDeploy agente e não pode ser usado para executar scripts.
- `AfterInstall`: use esse evento de ciclo de vida de implantação para tarefas como configurar seu aplicativo ou alterar as permissões dos arquivos.
- `ApplicationStart`: normalmente, você usa esse evento de ciclo de vida de implantação para reiniciar os serviços que foram interrompidos durante o `ApplicationStop`.
- `ValidateService`: este é o último evento de ciclo de vida de implantação. Ele é usado para verificar se a implantação foi concluída com êxito.
- `BeforeBlockTraffic`: você pode usar esse evento de ciclo de vida de implantação para executar tarefas em instâncias antes que elas tenham seu registro cancelado em um balanceador de carga.

  Para solucionar um problema de falha na implantação durante o evento de ciclo de vida de implantação `BeforeBlockTraffic`, consulte Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação.
- `BlockTraffic`: durante esse evento de ciclo de vida de implantação, o tráfego da Internet é impedido de acessar as instâncias que atualmente estão distribuindo o tráfego. Esse evento é reservado para o CodeDeploy agente e não pode ser usado para executar scripts.
- `AfterBlockTraffic`: você pode usar esse evento de ciclo de vida de implantação para executar tarefas em instâncias depois que elas tenham seu registro cancelado em seu respectivo balanceador de carga.

  Para solucionar um problema de falha na implantação durante o evento de ciclo de vida de implantação `AfterBlockTraffic`, consulte Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação.
- `BeforeAllowTraffic`: você pode usar esse evento de ciclo de vida de implantação para executar tarefas em instâncias antes que elas sejam registradas em um balanceador de carga.

- `AllowTraffic`: durante esse evento de ciclo de vida de implantação, o tráfego da Internet pode acessar instâncias após uma implantação. Esse evento é reservado para o CodeDeploy agente e não pode ser usado para executar scripts.
- `AfterAllowTraffic`: você pode usar esse evento de ciclo de vida de implantação para executar tarefas em instâncias depois que elas sejam registradas em um balanceador de carga.

Disponibilidade de hooks de eventos de ciclo de vida

A tabela a seguir lista os ganchos de eventos de ciclo de vida disponíveis para cada cenário de implantação e reversão.

Nome do evento de ciclo de vida	Implantação de inicialização do Auto Scaling[1]	Implantação de encerramento do Auto Scaling[1]	In-place implantação[2]	Blue/green implantação: instâncias originais	Blue/green implantação: instâncias de substituição	Blue/green reversão da implantação: instâncias originais	Blue/green reversão de implantação: instâncias de substituição
ApplicationStop	✓	✓	✓		✓		
DownloadBundle[3]	✓		✓		✓		
BeforeInstall	✓		✓		✓		
Instalar[3]	✓		✓		✓		
AfterInstall	✓		✓		✓		
ApplicationStart	✓		✓		✓		

Nome do evento de ciclo de vida	Implantação de inicialização do Auto Scaling[1]	Implantação de encerramento do Auto Scaling[1]	In-place implantação[2]	Blue/green implantação: instâncias originais	Blue/green implantação: instâncias de substituição	Blue/green reversão da implantação: instâncias originais	Blue/green reversão de implantação: instâncias de substituição
ValidateService	✓		✓		✓		
BeforeBlockTraffic		✓	✓	✓			✓
BlockTraffic[3]		✓	✓	✓			✓
AfterBlockTraffic		✓	✓	✓			✓
BeforeAllowTraffic	✓		✓		✓	✓	
AllowTraffic[3]	✓		✓		✓	✓	
AfterAllowTraffic	✓		✓		✓	✓	

[1] Para obter informações sobre as implantações do Amazon EC2 Auto Scaling, consulte Como o Amazon EC2 Auto Scaling funciona com CodeDeploy.

[2] Também se aplica à reversão de uma implantação no local.

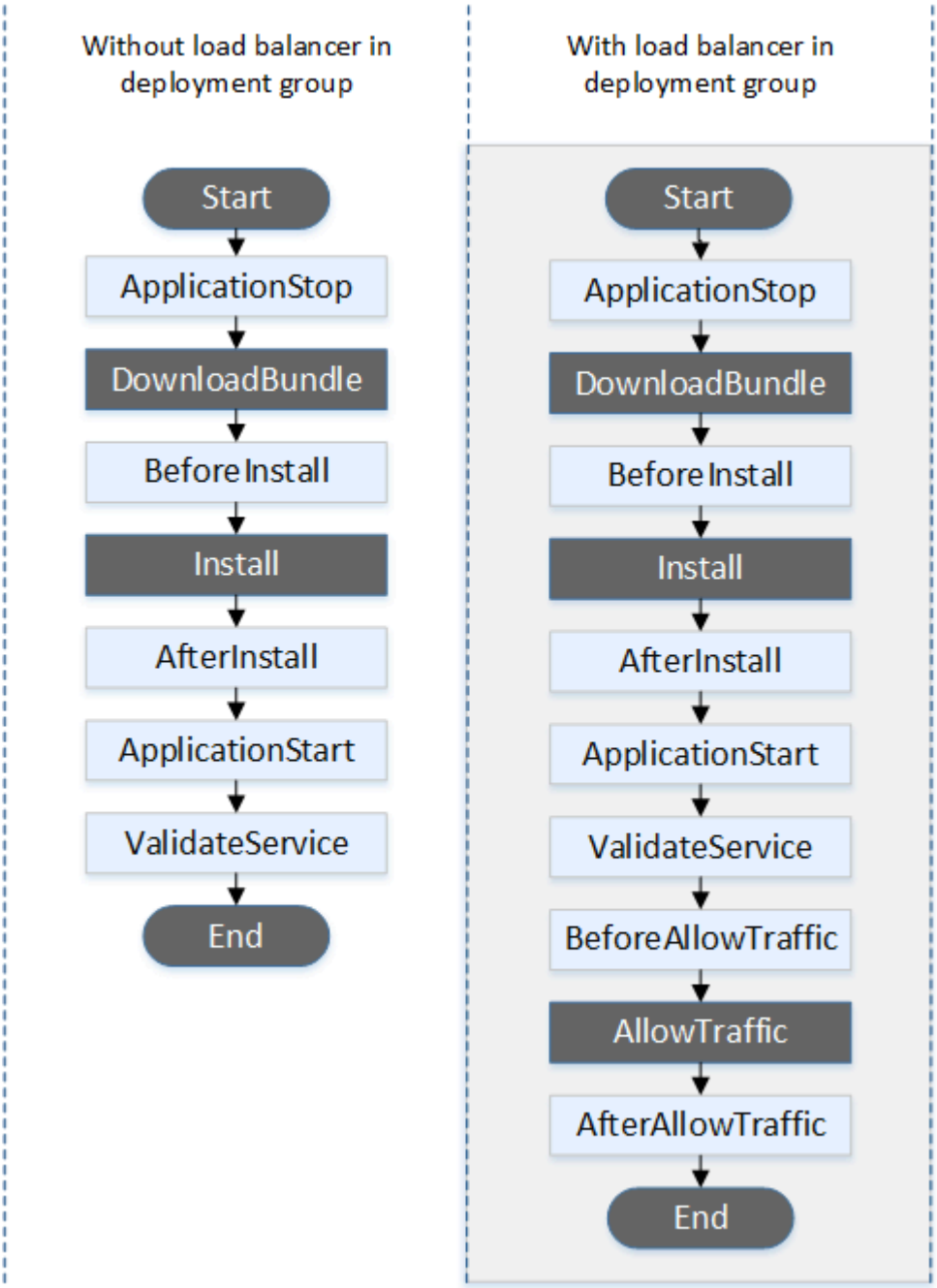
[3] Reservado para CodeDeploy operações. Não pode ser usado para executar scripts.

Ordem de execução de hooks em uma implantação

Implantações de inicialização do Auto Scaling

Durante uma implantação de lançamento do Auto Scaling, CodeDeploy executa ganchos de eventos na seguinte ordem.

Para obter mais informações sobre as implantações de inicialização do Auto Scaling, consulte Como o Amazon EC2 Auto Scaling funciona com CodeDeploy.

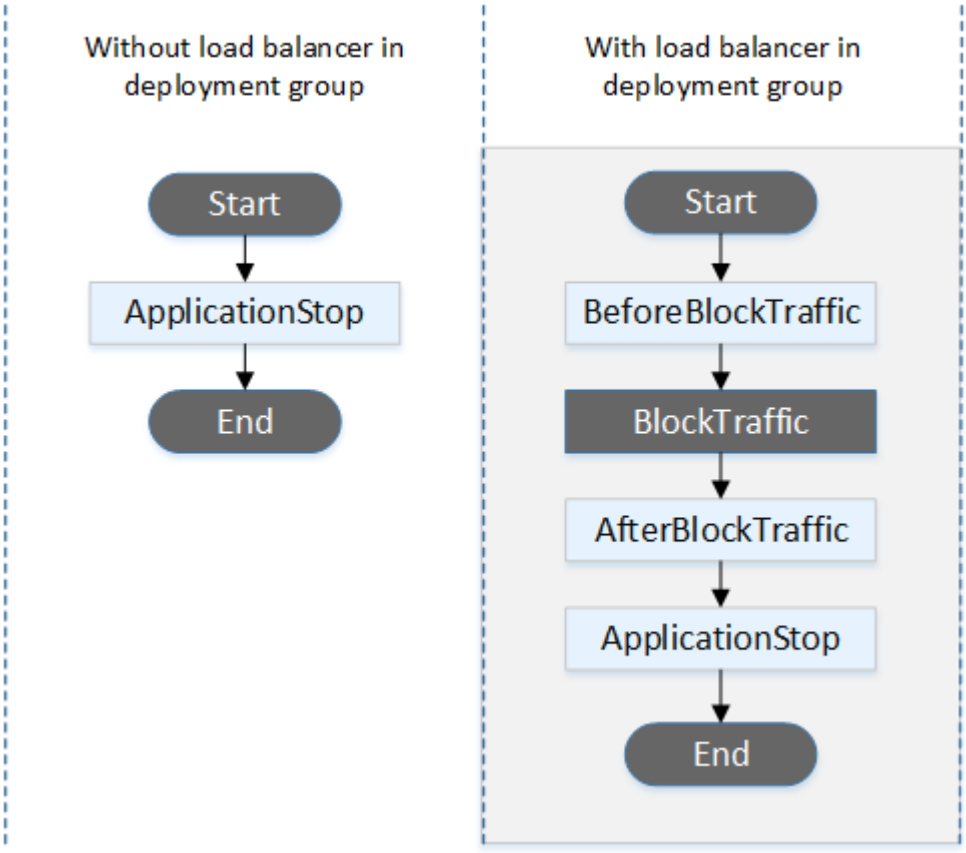


> **Note**
>
> Os eventos Start DownloadBundleAllowTraffic, Install e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama. No entanto, você pode editar a `'files'` seção do AppSpec arquivo para especificar o que será instalado durante o evento de instalação.

Implantações de encerramento do Auto Scaling

Durante uma implantação de encerramento do Auto Scaling, CodeDeploy executa ganchos de eventos na seguinte ordem.

Para obter mais informações sobre as implantações de encerramento do Auto Scaling, consulte Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling.

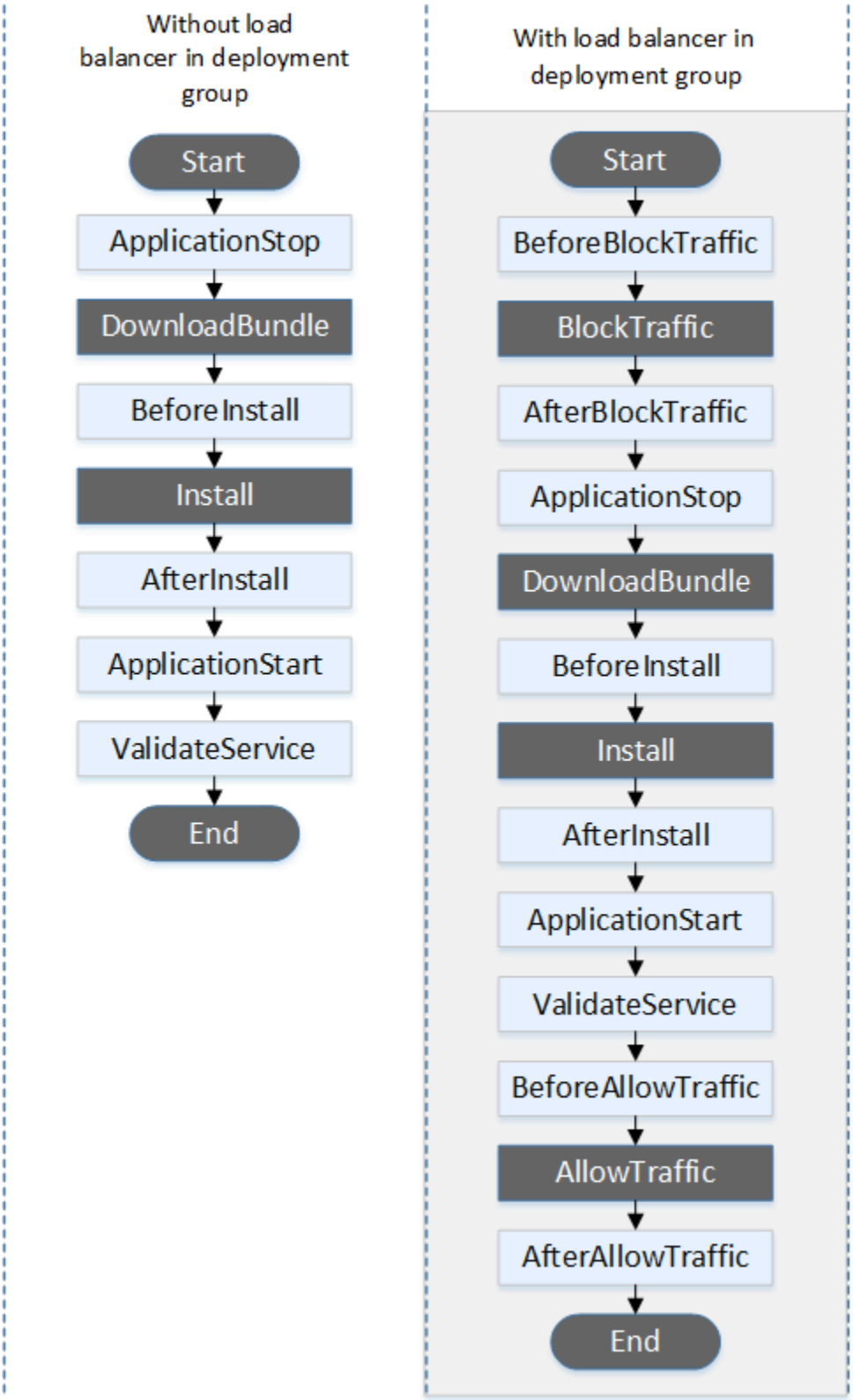


> **Note**
>
> Os eventos Start e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama. BlockTraffic

In-place implantações

Em uma implantação no local, incluindo a reversão de uma implantação no local, ganchos de eventos são executados na seguinte ordem:

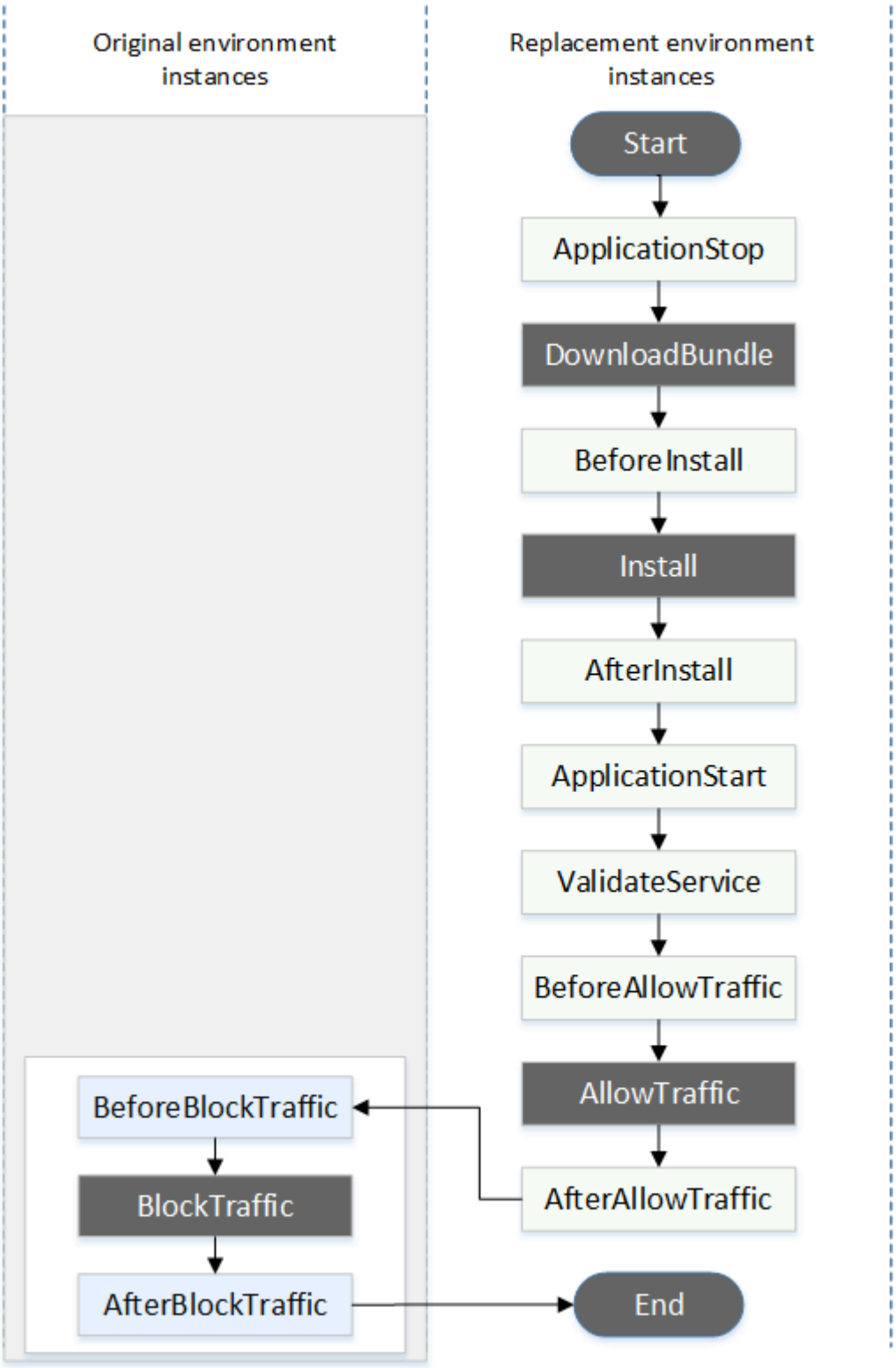
> **Note**
>
> Para implantações no local, os seis hooks relacionados ao bloqueio e à permissão de tráfego apenas serão aplicáveis se você especificar um Classic Load Balancer, Application Load Balancer ou Network Load Balancer do Elastic Load Balancing no grupo de implantação.


Without load balancer in deployment group
Start
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
End
With load balancer in deployment group
Start
BeforeBlockTraffic
BlockTraffic
AfterBlockTraffic
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
End

> **Note**
>
> Os eventos Start DownloadBundle, Install e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama. No entanto, você pode editar a `'files'` seção do AppSpec arquivo para especificar o que será instalado durante o evento de instalação.

Blue/green implantações

Em uma blue/green implantação, os ganchos de eventos são executados na seguinte ordem:


Original environment instances
Replacement environment instances
Start
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
BeforeBlockTraffic
BlockTraffic
AfterBlockTraffic
End

> **Note**
>
> Os eventos Start DownloadBundle, Install BlockTraffic, AllowTraffic, e End na implantação não podem ser programados por script, e é por isso que eles aparecem em cinza neste diagrama. No entanto, você pode editar a seção “arquivos” do AppSpec arquivo para especificar o que será instalado durante o evento de instalação.

Estrutura da seção 'hooks'

A seção `'hooks'` tem a seguinte estrutura:

```
hooks:
   deployment-lifecycle-event-name:
     - location: script-location
       timeout: timeout-in-seconds
       runas: user-name
```

É possível incluir os seguintes elementos em uma entrada hook após o nome do evento de ciclo de vida de implantação:

location

Obrigatório. A localização no pacote do arquivo de script para a revisão. A localização dos scripts que você especifica na seção `hooks` é relativa à raiz do empacotamento de revisão de aplicativo. Para obter mais informações, consulte Planeje uma revisão para CodeDeploy.

timeout

Opcional. O número de segundos para permitir que o script seja executado antes que ele seja considerado com falha. O padrão é 3600 segundos (1 hora).

> **Note**
>
> 3600 segundos (1 hora) é o tempo máximo permitido para a execução do script para cada evento de ciclo de vida de implantação. Se os scripts excederem esse limite, a implantação será interrompida, e a implantação na instância falhará. Certifique-se de que o número total de segundos especificado em timeout para todos os scripts em cada evento de ciclo de vida de implantação não exceda esse limite.

runas

Opcional. O usuário para representar ao executar o script. Por padrão, esse é o CodeDeploy agente em execução na instância. CodeDeploy não armazena senhas, portanto, o usuário não pode ser representado se o usuário runas precisar de uma senha. Esse elemento se aplica somente às instâncias Amazon Linux e Ubuntu Server.

Referenciar arquivos em scripts de hook

Se você estiver conectando um script a um evento de CodeDeploy ciclo de vida, conforme descrito emAppSpec seção 'ganchos', e quiser referenciar um arquivo (por exemplo,`helper.sh`) em seu script, precisará especificar usando: `helper.sh`

- (Recomendado) Um caminho absoluto. Consulte Utilizar caminhos absolutos.
- Um caminho relativo. Consulte Utilizar caminhos relativos.

Utilizar caminhos absolutos

Para referenciar um arquivo utilizando o caminho absoluto, você pode:

- Especifique o caminho absoluto na `files` seção do AppSpec arquivo, na `destination` propriedade. Especificar o mesmo caminho absoluto no script de hook. Para obter mais informações, consulte AppSpec seção 'arquivos' (somente EC2/On-Premises implantações).
- Especificar um caminho absoluto dinâmico no script de hook. Para obter mais informações, consulte Local de arquivamento da implantação.

Local de arquivamento da implantação

Durante o evento do DownloadBundleciclo de vida, o CodeDeploy agente extrai a revisão da implantação em um diretório com o seguinte formato:

root-directory/*deployment-group-id*/*deployment-id*/deployment-archive

A *root-directory* parte do caminho é sempre definida como o padrão mostrado na tabela a seguir ou é controlada pela definição de `:root_dir` configuração. Para obter mais informações sobre as definições das configurações, consulte CodeDeploy referência de configuração do agente.

Plataforma de agentes	Diretório raiz padrão
Linux: todas as distribuições de rpm	`/opt/codedeploy-agent/deployment-root`
Ubuntu Server: todas as distribuições de deb	`/opt/codedeploy-agent/deployment-root`
Windows Server	`%ProgramData%\Amazon\CodeDeploy`

A partir de seus scripts de gancho, você pode acessar o arquivo de implantação atual usando o caminho do diretório raiz `DEPLOYMENT_ID` e as variáveis de `DEPLOYMENT_GROUP_ID` ambiente e. Para obter mais informações sobre as variáveis, consulte Disponibilidade variáveis de ambientes para hooks.

Por exemplo, veja como é possível acessar um arquivo `data.json` que reside na raiz da revisão no Linux:

```
#!/bin/bash

rootDirectory="/opt/codedeploy-agent/deployment-root" # note: this will be different if
 you
                                                      # customize the :root_dir
 configuration
dataFile="$rootDirectory/$DEPLOYMENT_GROUP_ID/$DEPLOYMENT_ID/deployment-archive/
data.json"
data=$(cat dataFile)
```

Como outro exemplo, veja como é possível acessar um arquivo `data.json` que reside na raiz da revisão utilizando o Powershell no Windows:

```
$rootDirectory="$env:ProgramData\Amazon\CodeDeploy" # note: this will be different if
 you
                                                    # customize the :root_dir
 configuration
$dataFile="$rootDirectory\$env:DEPLOYMENT_GROUP_ID\$env:DEPLOYMENT_ID\deployment-
archive\data.json"
$data=(Get-Content $dataFile)
```

Utilizar caminhos relativos

Para referenciar um arquivo usando seu caminho relativo, você precisará conhecer o diretório de trabalho do CodeDeploy agente. Os caminhos dos arquivos são relativos a esse diretório.

A tabela a seguir mostra o diretório de trabalho de cada plataforma compatível do CodeDeploy agente.

Plataforma de agentes	Método de gerenciamento do processo	Diretório de trabalho de scripts de eventos de ciclo de vida
Linux: todas as distribuições de rpm	systemd (padrão)	`/`
	init.d: saiba mais	`/opt/codedeploy-agent`
Ubuntu Server: todas as distribuições de debian	todas	`/opt/codedeploy-agent`
Windows Server	não aplicável	`C:\Windows\System32`

Disponibilidade variáveis de ambientes para hooks

Durante cada evento de ciclo de vida de implantação, os scripts hook podem acessar as seguintes variáveis de ambiente:

APPLICATION_NAME

O nome do aplicativo CodeDeploy que faz parte da implantação atual (por exemplo,`WordPress_App`).

DEPLOYMENT_ID

O ID CodeDeploy foi atribuído à implantação atual (por exemplo,`d-AB1CDEF23`).

DEPLOYMENT_GROUP_NAME

O nome do grupo de implantação CodeDeploy que faz parte da implantação atual (por exemplo,`WordPress_DepGroup`).

DEPLOYMENT_GROUP_ID

O ID do grupo de implantação CodeDeploy que faz parte da implantação atual (por exemplo,`b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE`).

LIFECYCLE_EVENT

O nome do evento de ciclo de vida de implantação atual (por exemplo, `AfterInstall`).

Essas variáveis de ambiente são locais para cada evento de ciclo de vida de implantação.

Há variáveis de ambiente adicionais disponíveis para scripts de hook, dependendo da origem do empacotamento de implantação:

Empacotamento do Amazon S3

- BUNDLE_BUCKET

  O nome do bucket do Amazon S3 do qual o empacotamento de implantação foi baixado por download (por exemplo, `my-s3-bucket`).

- BUNDLE_KEY

  A chave do objeto para o empacotamento baixado dentro do bucket do Amazon S3 (por exemplo, `WordPress_App.zip`).

- BUNDLE_VERSION

  A versão do objeto do empacotamento (por exemplo, `3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo`). Essa variável só é definida se o bucket do Amazon S3 tiver o versionamento de objetos ativado.

- BUNDLE_ETAG

  A etag do objeto do empacotamento (por exemplo, `b10a8db164e0754105b7a99be72e3fe5-4`).

Pacote de GitHub

- BUNDLE_COMMIT

  O hash de confirmação SHA256 do empacotamento gerado pelo Git (por exemplo, `d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26`).

O script a seguir mudará a porta de escuta em um servidor Apache HTTP para 9090 em vez de 80 se o valor de DEPLOYMENT_GROUP_NAME for igual a `Staging`. Este script deve ser invocado durante o evento de ciclo de vida de implantação `BeforeInstall`:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

O exemplo de script a seguir alterará o nível de detalhamento das mensagens registradas em seu log de erros de aviso para depuração quando o valor da variável de ambiente DEPLOYMENT_GROUP_NAME for igual a `Staging`. Este script deve ser invocado durante o evento de ciclo de vida de implantação `BeforeInstall`:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

O exemplo de script a seguir substitui o texto na página da Web especificada pelo texto que exibe o valor dessas variáveis de ambiente. Este script deve ser invocado durante o evento de ciclo de vida de implantação `AfterInstall`:

```
#!/usr/bin/python

import os

strToSearch="<h2>This application was deployed using CodeDeploy.</h2>"
strToReplace="<h2>This page for "+os.environ['APPLICATION_NAME']+"
 application and "+os.environ['DEPLOYMENT_GROUP_NAME']+" deployment group with
 "+os.environ['DEPLOYMENT_GROUP_ID']+" deployment group ID was generated by a
 "+os.environ['LIFECYCLE_EVENT']+" script during "+os.environ['DEPLOYMENT_ID']+"
 deployment.</h2>"

fp=open("/var/www/html/index.html","r")
buffer=fp.read()
fp.close()

fp=open("/var/www/html/index.html","w")
fp.write(buffer.replace(strToSearch,strToReplace))
fp.close()
```

Exemplo de hooks

Veja a seguir um exemplo de uma entrada de ganchos que especifica dois ganchos para o evento de ciclo de vida `AfterInstall`:

```
hooks:
   AfterInstall:
     - location: Scripts/RunResourceTests.sh
       timeout: 180
     - location: Scripts/PostDeploy.sh
       timeout: 180
```

O script `Scripts/RunResourceTests.sh` será executado durante o estágio `AfterInstall` do processo de implantação. A implantação não terá êxito se o script precisar de mais de 180 segundos (3 minutos) para ser executado.

A localização dos scripts que você especifica na seção hooks é relativa à raiz do pacote de revisão de aplicativo. No exemplo anterior, um arquivo chamado `RunResourceTests.sh` está em um diretório chamado `Scripts`. O diretório `Scripts` está no nível raiz do pacote. Para obter mais informações, consulte Planeje uma revisão para CodeDeploy.

AppSpec Exemplo de arquivo

Este tópico fornece AppSpec arquivos de exemplo para um AWS Lambda e uma EC2/On-Premises implantação.

Tópicos

- AppSpec Exemplo de arquivo para uma implantação do Amazon ECS
- AppSpec Exemplo de arquivo para um AWS implantação do Lambda
- AppSpec Exemplo de arquivo para uma EC2/On-Premises implantação

AppSpec Exemplo de arquivo para uma implantação do Amazon ECS

Aqui está um exemplo de um AppSpec arquivo escrito em YAML para implantar um serviço Amazon ECS.

```
version: 0.0
Resources:
  - TargetService:
```

```
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:us-east-1:111222333444:task-definition/my-task-
definition-family-name:1"
        LoadBalancerInfo:
          ContainerName: "SampleApplicationName"
          ContainerPort: 80
# Optional properties
        PlatformVersion: "LATEST"
        NetworkConfiguration:
          AwsvpcConfiguration:
            Subnets: ["subnet-1234abcd","subnet-5678abcd"]
            SecurityGroups: ["sg-12345678"]
            AssignPublicIp: "ENABLED"
        CapacityProviderStrategy:
          - Base: 1
            CapacityProvider: "FARGATE_SPOT"
            Weight: 2
          - Base: 0
            CapacityProvider: "FARGATE"
            Weight: 1
Hooks:
  - BeforeInstall: "LambdaFunctionToValidateBeforeInstall"
  - AfterInstall: "LambdaFunctionToValidateAfterInstall"
  - AfterAllowTestTraffic: "LambdaFunctionToValidateAfterTestTrafficStarts"
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeAllowingProductionTraffic"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterAllowingProductionTraffic"
```

Aqui está uma versão do exemplo anterior escrito em JSON.

```
{
    "version": 0.0,
    "Resources": [
        {
            "TargetService": {
                "Type": "AWS::ECS::Service",
                "Properties": {
                    "TaskDefinition": "arn:aws:ecs:us-east-1:111222333444:task-
definition/my-task-definition-family-name:1",
                    "LoadBalancerInfo": {
                        "ContainerName": "SampleApplicationName",
                        "ContainerPort": 80
                    },
```

```
                    "PlatformVersion": "LATEST",
                    "NetworkConfiguration": {
                        "AwsvpcConfiguration": {
                            "Subnets": [
                                "subnet-1234abcd",
                                "subnet-5678abcd"
                            ],
                            "SecurityGroups": [
                                "sg-12345678"
                            ],
                            "AssignPublicIp": "ENABLED"
                        }
                    },
                    "CapacityProviderStrategy": [
                        {
                            "Base" : 1,
                            "CapacityProvider" : "FARGATE_SPOT",
                            "Weight" : 2
                        },
                        {
                            "Base" : 0,
                            "CapacityProvider" : "FARGATE",
                            "Weight" : 1
                        }
                    ]
                }
            }
        }
    ],
    "Hooks": [
        {
            "BeforeInstall": "LambdaFunctionToValidateBeforeInstall"
        },
        {
            "AfterInstall": "LambdaFunctionToValidateAfterInstall"
        },
        {
            "AfterAllowTestTraffic": "LambdaFunctionToValidateAfterTestTrafficStarts"
        },
        {
            "BeforeAllowTraffic":
"LambdaFunctionToValidateBeforeAllowingProductionTraffic"
        },
        {
```

```
            "AfterAllowTraffic":
 "LambdaFunctionToValidateAfterAllowingProductionTraffic"
        }
    ]
}
```

Veja a seguir a sequência de eventos durante a implantação:

1. Antes que o aplicativo do Amazon ECS atualizado seja instalado no conjunto de tarefas de substituição, a função do Lambda chamada `LambdaFunctionToValidateBeforeInstall` é executada.
2. Depois que o aplicativo do Amazon ECS atualizado seja instalado no conjunto de tarefas de substituição, mas antes que ele receba tráfego, a função do Lambda chamada `LambdaFunctionToValidateAfterInstall` é executada.
3. Depois que o aplicativo do Amazon ECS no conjunto de tarefas de substituição começar a receber tráfego do receptor de teste, a função do Lambda chamada `LambdaFunctionToValidateAfterTestTrafficStarts` será executada. Essa função provavelmente executa testes de validação para determinar se a implantação continua. Se você não especificar um listener de teste no seu grupo de implantação, este gancho será ignorado.
4. Depois que todos os testes de validação no hook `AfterAllowTestTraffic` forem concluídos, e antes que o tráfego de produção seja fornecido para o aplicativo do Amazon ECS atualizado, a função do Lambda chamada `LambdaFunctionToValidateBeforeAllowingProductionTraffic` é executada.
5. Depois que o tráfego de produção for fornecido para o aplicativo do Amazon ECS atualizado no conjunto de tarefas de substituição, a função do Lambda chamada `LambdaFunctionToValidateAfterAllowingProductionTraffic` é executada.

As funções do Lambda que são executadas durante qualquer hook podem executar testes de validação ou coletar métricas de tráfego.

AppSpec Exemplo de arquivo para um AWS implantação do Lambda

Aqui está um exemplo de um AppSpec arquivo escrito em YAML para implantar uma versão da função Lambda.

```
version: 0.0
Resources:
```

```
  - myLambdaFunction:
      Type: AWS::Lambda::Function
      Properties:
        Name: "myLambdaFunction"
        Alias: "myLambdaFunctionAlias"
        CurrentVersion: "1"
        TargetVersion: "2"
Hooks:
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Aqui está uma versão do exemplo anterior escrito em JSON.

```
{
  "version": 0.0,
  "Resources": [{
   "myLambdaFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "myLambdaFunction",
     "Alias": "myLambdaFunctionAlias",
     "CurrentVersion": "1",
     "TargetVersion": "2"
    }
   }
  }],
  "Hooks": [{
    "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
      },
      {
    "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
   }
  ]
 }
```

Veja a seguir a sequência de eventos durante a implantação:

1. Antes de deslocar o tráfego da versão 1 de uma função do Lambda chamada `myLambdaFunction` para a versão 2, execute uma função do Lambda chamada `LambdaFunctionToValidateBeforeTrafficShift` que comprova que a implantação está pronta para iniciar o deslocamento do tráfego.

2. Se `LambdaFunctionToValidateBeforeTrafficShift` retornou um código de saída de 0 (êxito), comece a deslocar tráfego para a versão 2 da `myLambdaFunction`. A configuração de implantação para esta implantação determina a taxa em que o tráfego é deslocado.
3. Após o deslocamento do tráfego da versão 1 de uma função do Lambda chamada `myLambdaFunction` para a versão 2 for concluído, execute uma função do Lambda chamada `LambdaFunctionToValidateAfterTrafficShift` que comprova que a implantação foi concluída com êxito.

AppSpec Exemplo de arquivo para uma EC2/On-Premises implantação

Aqui está um exemplo de um AppSpec arquivo para uma implantação local em uma instância Amazon Linux, Ubuntu Server ou RHEL.

> **Note**
>
> As implantações em instâncias do Windows Server não oferecem suporte ao elemento `runas`. Se você estiver implantando em instâncias do Windows Server, não o inclua em seu AppSpec arquivo.

```
version: 0.0
os: linux
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
hooks:
  BeforeInstall:
    - location: Scripts/UnzipResourceBundle.sh
    - location: Scripts/UnzipDataBundle.sh
  AfterInstall:
    - location: Scripts/RunResourceTests.sh
      timeout: 180
  ApplicationStart:
    - location: Scripts/RunFunctionalTests.sh
      timeout: 3600
  ValidateService:
    - location: Scripts/MonitorService.sh
      timeout: 3600
```

```
      runas: codedeployuser
```

Para uma instância do Windows Server, mude `os: linux` para `os: windows`. Além disso, você deve qualificar completamente os caminhos de `destination` (por exemplo, `c:\temp\webapps\Config` e `c:\temp\webapps\myApp`). Não inclua o elemento `runas`.

Veja a seguir a sequência de eventos durante a implantação:

1. Execute o script localizado em `Scripts/UnzipResourceBundle.sh`.
2. Se o script anterior retornou um código de saída de 0 (êxito), execute o script localizado em `Scripts/UnzipDataBundle.sh`.
3. Copie o arquivo do caminho `Config/config.txt` para o caminho `/webapps/Config/config.txt`.
4. Copie recursivamente todos os arquivos no diretório `source` para o diretório `/webapps/myApp`.
5. Execute o script localizado em `Scripts/RunResourceTests.sh` com um tempo limite de 180 segundos (3 minutos).
6. Execute o script localizado em `Scripts/RunFunctionalTests.sh` com um tempo limite de 3600 segundos (1 hora).
7. Execute o script localizado em `Scripts/MonitorService.sh` como o usuário `codedeploy` com um tempo limite de 3600 segundos (1 hora).

AppSpec Espaçamento de arquivos

A seguir está o formato correto para espaçamento de AppSpec arquivos. Os números entre colchetes indicam o número de espaços que deve haver entre os itens. Por exemplo, `[4]` significa inserir quatro espaços entre os itens. CodeDeploy gera um erro que pode ser difícil de depurar se os locais e o número de espaços em um AppSpec arquivo não estiverem corretos.

```
version:[1]version-number
os:[1]operating-system-name
files:
[2]-[1]source:[1]source-files-location
[4]destination:[1]destination-files-location
permissions:
[2]-[1]object:[1]object-specification
[4]pattern:[1]pattern-specification
[4]except:[1]exception-specification
[4]owner:[1]owner-account-name
```

```
[4]group:[1]group-name
[4]mode:[1]mode-specification
[4]acls:
[6]-[1]acls-specification
[4]context:
[6]user:[1]user-specification
[6]type:[1]type-specification
[6]range:[1]range-specification
[4]type:
[6]-[1]object-type
hooks:
[2]deployment-lifecycle-event-name:
[4]-[1]location:[1]script-location
[6]timeout:[1]timeout-in-seconds
[6]runas:[1]user-name
```

Aqui está um exemplo de um AppSpec arquivo corretamente espaçado:

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

Para obter mais informações sobre espaçamento, consulte a especificação YAML.

Valide seu AppSpec arquivo e a localização do arquivo

Sintaxe de arquivos

Você pode usar o script do AppSpec Assistente AWS fornecido para validar o conteúdo de um AppSpec arquivo. Você pode encontrar o script junto com os modelos de AppSpec arquivo em GitHub.

Também é possível usar uma ferramenta baseada em navegador, como YAML Lint ou Online YAML Parser para ajudar a verificar a sintaxe de YAML.

Local do arquivo

Para verificar se você colocou seu AppSpec arquivo no diretório raiz da estrutura de diretórios do conteúdo de origem do aplicativo, execute um dos seguintes comandos:

Em instâncias locais do Linux, macOS ou Unix:

```
ls path/to/root/directory/appspec.yml
```

Se o AppSpec arquivo não estiver localizado lá, um erro “Não existe esse arquivo ou diretório” será exibido.

Em instâncias locais do Windows:

```
dir path\to\root\directory\appspec.yml
```

Se o AppSpec arquivo não estiver localizado lá, um erro “Arquivo não encontrado” será exibido.

CodeDeploy referência de configuração do agente

Quando o CodeDeploy agente é instalado, um arquivo de configuração é colocado na instância. Esse arquivo de configuração especifica caminhos de diretório e outras configurações CodeDeploy a serem usadas ao interagir com a instância. É possível alterar algumas das opções de configuração no arquivo.

Para instâncias do Amazon Linux, Ubuntu Server e Red Hat Enterprise Linux (RHEL), o arquivo de configuração se chama `codedeployagent.yml`. É possível encontrá-la no diretório `/etc/codedeploy-agent/conf`.

Para instâncias do Windows Server, o arquivo de configuração se chama `conf.yml`. É possível encontrá-la no diretório `C:\ProgramData\Amazon\CodeDeploy`.

As configurações são:

:log_aws_wire:	Defina **`true`** para que o CodeDeploy agente capture logs de conexão do Amazon S3 e os grave em um arquivo de log no local indicado pela configuração:log_dir:. Para a versão 2.0.x e posterior, o nome do arquivo é `codedeploy-agent.aws_wire.log` (criado com as permissões 0640). Para a versão 1.8.x e anteriores, o nome do arquivo é. `codedeploy-agent.wire.log` **Warning** Você deve definir :log_aws_wire: como `true` somente durante o tempo necessário para capturar logs de conexão. O arquivo de registro de conexões pode crescer rapidamente para um tamanho muito grande. A saída do log de conexão pode conter informações confidenciais, incluindo o conteúdo em texto simples dos arquivos transferidos para ou do Amazon S3 enquanto essa configuração estava definida como. `true` Os registros eletrônicos contêm informações sobre todas as atividades do Amazon S3 associadas à AWS conta enquanto essa configuração estava definida`true`, não apenas atividades

	relacionadas às CodeDeploy implantaç ões. A configuração padrão é `false`. Essa configuração aplica-se a todos os tipos de instâncias. Você deve adicionar essa definição de configuração a instâncias do Windows Server para poder usá-la.
:log_dir:	A pasta na instância em que os arquivos de log relacionados às operações do CodeDeploy agente são armazenados. A configuração padrão é `'/var/log/aws/codedeploy-agent'` para instâncias Amazon Linux, Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy\log` para instâncias Windows Server.
:pid_dir:	A pasta onde `codedeploy-agent.pid` está armazenado. Esse arquivo contém o ID do processo (PID) do CodeDeploy agente. A configuração padrão é `'/opt/codedeploy-agent/state/.pid'` . Essa configuração se aplica somente às instâncias Amazon Linux, Ubuntu Server e RHEL.

:program_name:	O nome do programa do CodeDeploy agente. A configuração padrão é `codedeploy-agent`. Essa configuração se aplica somente às instâncias Amazon Linux, Ubuntu Server e RHEL.
:root_dir:	A pasta em que as revisões relacionadas, o histórico de implantação e os scripts de implantação na instância estão armazenados. A configuração padrão é `/opt/codedeploy-agent/deployment-root` para instâncias Amazon Linux, Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy` para instâncias Windows Server.
:verbose:	Defina como `true` para que o CodeDeploy agente imprima arquivos de log de mensagens de depuração na instância. A configuração padrão é `false`.
:wait_between_runs:	O intervalo, em segundos, entre a consulta do CodeDeploy agente CodeDeploy para implantações pendentes. Para a versão 2.0.x e posterior, a configuração padrão é. `30` Para a versão 1.8.x e anteriores, a configuração padrão é. `1`

:on_premises_config_file:	Para instâncias on-premises, o caminho para um local alternativo para o arquivo de configuração denominado `codedeploy.onpremises.yml` (para Ubuntu Server e RHEL) ou `conf.onpremises.yml` (para Windows Server). Por padrão, esses arquivos são armazenados em `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml` para Ubuntu Server e RHEL e `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml` para Windows Server. Disponível na versão 1.0.1.686 e em versões posteriores do agente do CodeDeploy.
:proxy_uri:	(Opcional) O proxy HTTP por meio do qual você deseja que o CodeDeploy agente se conecte AWS para suas CodeDeploy operações. Use um formato semelhante a `https://user:password@my.proxy:443/path?query` . Disponível na versão 1.0.1.824 e em versões posteriores do agente do CodeDeploy.

:max_revisions:	(Opcional) O número de revisões de aplicativos para um grupo de implantação que você deseja que o CodeDeploy agente arquive. Todas as revisões que excederem o número especific ado serão excluídas. Insira qualquer número inteiro positivo. Se nenhum valor for especificado, CodeDeploy reterá as cinco revisões mais recentes, além da revisão atualmente implantada. Com suporte na versão 1.0.1.966 e versões posteriores do agente do CodeDeploy.
:enable_auth_policy:	(Opcional) Defina como `true` se quiser usar a autorização do IAM para configurar o controle de acesso e limitar a permissão da função ou do usuário do IAM que o CodeDeploy agente está usando. Para Use CodeDeploy com a Amazon Virtual Private Cloud, esse valor deverá ser `true`. A configuração padrão é `false`.
:disable_imds_v1:	Essa configuração está disponível com o CodeDeploy agente 1.7.0 e versões posterior es. Defina como `true` para desabilitar o fallback para o IMDSv1 quando ocorrerem erros do IMDSv2. O padrão é `false` (habilitar o fallback).

:disable_core_dumps:	Para a versão 2.0.x e posterior. Defina como `true` para suprimir os despejos principais. Definido como `false` para investiga ção de acidentes. A configuração padrão é `true`.
:ativar_porta_de comando:	Para a versão 2.0.x e posterior. Defina como `true` para ativar a interface de gerenciamento local somente de loopback (127.0.0.1) para consultar o status do agente e injetar comandos localmente. A configuração padrão é `false`.
:habilitar_archive_reuse:	Para a versão 2.1.0 e posterior. Defina como `false` para sempre baixar a revisão do aplicativo, mesmo em uma `RESTART` implantação em que a revisão já esteja na instância. Quando essa configuração é`true`, o agente instala a partir da cópia que já está na instância e baixa a revisão se essa cópia estiver ausente ou incompleta. A configuração padrão é `true`. Para obter mais informações sobre esse comportamento, consulte Histórico de versões do CodeDeploy agente.
:reject_symlinks_in_bundle:	Para a versão 2.0.x e posterior. Defina como `true` para rejeitar pacotes que contenham links simbólicos ou hardlinks. A configuração padrão é `false`.

:reject_path_traversal_in_bundle:	Para a versão 2.0.x e posterior. Defina como `true` para rejeitar entradas de arquivamento que contenham `../` caminhos percorridos ou caminhos absolutos e campos de AppSpec origem que escapam da raiz do pacote. A configuração padrão é `false`.
:reject_unsafe_permissions_in_bundle:	Para a versão 2.0.x e posterior. Defina como `true` para rejeitar pacotes que contenham SUID/SGID arquivos ou modos de AppSpec permissão definidos SUID/SGID. A configuração padrão é `false`.
:reject_unconfined_selinux_em_bundle:	Para a versão 2.0.x e posterior. Defina como `true` para rejeitar pacotes que contenham arquivos com tipos MAC-disabling SELinux. A configuração padrão é `false`.
:rejeitar alvos de permissão de link simbólico:	Para a versão 2.0.x e posterior. Defina como `true` para rejeitar uma `permissions` entrada de AppSpec arquivo cujo destino é um link simbólico. Por padrão, o agente segue o link quando aplica alterações de propriedade, modo, ACL e SELinux. A configuração padrão é `false`.

:ignore_ownership_in_bundle:	Para a versão 2.0.x e posterior. Defina como `true` para redefinir a propriedade dos arquivos extraídos `tar` e dos `tgz` pacotes `root` em vez de aplicar o ID do usuário e o ID do grupo registrados no arquivamento. Os arquivos nos `zip` pacotes são de propriedade de`root`, independentemente dessa configura ção. A configuração padrão é `false`.
:restringir_agent_dir_permissions:	Para a versão 2.0.x e posterior. Somente instâncias Linux. Defina como `true` para restringir os diretório s e arquivos na raiz de instalação do agente ao acesso somente raiz, em vez dos modos legíveis mundialmente usados por padrão. A configuração padrão é `false`.
:restrict_log_dir_permissions:	Para a versão 2.0.x e posterior. Somente instâncias Linux. Defina como `true` para restringir o diretório de registros do agente, que contém o registro do agente e o registro do atualizador, ao acesso somente raiz, em vez dos modos legíveis por todos usados por padrão. A configuração padrão é `false`.

:restrinja o ambiente do hook_to_allowlist:	Para a versão 2.0.x e posterior. Defina como `true` para executar ganchos de eventos de ciclo de vida com um ambiente mínimo que contenha somente as variáveis básicas do shell e as CodeDeploy variáveis de implantação. Por padrão, os ganchos herdam o ambiente completo do processo do agente. A configuração padrão é `false`.
:strip_loader_env_in_hooks:	Para a versão 2.0.x e posterior. Instâncias do Linux e instâncias do macOS com a versão 2.1.2 e posterior. Defina como `true` para remover as `LD_AUDIT` variáveis`LD_PRELOAD` ,`LD_LIBRARY_PATH` , e do ambiente dos ganchos de eventos do ciclo de vida. No macOS, a configuração remove as `DYLD_VERSIONED_FRAMEWORK_PATH` variáveis`DYLD_INSERT_LIBRARIES` ,,,`DYLD_LIBRARY_PATH` ,`DYLD_FRAMEWORK_PATH` , e `DYLD_FALLBACK_LIBRARY_PATH` `DYLD_FALLBACK_FRAMEWORK_PATH` `DYLD_VERSIONED_LIBRARY_PATH` , em vez disso. A configuração padrão é `false`.

:disable_powershell_profile_in_hooks:	Para a versão 2.0.x e posterior. Somente instâncias do Windows Server. Defina como `true` para executar ganchos de eventos `.ps1` de ciclo de vida com as opções e. `-NoProfile -NonInteractive` Por padrão, esses ganchos carregam o PowerShel l perfil do usuário com o qual o agente é executado. A configuração padrão é `false`.
:archive_max_extration_size:	Para a versão 2.0.x e posterior. Define um limite para o tamanho declarado do pacote não comprimido. Se o conteúdo extraído exceder esse valor, a implantação falhará. O padrão é sem limite (ilimitado).
:wait_after_error:	O número de segundos que o agente espera após encontrar um erro antes de retomar a operação normal. A configuração padrão é `30`.
:timeout do http_read_out:	O tempo limite de leitura HTTP em segundos para chamadas de API de serviço. A configuração padrão é `80`.
:kill_agent_max_wait_time_seconds:	O número máximo de segundos que o agente espera por um desligamento normal antes da finalização forçada. Para a versão 2.0.x e posterior, esse valor é limitado a 24 horas (86400 segundos). A configuração padrão é `7200` (2 horas).

:habilitar_log_de_implantações:	Defina como `true` para gravar o registro agregado de implantações ()`codedeploy-agent-deployments.log` . A configuração padrão é `true`.
:use_fips_mode:	Defina como `true` para usar FIPS-compliant endpoints. Aplica-se somente às regiões dos EUA. A configuração padrão é `false`.
:deploy_control_endpoint:	(Opcional) Um URL de endpoint do plano de controle personalizado para o serviço. CodeDeploy O padrão é não substituir (o agente resolve o endpoint da região).
:s3_endpoint_override:	(Opcional) Um URL de endpoint personalizado do Amazon S3 para downloads de pacotes. O padrão é não substituir (o agente resolve o endpoint do Amazon S3 da região).
: s3_download_max_concurrency:	Para a versão 2.1.2 e posterior. O número de solicitações de intervalo HTTP paralelo que o agente usa para baixar uma revisão do aplicativo do Amazon S3. Defina um valor maior do que 1 para baixar grandes revisões em intervalos paralelos. Com`1`, o agente baixa a revisão com uma única `GetObject` solicitação. Os valores acima `64` são tratados como`64`, e `0` são tratados como`1`. A configuração padrão é `1`.

:s3_download_chunk_size:	Para a versão 2.1.2 e posterior. O tamanho de cada solicitação de intervalo quando `:s3_download_max_concurrency:` é maior que `1`. Insira um número de bytes ou uma string de tamanho, como `"16MiB"` ou `"16MB"`. Uma revisão desse tamanho ou menor é baixada com uma solicitação. Valores abaixo de 64 KiB são tratados como 64 KiB. A configuração padrão é `"8MiB"`.
:rastreamento_de_implantação em andamento:	O nome do subdiretório usado para o estado de rastreamento da implantação. A configuração padrão é `ongoing-deployment` .

Tópicos relacionados

Trabalhando com o CodeDeploy agente

Gerenciando as operações CodeDeploy do agente

CloudFormation modelos para CodeDeploy referência

Esta seção apresenta os CloudFormation recursos, a transformação e o gancho projetados para funcionar com CodeDeploy implantações. Para ver um passo a passo da criação de uma atualização de pilha gerenciada pelo CloudFormation hook for, consulte CodeDeploy Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation

> **Note**
>
> CloudFormation os ganchos fazem parte dos CloudFormation componentes AWS e são diferentes dos ganchos de eventos do CodeDeploy ciclo de vida.

Além dos outros métodos disponíveis em CodeDeploy, você pode usar CloudFormation modelos para realizar as seguintes tarefas:

- Crie aplicativos.
- Crie grupos de implantação e especifique uma revisão de destino.
- Crie configurações de implantação.
- Crie instâncias do Amazon EC2.

CloudFormation é um serviço que ajuda você a modelar e configurar seus AWS recursos usando modelos. Um CloudFormation modelo é um arquivo de texto cujo formato está em conformidade com o padrão JSON. Você cria um modelo que descreve todos os AWS recursos desejados e se CloudFormation encarrega de provisionar e configurar esses recursos para você.

Para obter mais informações, consulte O que é o AWS CloudFormation? e Trabalhar com modelos do AWS CloudFormation no Guia do usuário do AWS CloudFormation .

Se você planeja usar CloudFormation modelos compatíveis com CodeDeploy os da sua organização, como administrador, você deve conceder acesso aos CloudFormation AWS serviços e ações dos quais CloudFormation depende. Para conceder permissões para criar aplicativos, grupos de implantação e configurações de implantação, adicione a política a seguir ao conjunto de permissões dos usuários com CloudFormation quem trabalharão:

JSON

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

Para obter mais informações sobre políticas, consulte os seguintes tópicos:

- Para ver a política que deve ser adicionada ao conjunto de permissões dos usuários que criarão instâncias do Amazon EC2, consulte Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo).
- Para obter informações sobre como adicionar políticas aos conjuntos de permissões, consulte Criar um conjunto de permissões no Guia do usuário do IAM.
- Para saber como restringir os usuários a um conjunto limitado de CodeDeploy ações e recursos, consulteAWS políticas gerenciadas (predefinidas) para CodeDeploy.

A tabela a seguir mostra as ações que um CloudFormation modelo pode realizar em seu nome e inclui links para mais informações sobre os tipos de AWS recursos e seus tipos de propriedades que você pode adicionar a um CloudFormation modelo.

Ação	CloudFormation referência	Tipo de referência
Crie um CodeDeploy aplicativo.	AWS::CodeDeploy: :aplicativo	CloudFormation recurso
Crie e especifique os detalhes de um grupo de implantação a ser usado para implantar as revisões de aplicativos. [1]	AWS::CodeDeploy::DeploymentGroup	CloudFormation recurso
Crie um conjunto de regras de implantação, condições de sucesso de implantação e condições de falha de implantação que CodeDeploy serão usadas durante uma implantação.	AWS::CodeDeploy::DeploymentConfig	CloudFormation recurso
Crie uma instância do Amazon EC2. [2]	AWS::EC2::instance	CloudFormation recurso
Use a CloudFormation `AWS::CodeDeployBlueGreen` transformação e o `AWS::CodeDeploy::B`	AWS::CodeDeployBlueGreen	A transformação `AWS::CodeDeployBlueGreen` é uma macro hospedada pelo CloudFormation

Ação	CloudFormation referência	Tipo de referência
`lueGreen` gancho para gerenciar atualizações de pilha, criar recursos e transferir tráfego para CodeDeploy blue/green implantações. [3]	AWS::CodeDeploy::BlueGreen	O `AWS::CodeDeploy::BlueGreen` gancho é estruturado como um `Hook` recurso em CloudFormation. O gancho inclui parâmetros que substituem seu CodeDeploy AppSpec arquivo apontando para ganchos de eventos de CodeDeploy ciclo de vida designados.

Ação	CloudFormation referência	Tipo de referência
[1] Se você especificar a versão da revisão de aplicativo a ser implantada como parte do grupo de implantação, sua revisão de destino será implantada assim que o processo de provision amento estiver completo. Para obter mais informações sobre a configuração do modelo, consulte a revisão de CodeDeplo y DeploymentGroup implantação S3Location e a revisão de CodeDeploy DeploymentGroup implantação GitHubLocation no Guia do AWS CloudFormation usuário. [2] Fornecemos modelos que você pode usar para criar instância s do Amazon EC2 nas regiões em que CodeDeploy há suporte. Para obter mais informações sobre como usar esses templates, consulte Crie uma instância do Amazon EC2 para (CodeDeploy CloudFormation modelo). [3] Somente blue/green implantações do Amazon ECS são suportadas por essa configuração de implantação. Para obter mais informações sobre as configurações de implantação para blue/green implantações do Amazon ECS por meio CloudFormation de, consulte. Configurações de implantação para CloudFormation blue/green implantações (Amazon ECS) Para obter mais informações sobre blue/green implantações do Amazon ECS por meio de CloudFormation e como visualizar sua implantação em CodeDeploy, consulte. Crie uma blue/green implantação do Amazon ECS por meio de CloudFormation		

Use CodeDeploy com a Amazon Virtual Private Cloud

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e. CodeDeploy Você pode usar essa conexão para permitir CodeDeploy a comunicação com seus recursos em sua VPC sem passar pela Internet pública.

O Amazon VPC é um AWS serviço que você pode usar para lançar AWS recursos em uma rede virtual que você define. Com a VPC, você tem controle sobre as configurações de rede, como o

intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Com os VPC endpoints, o roteamento entre a VPC e os AWS serviços é gerenciado pela AWS rede, e você pode usar políticas do IAM para controlar o acesso aos recursos do serviço.

Para conectar sua VPC a CodeDeploy, você define uma interface para a qual VPC endpoint. CodeDeploy Um endpoint de interface é uma interface de rede elástica com um endereço IP privado que serve como ponto de entrada para o tráfego destinado a um serviço compatível AWS. O endpoint fornece conectividade confiável e escalável CodeDeploy sem a necessidade de um gateway de internet, instância de tradução de endereços de rede (NAT) ou conexão VPN. Para obter mais informações, consulte O que é a Amazon VPC? no Guia do usuário da Amazon VPC.

Os endpoints VPC da Interface são alimentados por AWS PrivateLink uma AWS tecnologia que permite a comunicação privada entre AWS serviços usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte AWS PrivateLink.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte Conceitos básicos no Guia do usuário da Amazon VPC.

Disponibilidade

CodeDeploy tem dois VPC endpoints: um para operações de CodeDeploy agentes e outro para CodeDeploy operações de API. A tabela abaixo mostra as AWS regiões suportadas para cada endpoint.

Para acessar mais informações sobre endpoints do FIPS, consulte Endpoints e cotas do AWS CodeDeploy.

Nome da região	Código da região	Endpoint do agente	Ponto final da API	Região compatível com FIPS?
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim	Sim	Sim
Leste dos EUA (Ohio)	us-east-2	Sim	Sim	Sim

Nome da região	Código da região	Endpoint do agente	Ponto final da API	Região compatível com FIPS?
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim	Sim	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim	Sim	Sim
África (Cidade do Cabo)	af-south-1	Sim	Sim	
Ásia-Pacífico (Hong Kong)	ap-east-1	Sim	Sim	
Ásia-Pacífico (Hyderabad)	ap-south-2	Sim	Sim	
Ásia-Pacífico (Jacarta)	ap-southeast-3	Sim	Sim	
Ásia-Pacífico (Melbourne)	ap-southeast-4	Sim	Sim	
Ásia-Pacífico (Malásia)	ap-southeast-5	Sim	Sim	
Ásia-Pacífico (Nova Zelândia)	ap-southeast-6	Sim	Sim	
Ásia-Pacífico (Tailândia)	ap-southeast-7	Sim	Sim	
Ásia-Pacífico (Taipei)	ap-east-2	Sim	Sim	
Ásia-Pacífico (Mumbai)	ap-south-1	Sim	Sim	

Nome da região	Código da região	Endpoint do agente	Ponto final da API	Região compatível com FIPS?
Ásia Pacifico (Osaka)	ap-northeast-3	Sim	Sim	
Ásia-Pacífico (Seul)	ap-northeast-2	Sim	Sim	
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim	Sim	
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim	Sim	
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim	Sim	
Canadá (Central)	ca-central-1	Sim	Sim	
China (Pequim)	cn-north-1	Sim	Não	
China (Ningxia)	cn-northwest-1	Não	Não	
Europa (Frankfurt)	eu-central-1	Sim	Sim	
Europa (Irlanda)	eu-west-1	Sim	Sim	
Europa (Londres)	eu-west-2	Sim	Sim	
Europa (Milão)	eu-south-1	Sim	Sim	
Europa (Paris)	eu-west-3	Sim	Sim	
Europa (Espanha)	eu-south-2	Sim	Sim	

Nome da região	Código da região	Endpoint do agente	Ponto final da API	Região compatível com FIPS?
Europa (Estocolmo)	eu-north-1	Sim	Sim	
Europa (Zurique)	eu-central-2	Sim	Sim	
Israel (Tel Aviv)	il-central-1	Sim	Sim	
Oriente Médio (Bahrein)	me-south-1	Sim	Sim	
Oriente Médio (Emirados Árabes Unidos)	me-central-1	Sim	Sim	
México (Centro)	mx-central-1	Sim	Sim	
América do Sul (São Paulo)	sa-east-1	Sim	Sim	
AWS GovCloud (US-East)	us-gov-east-1	Não	Não	Sim
AWS GovCloud (US-West)	us-gov-west-1	Não	Não	Sim

Crie endpoints VPC para CodeDeploy

Para começar a usar CodeDeploy com sua VPC, crie uma interface VPC endpoint para. CodeDeploy CodeDeployrequer endpoints separados para as operações do Git do agente e CodeDeploy para as operações da API. Dependendo das necessidades da sua empresa, pode ser necessário criar mais de um VPC endpoint. Ao criar um VPC endpoint para CodeDeploy, escolha AWS Serviços e, em Nome do serviço, escolha entre as seguintes opções:

- com.amazonaws. ***region***.codedeploy: escolha essa opção se quiser criar um VPC endpoint para operações de API. CodeDeploy Por exemplo, escolha essa opção se seus usuários usarem

a CodeDeploy API ou os AWS SDKs com CodeDeploy os quais interagir em operações como `CreateApplicationGetDeployment`, e.AWS CLI`ListDeploymentGroups`

- com.amazonaws. ***region***.codedeploy-fips: escolha essa opção se quiser criar um VPC endpoint para operações de API para o endpoint FIPS. CodeDeploy
- com.amazonaws. ***region***.codedeploy-commands-secure: escolha essa opção se quiser criar um VPC endpoint para operações de agentes. CodeDeploy Você também precisará definir `:enable_auth_policy:` para `true` no arquivo de configuração do agente e anexar as permissões necessárias. Para obter mais informações, consulte Configurar as permissões do CodeDeploy agente e do IAM.

Se você estiver usando implantações Lambda ou ECS, você só precisa criar um VPC endpoint para com.amazonaws. ***region***.codedeploy. Os clientes que usam implantações do Amazon EC2 precisarão de VPC endpoints para ambos com.amazonaws. *region*.codedeploy e com.amazonaws. ***region***.codedeploy-commands-secure.

Configurar as permissões do CodeDeploy agente e do IAM

Para usar os endpoints da Amazon VPC com CodeDeploy, você deve definir o valor de `:enable_auth_policy:` to `true` no arquivo de configuração do agente localizado em suas instâncias EC2 ou locais. Para obter mais informações sobre o arquivo de configuração do agente, consulte CodeDeploy referência de configuração do agente.

Você também deve adicionar as seguintes permissões do IAM ao seu perfil de instância do Amazon EC2 (se estiver usando instâncias do Amazon EC2) ou ao usuário do IAM ou perfil (se estiver usando instâncias on-premises).

```
{
  "Statement": [
    {
      "Action": [
        "codedeploy-commands-secure:GetDeploymentSpecification",
        "codedeploy-commands-secure:PollHostCommand",
        "codedeploy-commands-secure:PutHostCommandAcknowledgement",
        "codedeploy-commands-secure:PutHostCommandComplete"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

CodeDeploy referência do kit de recursos

Muitos dos arquivos nos quais CodeDeploy confio são armazenados em buckets AWS Region-specific Amazon S3 disponíveis publicamente. Esses arquivos incluem arquivos de instalação do CodeDeploy agente, modelos e arquivos de amostra do aplicativo. Chamamos essa coleção de arquivos de CodeDeploy Resource Kit.

Tópicos

- Nomes de bucket do kit de recursos por região
- Conteúdo do kit de recursos
- Exibir uma lista dos arquivos do kit de recursos
- Fazer download dos arquivos do kit de recursos

Nomes de bucket do kit de recursos por região

Esta tabela lista os nomes das *`bucket-name`* substituições necessárias para alguns procedimentos no guia. Esses são os nomes dos buckets do Amazon S3 que contêm os arquivos do CodeDeploy Resource Kit.

> **Note**
>
> Para acessar o bucket do Amazon S3 na região Ásia-Pacífico (Hong Kong), você deve habilitar a região em sua AWS conta. Para obter mais informações, consulte Gerenciando AWS regiões.

Nome da região	***`Bucket-name`*** substituto	Identificador da região
Leste dos EUA (Norte da Virgínia)	aws-codedeploy-us-east-1	us-east-1

Nome da região	***Bucket-name*** substituto	Identificador da região
Leste dos EUA (Ohio)	aws-codedeploy-us-east-2	us-east-2
Oeste dos EUA (N. da Califórnia)	aws-codedeploy-us-west-1	us-west-1
Oeste dos EUA (Oregon)	aws-codedeploy-us-west-2	us-west-2
África (Cidade do Cabo)	aws-codedeploy-af-south-1	af-south-1
Ásia-Pacífico (Hong Kong)	aws-codedeploy-ap-east-1	ap-east-1
Ásia-Pacífico (Hyderabad)	aws-codedeploy-ap-south-2	ap-south-2
Ásia-Pacífico (Jacarta)	aws-codedeploy-ap-southeast-3	ap-southeast-3
Ásia-Pacífico (Melbourne)	aws-codedeploy-ap-southeast-4	ap-southeast-4
Ásia-Pacífico (Malásia)	aws-codedeploy-ap-southeast-5	ap-southeast-5
Ásia-Pacífico (Nova Zelândia)	aws-codedeploy-ap-southeast-6	ap-southeast-6
Ásia-Pacífico (Tailândia)	aws-codedeploy-ap-southeast-7	ap-southeast-7
Ásia-Pacífico (Taipei)	aws-codedeploy-ap-east-2	ap-east-2
Ásia-Pacífico (Mumbai)	aws-codedeploy-ap-south-1	ap-south-1
Ásia-Pacífico (Osaka)	aws-codedeploy-ap-northeast-3	ap-northeast-3
Ásia-Pacífico (Seul)	aws-codedeploy-ap-northeast-2	ap-northeast-2

Nome da região	***Bucket-name*** substituto	Identificador da região
Ásia-Pacífico (Singapura)	aws-codedeploy-ap-southeast-1	ap-southeast-1
Ásia-Pacífico (Sydney)	aws-codedeploy-ap-southeast-2	ap-southeast-2
Ásia-Pacífico (Tóquio)	aws-codedeploy-ap-northeast-1	ap-northeast-1
Canadá (Central)	aws-codedeploy-ca-central-1	ca-central-1
Europa (Frankfurt)	aws-codedeploy-eu-central-1	eu-central-1
Europa (Irlanda)	aws-codedeploy-eu-west-1	eu-west-1
Europa (Londres)	aws-codedeploy-eu-west-2	eu-west-2
Europa (Milão)	aws-codedeploy-eu-south-1	eu-south-1
Europa (Paris)	aws-codedeploy-eu-west-3	eu-west-3
Europa (Espanha)	aws-codedeploy-eu-south-2	eu-south-2
Europa (Estocolmo)	aws-codedeploy-eu-north-1	eu-north-1
Europa (Zurique)	aws-codedeploy-eu-central-2	eu-central-2
Israel (Tel Aviv)	aws-codedeploy-il-central-1	il-central-1
Oriente Médio (Bahrein)	aws-codedeploy-me-south-1	me-south-1
Oriente Médio (Emirados Árabes Unidos)	aws-codedeploy-me-central-1	me-central-1
México (Centro)	aws-codedeploy-mx-central-1	mx-central-1
América do Sul (São Paulo)	aws-codedeploy-sa-east-1	sa-east-1
AWS GovCloud (US-East)	aws-codedeploy-us-gov-east-1	us-gov-east-1

Nome da região	***Bucket-name*** substituto	Identificador da região
AWS GovCloud (US-West)	aws-codedeploy-us-gov-west-1	us-gov-west-1

Conteúdo do kit de recursos

A tabela a seguir lista os arquivos no CodeDeploy Resource Kit.

Arquivo	Description
`LATEST_VERSION`	Um arquivo usado por mecanismos de atualização, como o Amazon EC2 Systems Manager, para determinar a versão mais recente CodeDeploy do agente.
`VERSION`	O mecanismo de atualização automática foi removido na versão 1.1.0 do CodeDeploy agente e esse arquivo não é mais usado. Um arquivo usado por agentes do CodeDeplo y para atualização à medida que eles são executados em instâncias.
`codedeploy-agent.noarch.rpm`	O CodeDeploy agente para Amazon Linux e Red Hat Enterprise Linux (RHEL). Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `-1.0-0`).
`codedeploy-agent_all.deb`	O CodeDeploy agente do Ubuntu Server. Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `_1.0-0`).
`codedeploy-agent.msi`	O CodeDeploy agente do Windows Server. Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `-1.0-0`).

Arquivo	Description
`install`	Um arquivo que você pode usar para instalar o CodeDeploy agente com mais facilidade. O `latestv2/` prefixo serve ao script de instalação da versão 2.0.x.
`CodeDeploy_SampleCF_Template.json`	Um AWS CloudFormation modelo que você pode usar para iniciar de uma a três instâncias do Amazon EC2 executando o Amazon Linux ou o Windows Server. Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `-1.0.0`).
`CodeDeploy_SampleCF_ELB_Integration.json`	Um AWS CloudFormation modelo que você pode usar para criar um site de amostra com balanceamento de carga executado em um servidor Web Apache. O aplicativo está configurado para abranger todas as zonas de disponibilidade na região em que você o criou. Esse modelo cria três instâncias do Amazon EC2 e um perfil de instância do IAM para conceder às instâncias acesso aos recursos no Amazon S3, no Amazon EC2 Auto CloudFormation Scaling e no Elastic Load Balancing. Ele também cria um balanceador de carga e uma função CodeDeploy de serviço.
`SampleApp_ELB_Integration.zip`	Uma revisão de aplicativo de exemplo que pode ser implantada em uma instância do Amazon EC2 registrada em um balanceador de carga Elastic Load Balancing.

Arquivo	Description
`SampleApp_Linux.zip`	Uma amostra de revisão de aplicativo que pode ser implantada em uma instância do Amazon EC2 executando o Amazon Linux ou em uma instância do Ubuntu Server ou RHEL. Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `-1.0`).
`SampleApp_Windows.zip`	Uma amostra de revisão de aplicativo que você pode implantar em uma instância do Windows Server. Pode haver vários arquivos com o mesmo nome de arquivo base, mas versões diferentes (como `-1.0`).

Exibir uma lista dos arquivos do kit de recursos

Para ver uma lista de arquivos, use o aws s3 ls comando para sua região.

> **Note**
>
> Os arquivos em cada bucket são projetados para funcionar com recursos na região correspondente.

- `aws s3 ls --recursive s3://aws-codedeploy-us-east-2 --region us-east-2`
- `aws s3 ls --recursive s3://aws-codedeploy-us-east-1 --region us-east-1`
- `aws s3 ls --recursive s3://aws-codedeploy-us-west-1 --region us-west-1`
- `aws s3 ls --recursive s3://aws-codedeploy-us-west-2 --region us-west-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ca-central-1 --region ca-central-1`

- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-1 --region eu-west-1`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-2 --region eu-west-2`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-3 --region eu-west-3`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-central-1 --region eu-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-il-central-1 --region il-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-east-1 --region ap-east-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-1 --region ap-northeast-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-2 --region ap-northeast-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-1 --region ap-southeast-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-2 --region ap-southeast-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-4 --region ap-southeast-4`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-5 --region ap-southeast-5`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-6 --region ap-southeast-6`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-7 --region ap-southeast-7`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-east-2 --region ap-east-2`
- `aws s3 ls --recursive s3://aws-codedeploy-mx-central-1 --region mx-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-south-1 --region ap-south-1`
- `aws s3 ls --recursive s3://aws-codedeploy-sa-east-1 --region sa-east-1`

Fazer download dos arquivos do kit de recursos

Para baixar um arquivo, use o aws s3 cp comando para sua região.

> **Note**
>
> Certifique-se de usar o ponto final (`.`) perto do final. Isso baixa o arquivo no seu diretório atual.

Por exemplo, os seguintes comandos fazem download de um único arquivo chamado `SampleApp_Linux.zip` em uma das pastas `/samples/latest/` dos buckets:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2`
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1`
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1`
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2`
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1`
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1`

- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1`
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-5/samples/latest/SampleApp_Linux.zip . --region ap-southeast-5`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-6/samples/latest/SampleApp_Linux.zip . --region ap-southeast-6`
- `aws s3 cp s3://aws-codedeploy-ap-southeast-7/samples/latest/SampleApp_Linux.zip . --region ap-southeast-7`
- `aws s3 cp s3://aws-codedeploy-ap-east-2/samples/latest/SampleApp_Linux.zip . --region ap-east-2`
- `aws s3 cp s3://aws-codedeploy-mx-central-1/samples/latest/SampleApp_Linux.zip . --region mx-central-1`
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1`

- ```
  aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region
   sa-east-1
  ```

Para baixar todos os arquivos, use um dos seguintes comandos para sua região:

- ```
  aws s3 cp --recursive s3://aws-codedeploy-us-east-2 . --region us-east-2
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-us-east-1 . --region us-east-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-us-west-1 . --region us-west-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-us-west-2 . --region us-west-2
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ca-central-1 . --region ca-central-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-eu-west-1 . --region eu-west-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-eu-west-2 . --region eu-west-2
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-eu-west-3 . --region eu-west-3
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-eu-central-1 . --region eu-central-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-il-central-1 . --region il-central-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-east-1 . --region ap-east-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-1 . --region ap-northeast-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-2 . --region ap-northeast-2
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-1 . --region ap-southeast-1
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-2 . --region ap-southeast-2
  ```
- ```
  aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-4 . --region ap-southeast-4
  ```
  ```

- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-5 . --region ap-southeast-5`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-6 . --region ap-southeast-6`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-7 . --region ap-southeast-7`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-east-2 . --region ap-east-2`
- `aws s3 cp --recursive s3://aws-codedeploy-mx-central-1 . --region mx-central-1`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-south-1 . --region ap-south-1`
- `aws s3 cp --recursive s3://aws-codedeploy-sa-east-1 . --region sa-east-1`

CodeDeploy cotas

As tabelas a seguir descrevem as cotas em CodeDeploy.

> **Note**
>
> O limite EC2/On-Premises de horas de execução da implantação no local varia. Para configurações de implantação personalizadas criadas antes de junho de 2023, o limite é de oito horas. Para configurações de implantação personalizadas criadas em junho de 2023, o limite é de 12 horas. Para configurações de implantação predefinidas, o limite é de 12 horas.

Nome	Padrão	Ajustá	Description
AWS A implantação do Lambda é executada em horas	Cada região compatível: 50	Não	Número máximo de horas que uma implantação do AWS Lambda pode ser executada (48 horas para o tempo máximo entre a primeira e a última mudança de tráfego, mais uma hora para cada

Nome	Padrão	Ajustá	Description
			um dos dois possíveis ganchos do ciclo de vida)
Aplicativos associados por conta, por região	Cada região com suporte: 1.000	Sim	O número máximo de aplicativos associados a uma AWS conta em uma única região
Alarmes associados por grupo de implantação	Cada região compatível: 50	Sim	O número máximo de alarmes associados a um grupo de implantação
Grupos do Auto Scaling em um grupo de implantação	Cada região compatível: 10	Sim	Número máximo de grupos do Amazon EC2 Auto Scaling em um grupo de implantação
Implantações simultâneas por conta	Cada região compatível: 1.300	Sim	Número máximo de implantações simultâneas associadas a uma AWS conta. Cada implantação em uma instância ampliada do Amazon EC2 em um grupo do Amazon EC2 Auto Scaling conta como uma única implantação simultânea para cada aplicativo ao qual a instância do EC2 está associada.

Nome	Padrão	Ajustá	Description
Implantações simultâneas por grupo de implantação	Cada região compatível: 1	Não	Número máximo de implantações simultâneas em um grupo de implantação. Esse limite impede implantações simultâneas do mesmo aplicativo no mesmo grupo de implantação.
Configurações de implantação personalizadas por conta	Cada região com suporte: 50	Não	Número máximo de configurações de implantação personalizadas associadas a uma conta AWS
Grupos de implantação associados a um único aplicativo	Cada região com suporte: 1.000	Sim	O número máximo de grupos de implantação associados a um único aplicativo
EC2/On-Premises blue/green implantação executada em horas	Cada região compatível: 109	Não	Número máximo de horas que uma EC2/On-Premises blue/green implantação pode ser executada (48 horas para cada um dos dois limites acima, mais uma hora para cada um dos 13 possíveis eventos do ciclo de vida)
EC2/On-Premises a implantação local é executada em horas	Cada região compatível: 12	Não	Número máximo de horas que uma implantação EC2/On-Premises local pode ser executada

Nome	Padrão	Ajustá	Description
Gatilhos de notificação de eventos em um grupo de implantação	Cada região com suporte: 10	Sim	O número máximo de acionadores de notificações de evento em um grupo de implementação
GitHub tokens de conexão por conta	Cada região compatível: 25	Não	Número máximo de tokens de GitHub conexão para uma única AWS conta
Horas entre a conclusão de uma implantação e o encerramento das instâncias originais durante uma EC2/On-Premises blue/green implantação	Cada região compatível: 48	Não	Número máximo de horas entre a conclusão de uma implantação e o encerramento das instâncias originais durante uma EC2/On-Premises blue/green implantação
Horas entre a implantação de uma revisão e o momento em que o tráfego é transferido para as instâncias substitutas durante uma EC2/On-Premises blue/green implantação	Cada região compatível: 48	Não	Número máximo de horas entre a implantação de uma revisão e o momento em que o tráfego é transferido para as instâncias substitutas durante uma EC2/On-Premises blue/green implantação

Nome	Padrão	Ajustá	Description
Contagem de instâncias por implantação	us-east-1: 2.000 us-east-2:2.000 us-west-2: 2.000 eu-central-1:2.000 Cada uma das outras regiões compatíveis: 1.000	Sim	O número máximo de instâncias em uma única implantação
Minutos que uma blue/green implantação pode esperar após uma implantação bem-sucedida antes de encerrar instâncias da implantação original	Cada região compatível: 2.800	Não	Número máximo de minutos que uma blue/green implantação pode esperar após uma implantação bem-sucedida antes de encerrar instâncias da implantação original
Minutos entre a primeira e a última mudança de tráfego durante uma implantação linear ou canária do AWS Lambda	Cada região compatível: 2.880	Não	Número máximo de minutos entre a primeira e a última mudança de tráfego durante uma implantação linear ou canária do AWS Lambda

Nome	Padrão	Ajustá	Description
Minutos até que uma implantação falhe se um evento de ciclo de vida não for iniciado	Cada região compatível: 5	Não	Número máximo de minutos até uma implantação falhar se um evento de ciclo de vida não começar após (1) uma implantação ser acionada usando o console ou o comando de criação e implantação da AWS CLI ou (2) a conclusão do evento de ciclo de vida anterior.
Número de grupos de implantação que podem ser associados a um serviço Amazon ECS	Cada região compatível: 1	Não	Número máximo de grupos de implantação que podem ser associados a um serviço Amazon ECS
Número de instâncias que podem ser passadas para a ação BatchGetOnPremisesInstances da API	Cada região compatível: 100	Não	Número máximo de instâncias que podem ser passadas para a ação BatchGetOnPremisesInstances da API
Número de instâncias usadas por implantações simultâneas em andamento por conta	us-east-1: 3.000 Cada uma das outras regiões compatíveis: 1.000	Sim	O número máximo de instâncias que pode ser usado por implantações simultâneas que estão em andamento e associadas a uma conta

Nome	Padrão	Ajustá	Description
Número de listeners para uma rota de tráfego durante uma implantação do Amazon ECS	Cada região com suporte: 1	Não	Número máximo de listeners para uma rota de tráfego durante uma implantação do Amazon ECS
Segundos até que um evento do ciclo de vida de implantação falhe se não for concluído	Cada região compatível: 3.600 por segundo	Não	Número máximo de segundos até que um evento de ciclo de vida de implantação falhe se não for concluído
Tamanho do nome do grupo de implantação	Cada região com suporte: 100	Não	Número máximo de caracteres em um nome de grupo de implantação
Tamanho da chave da tag	Cada região compatível: 128	Não	Número máximo de caracteres em uma chave de tag
Tamanho do valor da tag	Cada região compatível: 256	Não	Número máximo de caracteres em um valor de tag
Tags em um grupo de implantação	Cada região com suporte: 10	Não	Número máximo de tags em um grupo de implantação
Tráfego que pode ser transferido em um incremento durante uma implantação do AWS Lambda	Cada região compatível: 99	Não	Porcentagem máxima de tráfego que pode ser transferida em um incremento durante uma implantação do AWS Lambda

Solução de problemas CodeDeploy

Use os tópicos desta seção para ajudar a resolver problemas e erros que você possa encontrar ao usar CodeDeploy.

> **Note**
>
> Você pode identificar as causas de muitas falhas de implantação por meio da revisão dos arquivos de log criados durante o processo de implantação. Para simplificar, recomendamos usar o Amazon CloudWatch Logs para monitorar centralmente os arquivos de log em vez de visualizá-los instância por instância. Para mais informações, consulte Monitoring Deployments with Amazon CloudWatch Tools.

Tópicos

- Tópicos gerais de solução de problemas
- Solucionar problemas de EC2/On-Premises implantação
- Solucionar problemas de implantação do Amazon ECS
- Solução de problemas AWS Problemas de implantação do Lambda
- Solução de problemas de grupo de implantação
- Solução de problemas de instância
- Solucionar problemas de GitHub token
- Solucionar problemas do Amazon EC2 Auto Scaling
- Códigos de erro para AWS CodeDeploy

Tópicos gerais de solução de problemas

Tópicos

- Lista de verificação geral para solução de problemas
- CodeDeploy os recursos de implantação são suportados apenas em alguns AWS Regiões
- Os procedimentos deste guia não coincidem com o CodeDeploy console
- Perfis do IAM necessários não estão disponíveis

- Usar alguns editores de texto para criar AppSpec arquivos e scripts de shell pode causar falhas nas implantações
- O uso do Finder no macOS para agrupar uma revisão de aplicativo pode causar falhas na implantação

Lista de verificação geral para solução de problemas

Você pode usar a seguinte lista de verificação para solucionar problemas com uma implantação que falhou.

1. Consulte Exibir detalhes CodeDeploy da implantação e View Instance Details para determinar por que a implantação falhou. Se você não conseguir determinar a causa, examine os itens desta lista de verificação.
2. Verifique se você configurou corretamente as instâncias:
   - A instância foi iniciada com um par de chaves do EC2 especificado? Para acessar mais informações, consulte EC2 Key Pairs no Guia do usuário do Amazon EC2.
   - O perfil de instância do IAM correto está anexado à instância? Para obter mais informações, consulte Configurar uma instância do Amazon EC2 para trabalhar com CodeDeploy e Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2.
   - A instância foi marcada? Para acessar mais informações, consulte Working with tags in the console no Guia do usuário do Amazon EC2.
   - O CodeDeploy agente está instalado, atualizado e em execução na instância? Para obter mais informações, consulte Gerenciando as operações CodeDeploy do agente. Para verificar qual versão do agente está instalada, consulte Determine a versão do CodeDeploy agente.
3. Verifique as configurações do aplicativo e do grupo de implantação:
   - Para verificar as configurações do seu aplicativo, consulte Veja os detalhes do aplicativo com CodeDeploy.
   - Para verificar as configurações do seu grupo de implantação, consulte Veja os detalhes do grupo de implantação com CodeDeploy.
4. Confirme se a revisão de aplicativo está corretamente configurada:
   - Verifique o formato do seu AppSpec arquivo. Para obter mais informações, consulte Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy e CodeDeploy AppSpec referência de arquivo.

- Verifique seu bucket ou GitHub repositório do Amazon S3 para verificar se a revisão do aplicativo está no local esperado.
- Revise os detalhes da revisão da sua CodeDeploy inscrição para garantir que ela esteja registrada corretamente. Para mais informações, consulte Visualize os detalhes da revisão do aplicativo com CodeDeploy.
- Se você estiver implantando a partir do Amazon S3, verifique seu bucket do Amazon S3 para CodeDeploy verificar se recebeu permissões para baixar a revisão do aplicativo. Para obter informações sobre políticas de bucket, consulte Pré-requisitos de implantação.
- Se você estiver implantando a partir de GitHub, verifique seu GitHub repositório para verificar se foram CodeDeploy concedidas permissões para baixar a revisão do aplicativo. Para obter mais informações, consulte Crie uma implantação com CodeDeploy e GitHub autenticação com aplicativos em CodeDeploy.

5. Verifique se a função de serviço está configurada corretamente. Para mais informações, consulte Etapa 2: criar uma função de serviço para CodeDeploy.
6. Confirme que você seguiu as etapas em Começando com CodeDeploy para:
   - Provisionou um usuário com as permissões apropriadas.
   - Instale ou atualize e configure a AWS CLI.
   - Crie um perfil de instância do IAM e um perfil de serviço.

   Para obter mais informações, consulte Gerenciamento de identidade e acesso para AWS CodeDeploy.
7. Confirme se você está usando a AWS CLI versão 1.6.1 ou posterior. Para verificar a versão instalada, chame aws --version.

Se você ainda não consegue solucionar sua implantação com falha, reveja os outros problemas neste tópico.

CodeDeploy os recursos de implantação são suportados apenas em alguns AWS Regiões

Se você não vê ou não consegue acessar aplicativos, grupos de implantação, instâncias ou outros recursos de implantação a partir do console AWS CLI ou do CodeDeploy console, certifique-se de fazer referência a uma das AWS regiões listadas em Região e endpoints em. Referência geral da AWS

As instâncias do EC2 e os grupos do Amazon EC2 Auto Scaling que são usados CodeDeploy em implantações devem ser lançados e criados em uma dessas regiões. AWS

Se você estiver usando o AWS CLI, execute o aws configure comando a partir do AWS CLI. Em seguida, você pode visualizar e definir sua AWS região padrão.

Se você estiver usando o CodeDeploy console, na barra de navegação, no seletor de regiões, escolha uma das AWS regiões compatíveis.

Important

Para usar serviços na região China (Pequim) ou China (Ningxia), é necessário ter uma conta e credenciais específicas para essas regiões. As contas e credenciais de outras AWS regiões não funcionam para as regiões de Pequim e Ningxia e vice-versa.
Informações sobre alguns recursos para as regiões da China, como nomes de buckets do CodeDeploy Resource Kit e procedimentos de instalação de CodeDeploy agentes, não estão incluídas nesta edição do Guia CodeDeploy do usuário.
Para obter mais informações:

- CodeDeployem Primeiros passos AWS na região da China (Pequim)
- CodeDeploy Guia do usuário para as regiões da China (versão em inglês | versão em chinês)

Os procedimentos deste guia não coincidem com o CodeDeploy console

Os procedimentos deste guia são gravados para refletir o novo design do console. Caso você esteja usando a versão mais antiga do console, muitos dos conceitos e dos procedimentos básicos deste guia ainda são aplicáveis. Para acessar a ajuda no novo console, escolha o ícone de informações.

Perfis do IAM necessários não estão disponíveis

Se você depende de um perfil de instância do IAM ou de uma função de serviço criada como parte de uma AWS CloudFormation pilha, se você excluir a pilha, todas as funções do IAM também serão excluídas. Talvez seja por isso que a função do IAM não é mais exibida no console do IAM e CodeDeploy não funciona mais conforme o esperado. Para corrigir esse problema, é necessário recriar manualmente o perfil do IAM excluído.

Usar alguns editores de texto para criar AppSpec arquivos e scripts de shell pode causar falhas nas implantações

Alguns editores de texto introduzem caracteres não imprimíveis e fora de conformidade nos arquivos. Se você usar editores de texto para criar ou modificar AppSpec arquivos ou arquivos de script de shell para execução em instâncias Amazon Linux, Ubuntu Server ou RHEL, qualquer implantação que dependa desses arquivos poderá falhar. Ao CodeDeploy usar esses arquivos durante uma implantação, a presença desses caracteres pode levar a falhas de validação de AppSpec arquivos difíceis de solucionar e falhas na execução de scripts.

No CodeDeploy console, na página de detalhes do evento da implantação, escolha Exibir registros. (Ou você usa o AWS CLI para chamar o comando get-deployment-instance.) Procure erros como `invalid character`, `command not found` ou `file not found`.

Para resolver este problema, recomendamos o seguinte:

- Não use editores de texto que introduzam caracteres não imprimíveis, como retornos de carro (`^M`caracteres) em seus AppSpec arquivos e arquivos de script de shell.
- Use editores de texto que exibam caracteres não imprimíveis, como retornos de carro em seus AppSpec arquivos e arquivos de script de shell, para que você possa encontrar e remover qualquer um que possa ser introduzido. Para ver exemplos desses tipos de editores de texto, pesquise os editores de texto que mostram retornos de carro na Internet.
- Use editores de texto executados em instâncias do Amazon Linux, Ubuntu Server ou RHEL para criar arquivos de script de shell que são executados em instâncias do Amazon Linux, Ubuntu Server ou RHEL. Para ver exemplos desses tipos de editores de texto, pesquise os editores de script de shell Linux na Internet.
- Se você precisa usar um editor de texto no Windows ou no macOS para criar arquivos de script de shell para execução em instâncias Amazon Linux, Ubuntu Server ou RHEL, use um programa ou utilitário que converta texto de formato Windows ou macOS no formato Unix. Para obter exemplos desses programas e utilitários, pesquise DOS para UNIX ou Mac para UNIX na Internet. Certifique-se de testar os arquivos de script de shell convertidos nos sistemas operacionais de destino.

O uso do Finder no macOS para agrupar uma revisão de aplicativo pode causar falhas na implantação

As implantações podem falhar se você usar o aplicativo de interface gráfica do usuário (GUI) do Finder em um Mac para agrupar (zip) um AppSpec arquivo e arquivos e scripts relacionados em um arquivo de revisão do aplicativo (.zip). Isso acontece porque o Finder cria uma pasta `__MACOSX` intermediária no arquivo .zip e coloca arquivos de componentes nessa pasta. Como o CodeDeploy não consegue encontrar os arquivos de componentes, a implantação falha.

Para resolver esse problema, recomendamos que você use o comando AWS CLI to call the push, que compacta os arquivos do componente na estrutura esperada. Como alternativa, é possível usar o Terminal em vez da GUI para compactar arquivos componentes. O terminal não cria uma pasta `__MACOSX` intermediária.

Solucionar problemas de EC2/On-Premises implantação

Tópicos

- CodeDeploy erro de credenciais CommandPoller ausentes do plugin
- A implantação falha com a mensagem “Falha na validação da mensagem assinada pelo PKCS7”
- A implantação ou reimplantação dos mesmos arquivos nos mesmos locais de instância falha com um erro indicando que a implantação falhou porque um arquivo especificado já existe na localização
- Caminhos de arquivo longos causam erros do tipo “Nenhum arquivo ou diretório”
- Long-running processos podem fazer com que as implantações falhem
- Solução de problemas em um evento de AllowTraffic ciclo de vida com falha sem nenhum erro relatado nos registros de implantação
- Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação
- Solução de problemas em um evento DownloadBundle de ciclo de vida de implantação com falha com UnknownError: não aberto para leitura
- Solução de problemas de erros relacionados a todos os eventos de ciclo de vida ignorados
- PowerShell Os scripts do Windows não usam a versão de 64 bits do Windows PowerShell por padrão

> **Note**
>
> As causas de muitas falhas de implantação podem ser identificadas por meio da revisão dos arquivos de log criados durante o processo de implantação. Para simplificar, recomendamos usar o Amazon CloudWatch Logs para monitorar centralmente os arquivos de log em vez de visualizá-los instância por instância. Para obter mais informações, consulte Exibir CodeDeploy registros no console de CloudWatch registros.

> **Tip**
>
> Para obter um runbook que automatiza muitas tarefas de solução de problemas relacionadas às EC2/On-Premises implantações, consulte AWSSupport-TroubleshootCodeDeploya referência do runbook do AWS Systems Manager Automation.

CodeDeploy erro de credenciais CommandPoller ausentes do plugin

Se você receber um erro semelhante a `InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile`, a causa pode ser uma das seguintes:

- A instância que você está implantando não tem um perfil de instância do IAM associado a ela.
- Seu perfil de instância do IAM não tem as permissões corretas configuradas.

Um perfil de instância do IAM concede ao CodeDeploy agente permissão para se comunicar CodeDeploy e baixar sua revisão do Amazon S3. Para instâncias do EC2, consulte Gerenciamento de identidade e acesso para AWS CodeDeploy. Para instâncias locais, consulte Working with On-Premises Instances.

A implantação falha com a mensagem “Falha na validação da mensagem assinada pelo PKCS7”

Essa mensagem de erro indica que a instância está executando uma versão do CodeDeploy agente que suporta somente o algoritmo de SHA-1 hash. O suporte para o algoritmo de SHA-2 hash foi introduzido na versão 1.0.1.854 do CodeDeploy agente, lançada em novembro de 2015. A partir de

17 de outubro de 2016, as implantações falharão se uma versão do CodeDeploy agente anterior à 1.0.1.854 for instalada. Para obter mais informações, consulte AWS para mudar para o algoritmo de hash SHA256 para certificados SSL, AVISO: Desativação de agentes de CodeDeploy host anteriores à versão 1.0.1.85 e. Atualize o CodeDeploy agente

A implantação ou reimplantação dos mesmos arquivos nos mesmos locais de instância falha com um erro indicando que a implantação falhou porque um arquivo especificado já existe na localização

Quando CodeDeploy tenta implantar um arquivo em uma instância, mas já existe um arquivo com o mesmo nome no local de destino especificado, a implantação nessa instância pode falhar. Você pode receber a mensagem de erro “A implantação falhou porque um arquivo especificado já existe neste local:”*`location-name`*. Isso ocorre porque, durante cada implantação, o CodeDeploy primeiro exclui todos os arquivos da implantação anterior, que estão listados em um arquivo de log de limpeza. Se houver arquivos nas pastas de instalação de destino que não estejam listados nesse arquivo de limpeza, o CodeDeploy agente, por padrão, interpreta isso como um erro e falha na implantação.

> **Note**
>
> Nas instâncias do Amazon Linux, RHEL e Ubuntu Server, o arquivo de limpeza está localizado em `/opt/codedeploy-agent/deployment-root/deployment-instructions/`. Em instâncias do Windows Server, o local é `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`.

A maneira mais fácil de evitar esse erro é especificar uma opção diferente do comportamento padrão de reprovar a implantação. Para cada implantação, você pode optar por reprovar a implantação, substituir os arquivos não listados no arquivo de limpeza ou manter os arquivos que já estão na instância.

A opção de substituição é útil quando, por exemplo, você colocou manualmente um arquivo em uma instância após a última implantação, mas adicionou um arquivo com o mesmo nome à próxima revisão de aplicativo.

É possível escolher a opção de retenção para os arquivos colocados na instância que você deseja que façam parte da próxima implantação sem precisar adicioná-los ao pacote de revisão de aplicativo. A opção de retenção também é útil se os arquivos do aplicativo já estiverem em seu

ambiente de produção e você quiser implantá-los usando CodeDeploy pela primeira vez. Para obter mais informações, consulte Crie uma implantação da EC2/On-Premises Compute Platform (console) e Comportamento de reversão com conteúdo existente.

Solução de problemas **`A implantação falhou porque um arquivo especificado já existe neste local`** (erros de implantação)

Se você optar por não especificar uma opção para substituir ou manter o conteúdo detectado pelo CodeDeploy nos seus locais de implantação de destino (ou se não especificar nenhuma opção de implantação para lidar com conteúdo existente em um comando programático), será possível optar por solucionar o erro.

As informações a seguir aplicam-se somente quando você opta por não reter ou substituir o conteúdo.

Se você tentar reimplantar arquivos com os mesmos nomes e locais, é mais provável que a reimplantação seja bem-sucedida se você especificar o nome do aplicativo e o grupo de implantação com o mesmo ID do grupo de implantação subjacente que você usou antes. CodeDeploy usa o ID do grupo de implantação subjacente para identificar os arquivos a serem removidos antes de uma reimplantação.

A implantação de novos arquivos ou a reimplantação dos mesmos arquivos nas mesmas localizações em instâncias pode falhar por estes motivos:

- Você especificou um nome de aplicativo diferente para uma redistribuição da mesma revisão nas mesmas instâncias. A redistribuição falha porque, mesmo que o nome do grupo de implantação seja o mesmo, o uso de um nome de aplicativo diferente significa que um ID de grupo de implantação subjacente diferente está sendo usado.
- Você excluiu e recriou um grupo de implantação para um aplicativo e depois tentou reimplantar a mesma revisão no grupo de implantação. A reimplantação falha porque, mesmo que o nome do grupo de implantação seja o mesmo, CodeDeploy faz referência a um ID de grupo de implantação subjacente diferente.
- Você excluiu um aplicativo e um grupo de implantação em e CodeDeploy, em seguida, criou um novo aplicativo e grupo de implantação com os mesmos nomes dos que você excluiu. Depois disso, você tentou reimplantar uma revisão que havia sido implantada no grupo de implantação anterior no novo grupo de implantação com o mesmo nome. A reimplantação falha porque, embora os nomes do aplicativo e do grupo de implantação sejam os mesmos, CodeDeploy ainda fazem referência ao ID do grupo de implantação que você excluiu.

- Você implantou uma revisão em um grupo de implantação e, em seguida, implantou a mesma revisão em outro grupo de implantação para as mesmas instâncias. A segunda implantação falha porque o CodeDeploy faz referência a um ID de grupo de implantação subjacente diferente.
- Você implantou uma revisão em um grupo de implantação e, em seguida, implantou outra revisão em outro grupo de implantação para as mesmas instâncias. Existe pelo menos um arquivo com o mesmo nome e no mesmo local em que o segundo grupo de implantação está tentando implantar. A segunda implantação falha porque CodeDeploy não remove o arquivo existente antes do início da segunda implantação. Ambas as implantações fazem referência a diferentes IDs de grupos de implantação.
- Você implantou uma revisão em CodeDeploy, mas há pelo menos um arquivo com o mesmo nome e no mesmo local. A implantação falha porque, por padrão, CodeDeploy não remove o arquivo existente antes do início da implantação.

Para resolver essas situações, siga um destes procedimentos:

- Remova os arquivos das localizações e instâncias em que eles foram implantados anteriormente e, em seguida, tente de novo a implantação.
- No AppSpec arquivo da sua revisão, nos eventos do ciclo de vida da BeforeInstall implantação ApplicationStop ou nos eventos do ciclo de vida da implantação, especifique um script personalizado para excluir arquivos em qualquer local que corresponda aos arquivos que sua revisão está prestes a instalar.
- Implante ou reimplante os arquivos em localizações ou instâncias que não faziam parte de implementações anteriores.
- Antes de excluir um aplicativo ou um grupo de implantação, implante uma revisão que contenha um AppSpec arquivo que especifique que nenhum arquivo deve ser copiado para as instâncias. Para a implantação, especifique o nome do aplicativo e o nome do grupo de implantação que usam os mesmos IDs de aplicativo e grupo de implantação subjacentes que aqueles que você está prestes a excluir. (Você pode usar o comando get-deployment-group para recuperar o ID do grupo de implantação.) CodeDeployusa o ID e o AppSpec arquivo do grupo de implantação subjacentes para remover todos os arquivos instalados na implantação anterior bem-sucedida.

Caminhos de arquivo longos causam erros do tipo “Nenhum arquivo ou diretório”

Para implantações em instâncias do Windows, se você tiver um caminho de arquivo com mais de 260 caracteres na seção de arquivos do seu arquivo appspec.yml, talvez as implantações falhem com um erro semelhante ao seguinte:

`No such file or directory @ dir_s_mkdir - C:\`*`your-long-file-path`*

Esse erro ocorre porque, por padrão, o Windows não permite caminhos de arquivo maiores que 260 caracteres, conforme detalhado na documentação da Microsoft.

Para as versões 1.4.0 ou posteriores do CodeDeploy agente, você pode ativar caminhos de arquivo longos de duas maneiras, dependendo do processo de instalação do agente:

Se o CodeDeploy agente ainda não tiver sido instalado:

1. Na máquina em que você planeja instalar o CodeDeploy agente, habilite a chave de registro do `LongPathsEnabled` Windows usando este comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
          -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Instale o CodeDeploy agente. Para obter mais informações, consulte Instale o CodeDeploy agente.

Se o CodeDeploy agente já tiver sido instalado:

1. Na máquina do CodeDeploy agente, habilite a chave de registro do `LongPathsEnabled` Windows usando este comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Reinicie o CodeDeploy agente para que a alteração da chave do registro entre em vigor. Para reiniciar o agente, use o seguinte comando:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Long-running processos podem fazer com que as implantações falhem

Para implantações em instâncias Amazon Linux, Ubuntu Server e RHEL, se você tiver um script de implantação que inicia um processo de longa execução, CodeDeploy pode passar muito tempo esperando pelo evento do ciclo de vida da implantação e, em seguida, falhar na implantação. Isso ocorre porque, se o processo demorar mais que o tempo esperado para os processos de primeiro e segundo plano nesse evento, o CodeDeploy interromperá e reprovará a implantação, mesmo que o processo ainda esteja sendo executado conforme o esperado.

Por exemplo, uma revisão de aplicativo contém dois arquivos em sua raiz, `after-install.sh` e `sleep.sh`. Seu AppSpec arquivo contém as seguintes instruções:

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

O `after-install.sh` arquivo é executado durante o evento do ciclo de vida do AfterInstall aplicativo. Seu conteúdo é o seguinte:

```
#!/bin/bash
/tmp/sleep.sh
```

O arquivo `sleep.sh` contém o seguinte, que suspende a execução do programa por três minutos (180 segundos), simulando alguns processos de execução longa:

```
#!/bin/bash
sleep 180
```

Quando `after-install.sh` liga`sleep.sh`, `sleep.sh` inicia e funciona por três minutos (180 segundos), ou seja, dois minutos (120 segundos) após o tempo CodeDeploy esperado `sleep.sh` (e, por relação,`after-install.sh`) parar de funcionar. Após o tempo limite de um minuto (60 segundos), CodeDeploy interrompe e falha a implantação no evento do ciclo de vida do AfterInstall

aplicativo, mesmo que `sleep.sh` continue funcionando conforme o esperado. O seguinte erro é exibido:

`Script at specified location: after-install.sh failed to complete in 60 seconds`.

Você não pode simplesmente adicionar um E comercial (&) em `after-install.sh` para executar `sleep.sh` em segundo plano.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

Isso pode deixar a implantação em um estado pendente por até o período padrão de tempo limite do evento do ciclo de vida da implantação de uma hora, após o qual CodeDeploy interrompe e falha a implantação no evento do ciclo de vida do AfterInstall aplicativo, como antes.

Em`after-install.sh`, chame da `sleep.sh` seguinte forma, o que permite CodeDeploy continuar após o início da execução do processo:

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

Na chamada anterior, `sleep.sh` é o nome do processo que você deseja começar a executar em segundo plano, redirecionando stdout, stderr e stdin para `/dev/null`.

Solução de problemas em um evento de AllowTraffic ciclo de vida com falha sem nenhum erro relatado nos registros de implantação

Em alguns casos, uma blue/green implantação falha durante o evento do AllowTraffic ciclo de vida, mas os registros de implantação não indicam a causa da falha.

Essa falha geralmente é devido a verificações de integridade que são configuradas incorretamente no Elastic Load Balancing para o Classic Load Balancer, Application Load Balancer ou Network Load Balancer usado para gerenciar o tráfego do grupo de implantação.

Para resolver o problema, revise e corrija quaisquer erros na configuração de verificação de integridade do load balancer.

Para Classic Load Balancers, consulte Configurar Health Checks no Guia do usuário para Classic Load Balancers e ConfigureHealthCheckna versão 2012-06-01 de referência da API Elastic Load Balancing.

Para Application Load Balancer, consulte Verificações de integridade de grupos de destino no Manual do usuário de Application Load Balancers.

Para Network Load Balancer, consulte Verificações de integridade de grupos de destino no Manual do usuário de Network Load Balancers.

Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação

Durante uma implantação, o CodeDeploy agente executa os scripts especificados para ApplicationStop BeforeBlockTraffic, e AfterBlockTraffic no AppSpec arquivo da implantação anterior bem-sucedida. (Todos os outros scripts são executados a partir do AppSpec arquivo na implantação atual.) Se um desses scripts contiver um erro e não for executado com sucesso, a implantação poderá falhar.

Os possíveis motivos dessas falhas incluem:

- O CodeDeploy agente encontra o *deployment-group-id*_last_successful_install arquivo no local correto, mas o local listado no *deployment-group-id*_last_successful_install arquivo não existe.

  Em instâncias do Amazon Linux, do Ubuntu Server e do RHEL, o arquivo deve existir em `/opt/codedeploy-agent/deployment-root/deployment-instructions`.

  Em instâncias do Windows Server, esse arquivo deve ser armazenado na pasta `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`.
- No local listado no *deployment-group-id*_last_successful_install arquivo, o AppSpec arquivo é inválido ou os scripts não são executados com êxito.
- O script contém um erro que não pode ser corrigido e, portanto, nunca é executado com êxito.

Use o CodeDeploy console para investigar por que uma implantação pode ter falhado durante qualquer um desses eventos. Na página de detalhes da implantação, escolha View events (Exibir eventos). Na página de detalhes da instância, na AfterBlockTrafficlinha

ApplicationStopBeforeBlockTraffic, ou, escolha Exibir registros. Ou use o AWS CLI para chamar o comando get-deployment-instance.

Se a causa da falha for um script da última implantação bem-sucedida que nunca foi executado com êxito, crie uma implantação e especifique que as AfterBlockTraffic falhas ApplicationStop BeforeBlockTraffic, e devem ser ignoradas. Há duas maneiras de fazer isso:

- Use o CodeDeploy console para criar uma implantação. Na página Criar implantação, em Falha no evento ApplicationStop do ciclo de vida, escolha Não falhar na implantação em uma instância se esse evento do ciclo de vida na instância falhar.
- Use o AWS CLI para chamar o create-deployment comando e incluir a `--ignore-application-stop-failures` opção.

Quando você implantar a revisão de aplicativo novamente, a implantação continuará mesmo se algum desses três eventos de ciclo de vida falhar. Se a nova revisão incluir scripts fixos para esses eventos de ciclo de vida, as implementações futuras poderão ser bem-sucedidas sem a aplicação dessa correção.

Solução de problemas em um evento DownloadBundle de ciclo de vida de implantação com falha com UnknownError: não aberto para leitura

Se você estiver tentando implantar uma revisão do aplicativo a partir do Amazon S3 e a implantação falhar durante o evento do ciclo de vida da DownloadBundle implantação com o erro: `UnknownError: not opened for reading`

- Houve um erro interno do serviço do Amazon S3. Implante novamente a revisão de aplicativo.
- O perfil de instância do IAM na sua instância do EC2 não tem permissões para acessar a revisão de aplicativo no Amazon S3. Para obter informações sobre políticas de bucket do Amazon S3, consulte Envie uma revisão CodeDeploy para o Amazon S3 (somente EC2/On-Premises implantações) e Pré-requisitos de implantação.
- As instâncias nas quais você implanta estão associadas a uma AWS região (por exemplo, Oeste dos EUA (Oregon)), mas o bucket do Amazon S3 que contém a revisão do aplicativo está associado a AWS outra região (por exemplo, Leste dos EUA (Norte da Virgínia)). Certifique-se de que a revisão do aplicativo esteja em um bucket do Amazon S3 associado à mesma AWS região das instâncias.

Na página de detalhes do evento da implantação, na linha Download bundle (Fazer download do pacote), escolha View logs (Exibir logs). Ou use o AWS CLI para chamar o comando get-deployment-instance. Se esse erro tiver ocorrido, deverá haver um erro na saída com o código `UnknownError` e a mensagem `not opened for reading`.

Para determinar o motivo desse erro:

1. Habilite o registro em log em pelo menos uma das instâncias e depois implante novamente a revisão de aplicativo.
2. Examine o arquivo de log para encontrar o erro. Mensagens de erro comuns para esse problema incluem a frase "acesso negado".
3. Depois de examinar os arquivos de log, recomendamos que você desabilite o registro em log para reduzir o tamanho do arquivo de log e a quantidade de informações confidenciais que poderão aparecer na saída em texto sem formatação na instância no futuro.

Para obter informações sobre como encontrar o arquivo de registro de conexões e ativar e desativar o registro de conexões, consulte `:log_aws_wire:` a referência de configuração do CodeDeploy agente.

Solução de problemas de erros relacionados a todos os eventos de ciclo de vida ignorados

Se todos os eventos de ciclo de vida de uma implantação do EC2 ou on-premises forem ignorados, você poderá receber um erro semelhante a `The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group are experiencing problems. (Error code: HEALTH_CONSTRAINTS)`. Aqui estão algumas causas e soluções possíveis:

- O CodeDeploy agente pode não estar instalado ou em execução na instância. Para determinar se o CodeDeploy agente está em execução:
  - Para um servidor Amazon Linux RHEL ou Ubuntu, execute o seguinte:

    ```
    systemctl status codedeploy-agent
    ```

  - Para o Windows, execute o seguinte:

```
powershell.exe -Command Get-Service -Name CodeDeployagent
```

Se o CodeDeploy agente não estiver instalado ou em execução, consulteVerifique se o CodeDeploy agente está em execução.

Talvez sua instância não consiga acessar o endpoint público CodeDeploy ou o Amazon S3 usando a porta 443. Faça uma das coisas a seguir:

- Atribua um endereço IP público à instância e use a respectiva tabela de rotas para permitir o acesso à Internet. Verifique se o grupo de segurança associado à instância permite o acesso de saída pela porta 443 (HTTPS). Para obter mais informações, consulte Protocolo de comunicação e porta para o CodeDeploy agente.
- Se uma instância for provisionada em uma sub-rede privada, use um gateway NAT em vez de um gateway da Internet na tabela de rotas. Para obter mais informações, consulte Gateways NAT.

- A função de serviço de CodeDeploy pode não ter as permissões necessárias. Para configurar uma função de serviço do CodeDeploy, consulte Etapa 2: criar uma função de serviço para CodeDeploy.
- Se você usa um proxy HTTP, verifique se ele está especificado na `:proxy_uri:` configuração do arquivo de configuração do CodeDeploy agente. Para obter mais informações, consulte CodeDeploy referência de configuração do agente.
- A assinatura de data e hora da sua instância de implantação pode não corresponder à assinatura de data e hora da sua solicitação de implantação. Procure um erro semelhante ao `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` do seu arquivo de log do CodeDeploy agente. Se você vir esse erro, siga as etapas em Solução de problemas de "InvalidSignatureException — A assinatura expirou: [hora] agora é anterior à [hora]" de erros de implantação. Para obter mais informações, consulte Exibir dados de registro para CodeDeploy EC2/On-Premises implantações.
- O CodeDeploy agente pode parar de ser executado porque a instância está com pouca memória ou espaço no disco rígido. Tente reduzir o número de implantações arquivadas na sua instância atualizando a `max_revisions` configuração na configuração do CodeDeploy agente. Se você fizer isso para uma instância do EC2 e o problema persistir, considere o uso de uma instância maior. Por exemplo, se o tipo de instância for `t2.small`, tente usar `t2.medium`. Para obter mais informações, consulte Arquivos instalados pelo CodeDeploy agente, CodeDeploy referência de configuração do agente e Tipos de instância.

- A instância na qual você está fazendo a implantação pode não ter um perfil de instância do IAM anexado, ou pode ter um perfil de instância do IAM anexado que não tenha as permissões necessárias.
  - Se não houver um perfil de instância do IAM anexado à sua instância, crie um com as permissões necessárias e anexe-o.
  - Se já houver um perfil de instância do IAM anexado à sua instância, verifique se ele tem as permissões necessárias.

  Depois de confirmar que o perfil de instância anexado está configurado com as permissões necessárias, reinicie a instância. Para obter mais informações, consulte Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2 e Perfis do IAM para o Amazon EC2 no Guia do usuário do Amazon EC2.

PowerShell Os scripts do Windows não usam a versão de 64 bits do Windows PowerShell por padrão

Se um PowerShell script do Windows executado como parte de uma implantação depender da funcionalidade de 64 bits (por exemplo, porque consome mais memória do que um aplicativo de 32 bits permite ou chama bibliotecas que são oferecidas somente em uma versão de 64 bits), o script pode falhar ou não ser executado conforme o esperado. Isso ocorre porque, por padrão, CodeDeploy usa a versão de 32 bits do Windows PowerShell para executar PowerShell scripts do Windows que fazem parte de uma revisão do aplicativo.

Adicione um código como o seguinte ao início de qualquer script que precise ser executado com a versão de 64 bits do Windows PowerShell:

```
# Are you running in 32-bit mode?
#   (\SysWOW64\ = 32-bit mode)

if ($PSHOME -like "*SysWOW64*")
{
  Write-Warning "Restarting this script under 64-bit Windows PowerShell."

  # Restart this script under 64-bit Windows PowerShell.
  #   (\SysNative\ redirects to \System32\ for 64-bit mode)

  & (Join-Path ($PSHOME -replace "SysWOW64", "SysNative") powershell.exe) -File `
    (Join-Path $PSScriptRoot $MyInvocation.MyCommand) @args
```

```
  # Exit 32-bit script.

  Exit $LastExitCode
}

# Was restart successful?
Write-Warning "Hello from $PSHOME"
Write-Warning "  (\SysWOW64\ = 32-bit mode, \System32\ = 64-bit mode)"
Write-Warning "Original arguments (if any): $args"

# Your 64-bit script code follows here...
# ...
```

Embora as informações do caminho do arquivo nesse código possam parecer contra-intuitivas, o Windows de 32 bits PowerShell usa um caminho como:

`c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`

O Windows de 64 bits PowerShell usa um caminho como:

`c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`

Solucionar problemas de implantação do Amazon ECS

Tópicos

- Ocorre um tempo limite enquanto se aguarda o conjunto de tarefas de substituição
- Ocorre um limite de tempo enquanto se aguarda a continuação de uma notificação
- O perfil do IAM não tem permissões suficientes
- A implantação atingiu o tempo limite enquanto aguardava um retorno de chamada de status
- A implantação falhou porque uma ou mais funções de validação de evento do ciclo de vida falharam
- Não foi possível atualizar o ELB devido ao seguinte erro: o grupo de destino do conjunto de tarefas primário deve estar atrás do receptor
- Às vezes, minha implantação falha ao usar o ajuste de escala automático
- Somente o ALB suporta roteamento de tráfego gradual; em vez disso, use o roteamento de AllAtOnce tráfego ao usar o grupo de implantação create/update
- Embora minha implantação tenha sido bem-sucedida, o conjunto de tarefas de substituição falha nas verificações de integridade do Elastic Load Balancing, e meu aplicativo está inativo

- Posso conectar vários balanceadores de carga a um grupo de implantação?
- Posso realizar CodeDeploy blue/green implantações sem um balanceador de carga?
- Como posso atualizar meu serviço Amazon ECS com novas informações durante uma implantação?

Ocorre um tempo limite enquanto se aguarda o conjunto de tarefas de substituição

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

```
The deployment timed out while waiting for the replacement task set to
become healthy. This time out period is 60 minutes.
```

Possível causa: esse erro pode ocorrer se houver um erro no arquivo de definição de tarefas ou em outros arquivos relacionados à implantação. Por exemplo, se houver um erro de digitação no campo `image` do seu arquivo de definição de tarefa, o Amazon ECS tentará extrair a imagem errada do contêiner e falhará continuamente, causando esse erro.

Possíveis correções e próximas etapas:

- Corrija erros tipográficos e problemas de configuração no arquivo de definição de tarefas e em outros arquivos.
- Confira o evento relacionado ao serviço Amazon ECS e descubra por que as tarefas de substituição não estão ficando íntegras. Para mais informações sobre eventos do Amazon ECS, consulte Eventos do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.
- Consulte a seção de Solução de problemas do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service para ver se há erros relacionados às mensagens do evento.

Ocorre um limite de tempo enquanto se aguarda a continuação de uma notificação

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

`The deployment timed out while waiting for a notification to continue. This time out period is n minutes.`

Possível causa: esse erro pode ocorrer se você especificou um tempo de espera no campo Especificar quando redirecionar o tráfego ao criar seu grupo de implantação, mas a implantação não pôde ser concluída antes que o tempo de espera expirasse.

Possíveis correções e próximas etapas:

- Em seu grupo de implantação, defina Especificar quando redirecionar o tráfego para um período maior e reimplante. Para obter mais informações, consulte Criar um grupo de implantação para uma implantação do Amazon ECS (console).
- Em seu grupo de implantação, altere Especificar quando redirecionar o tráfego para Redirecionar o tráfego imediatamente e reimplante. Para obter mais informações, consulte Criar um grupo de implantação para uma implantação do Amazon ECS (console).
- Reimplante e, em seguida, execute o AWS CLI comando aws deploy continue-deployment com a `--deployment-wait-type` opção definida como. `READY_WAIT` Certifique-se de executar esse comando antes que o horário especificado em Especificar quando redirecionar o tráfego expire.

O perfil do IAM não tem permissões suficientes

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

`The IAM role role-arn does not give you permission to perform operations in the following AWS service: AWSLambda.`

Causa possível: esse erro pode ocorrer se você especificou uma função do Lambda na `Hooks` seção do AppSpec arquivo, mas não deu CodeDeploy permissão ao serviço do Lambda.

Possível correção: adicione a `lambda:InvokeFunction` permissão à função CodeDeploy de serviço. Para adicionar essa permissão, adicione uma das seguintes políticas gerenciadas pela AWS ao perfil: **`AWSCodeDeployRoleForECS`** ou **`AWSCodeDeployRoleForECSLimited`**. Para obter informações sobre essas políticas e como adicioná-las à função CodeDeploy de serviço, consulteEtapa 2: criar uma função de serviço para CodeDeploy.

A implantação atingiu o tempo limite enquanto aguardava um retorno de chamada de status

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

```
The deployment timed out while waiting for a status callback. CodeDeploy expects a status callback within one hour after a deployment hook is invoked.
```

Causa possível: esse erro pode ocorrer se você especificou uma função do Lambda na `Hooks` seção do AppSpec arquivo, mas a função do Lambda não conseguiu chamar a `PutLifecycleEventHookExecutionStatus` API necessária para retornar um `Failed` status `Succeeded` or. CodeDeploy

Possíveis correções e próximas etapas:

- Adicione a `codedeploy:putlifecycleEventHookExecutionStatus` permissão à função de execução do Lambda usada pela função do Lambda que você especificou no AppSpec arquivo. Essa permissão concede à função Lambda a capacidade de retornar um status de `Succeeded` ou `Failed` para CodeDeploy. Para obter informações sobre funções de execução do Lambda, consulte Função de execução do Lambda no Guia do usuário do AWS Lambda .
- Verifique o código da função Lambda e os registros de execução para garantir que sua função CodeDeploy do Lambda esteja chamando a `PutLifecycleEventHookExecutionStatus` API para CodeDeploy informar se o teste de validação do ciclo de vida ou. `Succeeded` `Failed` Para obter informações sobre a `putlifecycleEventHookExecutionStatus` API, consulte PutLifecycleEventHookExecutionStatus a Referência AWS CodeDeploy da API. Para obter informações sobre os registros de execução do Lambda, consulte Acessando CloudWatch os registros da Amazon para AWS Lambda.

A implantação falhou porque uma ou mais funções de validação de evento do ciclo de vida falharam

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

```
The deployment failed because one or more of the lifecycle event validation functions failed.
```

Causa possível: esse erro pode ocorrer se você especificou uma função do Lambda na `Hooks` seção do AppSpec arquivo, mas a função do Lambda retornou `Failed` CodeDeploy quando foi chamada. `PutLifecycleEventHookExecutionStatus` Essa falha indica CodeDeploy que o teste de validação do ciclo de vida falhou.

Próxima etapa possível: verifique seus logs de execução do Lambda para ver por que o código do teste de validação está falhando. Para obter informações sobre os registros de execução do Lambda, consulte Acessando CloudWatch os registros da Amazon para AWS Lambda.

Não foi possível atualizar o ELB devido ao seguinte erro: o grupo de destino do conjunto de tarefas primário deve estar atrás do receptor

Problema: Você vê a seguinte mensagem de erro ao implantar seu aplicativo Amazon ECS usando: CodeDeploy

```
The ELB could not be updated due to the following error: Primary taskset
target group must be behind listener
```

Possível causa: esse erro pode ocorrer se você tiver configurado um receptor de teste opcional, e ele estiver configurado com o grupo de destino errado. Para obter mais informações sobre o ouvinte de teste em CodeDeploy, consulte Antes de começar uma implantação do e. O que acontece durante uma implantação do Para obter mais informações sobre conjuntos de tarefas, consulte TaskSet a Referência da API do Amazon Elastic Container Service e descreva-task-set na seção Amazon ECS da Referência de Comandos. AWS CLI

Possível correção: certifique-se de que o receptor de produção e o receptor de teste do Elastic Load Balancing estejam apontando para o grupo de destino que atualmente atende aos seus workloads. Há três lugares para verificar:

- No Amazon EC2, nas configurações de Receptores e regras do seu balanceador de carga. Para obter mais informações, consulte Receptores para o seu Application Load Balancer, no Guia do usuário do Application Load Balancers, e Receptores para o seu Network Load Balancer ,no Guia do usuário do Network Load Balancer.
- No Amazon ECS, no seu cluster, na configuração de Rede do seu serviço. Para obter mais informações, consulte as considerações sobre o Application Load Balancer e o Network Load Balancer no Guia do Desenvolvedor do Amazon Elastic Container Service.
- Em CodeDeploy, nas configurações do seu grupo de implantação. Para obter mais informações, consulte Criar um grupo de implantação para uma implantação do Amazon ECS (console).

Às vezes, minha implantação falha ao usar o ajuste de escala automático

Problema: Você está usando o Auto Scaling com CodeDeploy e percebe que suas implantações ocasionalmente falham. Para obter mais informações sobre os sintomas desse problema, consulte o tópico que diz Para serviços configurados para usar o escalonamento automático de serviços e o tipo de blue/green implantação, o escalonamento automático não é bloqueado durante uma implantação, mas a implantação pode falhar em algumas circunstâncias no Guia do desenvolvedor do Amazon Elastic Container Service.

Causa possível: Esse problema pode ocorrer se CodeDeploy os processos do Auto Scaling entrarem em conflito.

Possível solução: suspenda e retome os processos de Auto Scaling durante a CodeDeploy implantação usando a `RegisterScalableTarget` API (ou o `register-scalable-target` AWS CLI comando correspondente). Para obter mais informações, consulte Suspender e retomar o ajuste de escala do ajuste de escala automático do aplicativo no Guia do usuário do ajuste de escala automático do aplicativo.

> **Note**
>
> CodeDeploy não posso ligar `RegisterScaleableTarget` diretamente. Para usar essa API, você deve configurar o CodeDeploy envio de uma notificação ou evento para o Amazon Simple Notification Service (ou Amazon CloudWatch). Em seguida, você deve configurar o Amazon SNS (or CloudWatch) para chamar uma função do Lambda e configurar a função do Lambda para chamar a API. `RegisterScalableTarget` A API `RegisterScalableTarget` deve ser chamada com o parâmetro `SuspendedState` definido como `true`, para suspender as operações de ajuste de escala automático, e como `false`, para retomá-las.
>
> A notificação ou evento CodeDeploy enviado deve ocorrer quando uma implantação começa (para acionar as operações de suspensão do Auto Scaling) ou quando uma implantação é bem-sucedida, falha ou é interrompida (para acionar as operações de retomada do Auto Scaling).
>
> Para obter informações sobre como configurar CodeDeploy para gerar notificações ou CloudWatch eventos do Amazon SNS, consulteMonitoramento de implantações com Amazon Events CloudWatch. e. Monitoring Deployments with Amazon SNS Event Notifications

Somente o ALB suporta roteamento de tráfego gradual; em vez disso, use o roteamento de AllAtOnce tráfego ao usar o grupo de implantação create/update

Problema: Você vê a seguinte mensagem de erro ao criar ou atualizar um grupo de implantação em CodeDeploy:

```
Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing
instead when you create/update Deployment group.
```

Possível causa: esse erro pode ocorrer se você estiver usando um Network Load Balancer e tentar usar uma configuração de implantação predefinida diferente de `CodeDeployDefault.ECSAllAtOnce`.

Correções possíveis:

- Altere sua configuração de implantação predefinida para `CodeDeployDefault.ECSAllAtOnce`. Essa é a única configuração de implantação predefinida permitida pelo Network Load Balancers.

  Para obter mais informações sobre configurações de implantação pré-definidas, consulte Configurações de implantação predefinidas para uma plataforma de computação do Amazon ECS.
- Mude seu balanceador de carga para um Application Load Balancer. O Application Load Balancer é compatível com todas as configurações de implantação predefinidas. Para obter mais informações sobre como criar um Application Load Balancer, consulte Configure um balanceador de carga, grupos-alvo e ouvintes para implantações do CodeDeploy Amazon ECS.

Embora minha implantação tenha sido bem-sucedida, o conjunto de tarefas de substituição falha nas verificações de integridade do Elastic Load Balancing, e meu aplicativo está inativo

Problema: Embora CodeDeploy indique que minha implantação foi bem-sucedida, o conjunto de tarefas de substituição falha nas verificações de integridade do Elastic Load Balancing e meu aplicativo está inativo.

Causa possível: Esse problema pode ocorrer se você executou uma implantação CodeDeploy completa de uma só vez e seu conjunto de tarefas substituto (verde) contiver um código incorreto que está causando a falha nas verificações de integridade do Elastic Load Balancing. Com a

configuração de implantação completa, as verificações de integridade do balanceador de carga começam a ser executadas no conjunto de tarefas de substituição depois que o tráfego é transferido para ele (ou seja, após a ocorrência do evento CodeDeploy do `AllowTraffic` ciclo de vida). É por isso que você verá falhas nas verificações de integridade na tarefa de substituição definida após a mudança de tráfego, mas não antes. Para obter informações sobre os eventos do ciclo de vida que CodeDeploy geram, consulte. O que acontece durante uma implantação do

Correções possíveis:

- Altere sua configuração de implantação de tudo de uma vez para canário ou linear. Em uma configuração canária ou linear, as verificações de integridade do balanceador de carga começam a ser executadas no conjunto de tarefas de substituição enquanto CodeDeploy instala seu aplicativo no ambiente de substituição e antes que o tráfego seja alterado (ou seja, durante o evento do `Install` ciclo de vida e antes do evento). `AllowTraffic` Ao permitir que as verificações sejam executadas durante a instalação do aplicativo, mas antes que o tráfego seja transferido, um código incorreto do aplicativo será detectado e causará falhas na implantação antes que o aplicativo se torne disponível publicamente.

  Para obter informações sobre como configurar implantações canário ou lineares, consulte Altere as configurações do grupo de implantação com CodeDeploy.

  Para obter informações sobre eventos CodeDeploy do ciclo de vida que são executados durante uma implantação do Amazon ECS, consulte. O que acontece durante uma implantação do

  > **Note**
  >
  > As configurações de implantação canário e linear são compatíveis somente com Application Load Balancers.

- Se você quiser manter sua configuração de implantação completa, configure um receptor de teste e verifique o status de integridade do conjunto de tarefas de substituição com o gancho do ciclo de vida `BeforeAllowTraffic`. Para obter mais informações, consulte Lista de hooks do evento do ciclo de vida para uma implantação Amazon ECS.

Posso conectar vários balanceadores de carga a um grupo de implantação?

Não. Se você quiser usar vários Application Load Balancers ou Network Load Balancers, use as atualizações contínuas do Amazon ECS em vez de implantações. CodeDeploy blue/green Para obter mais informações sobre as atualizações contínuas, consulte Atualização contínua no Guia do desenvolvedor do Amazon Elastic Container Service. Para obter mais informações sobre o uso de vários balanceadores de carga com o Amazon ECS, consulte Registrar vários grupos de destino com um serviço no Guia do desenvolvedor do Amazon Elastic Container Service.

Posso realizar CodeDeploy blue/green implantações sem um balanceador de carga?

Não, você não pode realizar CodeDeploy blue/green implantações sem um balanceador de carga. Se você não conseguir usar um balanceador de carga, use o atributo de atualizações contínuas do Amazon ECS. Para obter mais informações sobre as atualizações contínuas do Amazon ECS, consulte Atualização contínua no Guia do desenvolvedor do Amazon Elastic Container Service.

Como posso atualizar meu serviço Amazon ECS com novas informações durante uma implantação?

Para CodeDeploy atualizar seu serviço Amazon ECS com um novo parâmetro enquanto ele conduz uma implantação, especifique o parâmetro na `resources` seção do AppSpec arquivo. Somente alguns parâmetros do Amazon ECS são suportados pelo CodeDeploy, como os parâmetros do arquivo de definição de tarefas e do nome do contêiner. Para obter uma lista completa dos parâmetros do Amazon ECS que CodeDeploy podem ser atualizados, consulteAppSpec seção 'recursos' para implantações do Amazon ECS.

> **Note**
>
> Se você precisar atualizar seu serviço Amazon ECS com um parâmetro que não seja suportado pelo CodeDeploy, conclua estas tarefas:
>
> 1. Chame a API `UpdateService` do Amazon ECS com o parâmetro que você deseja atualizar. Para obter uma lista completa de parâmetros que podem ser atualizados, consulte a Referência UpdateService de API do Amazon Elastic Container Service.

2. Para aplicar a alteração às tarefas, crie uma nova blue/green implantação do Amazon ECS. Para obter mais informações, consulte Criar uma implantação da plataforma de computação do Amazon ECS (console).

Solução de problemas AWS Problemas de implantação do Lambda

Tópicos

- AWS Lambda as implantações falham após interromper manualmente uma implantação do Lambda que não tem reversões configuradas

AWS Lambda as implantações falham após interromper manualmente uma implantação do Lambda que não tem reversões configuradas

Em alguns casos, o alias da função do Lambda especificada em uma implantação pode fazer referência a duas versões diferentes da função. O resultado é que as tentativas subsequentes de implantar a função do Lambda falham. A implantação do Lambda pode chegar a esse estado quando não há reversões configuradas e ela for interrompida manualmente. Para continuar, use o AWS Lambda console para garantir que a função não esteja configurada para mudar o tráfego entre duas versões:

1. Faça login no Console de gerenciamento da AWS e abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. No painel esquerdo, escolha Functions (Funções).
3. Selecione o nome da função Lambda que está em sua CodeDeploy implantação.
4. Em Aliases, escolha o alias usado em sua CodeDeploy implantação e, em seguida, escolha Editar.
5. Em Alias ponderado, escolha **none**. Isso garante que o alias não esteja configurado para deslocar uma porcentagem, ou peso, do tráfego para mais de uma versão. Anote a versão selecionada em Version (Versão).
6. Escolha Salvar.
7. Abra o CodeDeploy console e tente implantar a versão exibida no menu suspenso na etapa 5.

Solução de problemas de grupo de implantação

Marcar uma instância como parte de um grupo de implantação não implanta automaticamente seu aplicativo na nova instância

CodeDeploy não implanta automaticamente seu aplicativo em uma instância recém-marcada. Você deve criar uma nova implantação no grupo de implantação.

Você pode usar CodeDeploy para habilitar implantações automáticas em novas instâncias do EC2 nos grupos do Amazon EC2 Auto Scaling. Para obter mais informações, consulte Integração CodeDeploy com o Amazon EC2 Auto Scaling.

Solução de problemas de instância

Tópicos

- As tags devem ser definidas corretamente
- AWS CodeDeploy o agente deve estar instalado e em execução nas instâncias
- As implantações não falham por até uma hora quando uma instância é encerrada durante uma implantação
- Analisando arquivos de log para investigar falhas de implantação em instâncias
- Crie um novo arquivo de CodeDeploy log se ele tiver sido excluído acidentalmente
- Solução de problemas de "InvalidSignatureException — A assinatura expirou: [hora] agora é anterior à [hora]" de erros de implantação

As tags devem ser definidas corretamente

Use o comando list-deployment-instances para confirmar que as instâncias usadas para uma implantação estão marcadas corretamente. Se uma instância do EC2 estiver ausente na saída, use o console do EC2 para confirmar que as tags foram definidas nessa instância. Para acessar mais informações, consulte Working with tags in the console no Guia do usuário do Amazon EC2.

> **Note**
>
> Se você marcar uma instância e usá-la imediatamente CodeDeploy para implantar um aplicativo nela, a instância pode não ser incluída na implantação. Isso ocorre porque pode

levar alguns minutos até que CodeDeploy você possa ler as tags. Recomendamos que você aguarde pelo menos cinco minutos entre o momento de marcar uma instância e a tentativa de implantar nela.

AWS CodeDeploy o agente deve estar instalado e em execução nas instâncias

Para verificar se o CodeDeploy agente está instalado e em execução em uma instância, consulteVerifique se o CodeDeploy agente está em execução.

Para instalar, desinstalar ou reinstalar o CodeDeploy agente, consulteInstale o CodeDeploy agente.

As implantações não falham por até uma hora quando uma instância é encerrada durante uma implantação

CodeDeploy fornece uma janela de uma hora para que cada evento do ciclo de vida da implantação seja executado até a conclusão. Isso fornece tempo suficiente para scripts de execução longa.

Se os scripts não forem executados até a conclusão enquanto um evento de ciclo de vida estiver em andamento (por exemplo, se uma instância for encerrada ou o CodeDeploy agente for desligado), pode levar até uma hora para que o status da implantação seja exibido como Falha. Isso é verdade mesmo que o período de tempo limite especificado no script seja menor que uma hora. Isso ocorre porque, quando a instância é encerrada, o CodeDeploy agente é encerrado e não consegue processar mais scripts.

Porém, se uma instância for encerrada entre eventos de ciclo de vida ou antes do início da primeira etapa do evento de ciclo de vida, o tempo limite ocorrerá depois de apenas cinco minutos.

Analisando arquivos de log para investigar falhas de implantação em instâncias

Se o status de uma instância na implantação for algo diferente de `Succeeded`, você poderá analisar os dados do arquivo de log de implantação para ajudar a identificar o problema. Para obter informações sobre como acessar dados do log de implantação, consulte Exibir dados de registro para CodeDeploy EC2/On-Premises implantações.

Crie um novo arquivo de CodeDeploy log se ele tiver sido excluído acidentalmente

Se você excluir acidentalmente o arquivo de log de implantação em uma instância, CodeDeploy não cria um arquivo de log substituto. Para criar um novo arquivo de log, faça login na instância e depois execute estes comandos:

Para uma instância do Amazon Linux, Ubuntu Server ou RHEL, execute estes comandos nesta ordem, um de cada vez:

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Para uma instância do Windows Server:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Solução de problemas de "InvalidSignatureException — A assinatura expirou: [hora] agora é anterior à [hora]" de erros de implantação

CodeDeploy requer referências temporais precisas para realizar suas operações. Se a data e a hora da sua instância não estiverem definidas corretamente, elas podem não corresponder à data de assinatura da sua solicitação de implantação, que é CodeDeploy rejeitada.

Para evitar falhas de implantação relacionadas a configurações de tempo incorretas, consulte os seguintes tópicos:

- Definição da hora de sua instância do Linux
- Definição de horário para uma instância do Windows

Solucionar problemas de GitHub token

Token OAuth GitHub inválido

CodeDeploy os aplicativos criados após junho de 2017 usam tokens GitHub OAuth para cada AWS região. O uso de tokens vinculados a AWS regiões específicas oferece mais controle sobre quais CodeDeploy aplicativos têm acesso a um GitHub repositório.

Se você receber um erro de GitHub token, talvez tenha um token antigo que agora é inválido.

Para corrigir um token GitHub OAuth inválido

1. Remova o token antigo usando um dos seguintes métodos:
   - Para remover o token antigo usando a API, use DeleteGitHubAccountToken.
   - Para remover o token antigo usando o AWS Command Line Interface:
     a. Vá para o computador em que o token reside.
     b. Verifique se o AWS CLI está instalado neste computador. Para obter instruções de instalação, consulte Instalação, atualização e desinstalação da AWS CLI no Guia do usuário AWS Command Line Interface .
     c. Digite o seguinte comando no computador em que o token reside:

        ```
        aws delete-git-hub-account-token
        ```

        Para obter detalhes sobre a sintaxe do comando, consulte delete-git-hub-account-token.
2. Adicione um novo token OAuth. Para obter mais informações, consulte Integrando com CodeDeploy GitHub.

Número máximo de tokens GitHub OAuth excedido

Quando você cria uma CodeDeploy implantação, o número máximo de GitHub tokens permitidos é 10. Se você receber um erro sobre tokens GitHub OAuth, verifique se você tem 10 ou menos tokens. Se você tiver mais de 10 tokens, os primeiros tokens que foram criados serão considerados inválidos. Por exemplo, se você tiver 11 tokens, o primeiro token criado será inválido. Se tiver 12 tokens, os dois primeiros tokens que você criou serão inválidos. Para obter informações sobre como usar a CodeDeploy API para remover tokens antigos, consulte DeleteGitHubAccountToken.

Solucionar problemas do Amazon EC2 Auto Scaling

Tópicos

- Solução de problemas gerais do Amazon EC2 Auto Scaling
- “CodeDeployRole não lhe dá permissão para realizar operações nas seguintes AWS serviço: AmazonAutoScaling "erro
- As instâncias em um grupo de Amazon EC2 Auto Scaling são continuamente provisionadas e encerradas antes que uma revisão possa ser implantada
- O encerramento ou a reinicialização de uma instância do Amazon EC2 Auto Scaling pode causar falhas nas implantações
- Evite associar vários grupos de implantação a um único grupo de Amazon EC2 Auto Scaling
- As instâncias do EC2 em um grupo Amazon EC2 Auto Scaling falham ao ser iniciadas e recebem o erro "Heartbeat Timeout"
- Ganchos do ciclo de vida do Amazon EC2 Auto Scaling podem fazer com que as implantações automáticas em grupos do Amazon EC2 Auto Scaling parem ou falhem.
- Erro “A implantação falhou porque nenhuma instância foi encontrada para seu grupo de implantação”

Solução de problemas gerais do Amazon EC2 Auto Scaling

Implantações em instâncias do EC2 em um grupo de Amazon EC2 Auto Scaling podem falhar pelos seguintes motivos:

- O Amazon EC2 Auto Scaling inicia e termina continuamente instâncias do EC2. Se CodeDeploy não puder implantar automaticamente a revisão do seu aplicativo, o Amazon EC2 Auto Scaling inicia e encerra continuamente as instâncias do EC2.

  Desassocie o grupo Amazon EC2 Auto Scaling do grupo de implantação ou altere CodeDeploy a configuração do seu grupo Amazon EC2 Auto Scaling para que o número desejado de instâncias corresponda ao número atual de instâncias (evitando assim que o Amazon EC2 Auto Scaling lance mais instâncias do EC2). Para obter mais informações, consulte Altere as configurações do grupo de implantação com CodeDeploy ou Escalabilidade manual do Amazon EC2 Auto Scaling.
- O CodeDeploy agente não responde. O CodeDeploy agente pode não ser instalado se os scripts de inicialização (por exemplo, scripts cloud-init) executados imediatamente após a inicialização ou inicialização de uma instância do EC2 levarem mais de uma hora para serem

executados. CodeDeploy tem um tempo limite de uma hora para o CodeDeploy agente responder às implantações pendentes. Para resolver esse problema, mova seus scripts de inicialização para sua revisão de aplicativo CodeDeploy.

- Uma instância do EC2 em um grupo do Amazon EC2 Auto Scaling é reinicializada durante uma implantação. Sua implantação pode falhar se uma instância do EC2 for reinicializada durante uma implantação ou se o CodeDeploy agente for desligado durante o processamento de um comando de implantação. Para obter mais informações, consulte O encerramento ou a reinicialização de uma instância do Amazon EC2 Auto Scaling pode causar falhas nas implantações.
- Várias revisões de aplicativos são implantadas simultaneamente na mesma instância do EC2 em um grupo de Amazon EC2 Auto Scaling. A implantação de várias revisões de aplicativos na mesma instância do EC2 em um grupo de Amazon EC2 Auto Scaling ao mesmo tempo poderá falhar se uma das implantações tiver scripts que forem executados por mais de alguns minutos. Não implante várias revisões de aplicativos nas mesmas instâncias do EC2 em um grupo de Amazon EC2 Auto Scaling.
- Uma implantação falhará para novas instâncias do EC2 que forem iniciadas como parte de um grupo de Amazon EC2 Auto Scaling. Nesse cenário, a execução de scripts em uma implantação pode impedir a execução de instâncias do EC2 no grupo Amazon EC2 Auto Scaling. (Outras instâncias do EC2 no grupo do Amazon EC2 Auto Scaling podem parecer estar em execução normalmente.) Para resolver esse problema, verifique se todos os outros scripts são concluídos primeiro:
  - CodeDeploy o agente não está incluído na sua AMI: se você usar o cfn-init comando para instalar o CodeDeploy agente ao iniciar uma nova instância, coloque o script de instalação do agente no final da `cfn-init` seção do seu CloudFormation modelo.
  - CodeDeploy o agente está incluído na sua AMI: configure a AMI para que o agente esteja em um `Stopped` estado quando a instância for criada e, em seguida, inclua um script para iniciar o agente como etapa final na sua biblioteca de `cfn-init` scripts.

“CodeDeployRole não lhe dá permissão para realizar operações nas seguintes AWS serviço: AmazonAutoScaling "erro

As implantações que usam um grupo do Auto Scaling criado com um modelo de execução exige as permissões a seguir. Elas são um acréscimo às permissões concedidas pela política `AWSCodeDeployRole` AWS gerenciada.

- `EC2:RunInstances`

- `EC2:CreateTags`
- `iam:PassRole`

Você pode receber esse erro se não tiver essas permissões. Para obter mais informações, consulte Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling, Criação de um modelo de execução para um grupo do Auto Scaling e Permissões no Manual do usuário do Amazon EC2 Auto Scaling.

As instâncias em um grupo de Amazon EC2 Auto Scaling são continuamente provisionadas e encerradas antes que uma revisão possa ser implantada

Em alguns casos, um erro pode impedir uma implantação bem-sucedida em instâncias recém-provisionadas em um grupo Amazon EC2 Auto Scaling. Os resultados são instâncias não íntegras e implantações sem sucesso. Como a implantação não pode ser executada ou concluída com êxito, as instâncias são encerradas logo depois de serem criadas. A configuração do grupo Amazon EC2 Auto Scaling fará com que outro lote de instâncias seja provisionado para tentar atender ao requisito mínimo de hosts íntegros. Este lote também é encerrado, e o ciclo continua.

As possíveis causas incluem:

- Falha nas verificações de integridade do Amazon EC2 Auto Scaling.
- Um erro na revisão de aplicativo.

Para contornar esse problema, siga estas etapas:

1. Crie manualmente uma instância do EC2 que não faça parte do grupo Amazon EC2 Auto Scaling. Marque a instância com uma tag de instância do EC2 exclusiva.
2. Adicione a nova instância ao grupo de implementação afetado.
3. Implante uma nova revisão de aplicativo sem erros no grupo de implantação.

Isso faz com que o grupo do Amazon EC2 Auto Scaling implante a revisão do aplicativo em futuras instâncias do grupo do Amazon EC2 Auto Scaling.

> **Note**
>
> Depois de confirmar que as implantações foram bem-sucedidas, exclua a instância que você criou para evitar cobranças contínuas em sua AWS conta.

O encerramento ou a reinicialização de uma instância do Amazon EC2 Auto Scaling pode causar falhas nas implantações

Se uma instância do EC2 for iniciada através do Amazon EC2 Auto Scaling e, depois, for encerrada ou reinicializada, as implantações nessa instância poderão falhar pelos seguintes motivos:

- Durante uma implantação em andamento, um evento de dimensionamento ou qualquer outro evento de encerramento faz com que a instância se separe do grupo do Amazon EC2 Auto Scaling e seja encerrada. Como a implantação não pode ser concluída, ela falha.
- A instância é reinicializada, mas leva mais de cinco minutos para ser iniciada. CodeDeploy trata isso como um tempo limite. O serviço reprova todas as implantações atuais e futuras na instância.

Para resolver esse problema:

- Em geral, verifique se todas as implantações foram concluídas antes que a instância seja encerrada ou reinicializada. Verifique se todas as implantações foram iniciadas após a reinicialização da instância.
- As implantações poderão falhar se você especificar uma imagem de máquina da Amazon (AMI) de base do Windows Server para uma configuração do Amazon EC2 Auto Scaling e usar o serviço EC2Config para definir o nome do computador da instância. Para corrigir esse problema, na AMI base do Windows Server, na guia Geral de Propriedades do serviço EC2, desmarque Definir o nome do computador. Depois que você desmarcar a caixa de seleção, esse comportamento será desabilitado para todas as novas instâncias do Amazon EC2 Auto Scaling no Windows Server executadas com essa AMI base do Windows Server. Para as instâncias do Amazon EC2 Auto Scaling no Windows Server em que esse comportamento esteja habilitado, você não precisa desmarcar essa caixa de seleção. Basta reimplantar implementações com falha nessas instâncias depois que elas forem reinicializadas.

Evite associar vários grupos de implantação a um único grupo de Amazon EC2 Auto Scaling

Como prática recomendada, você deve associar apenas um grupo de implantação a cada grupo de Amazon EC2 Auto Scaling.

Isso porque, se o Amazon EC2 Auto Scaling aumentar verticalmente a escala de uma instância que tem hooks associados a vários grupos de implantação, ela enviará notificações para todos os hooks de uma só vez. Isso faz com que várias implantações em cada instância sejam iniciadas ao mesmo tempo. Quando várias implantações enviam comandos ao CodeDeploy agente ao mesmo tempo, o tempo limite de cinco minutos entre um evento do ciclo de vida e o início da implantação ou o final do evento do ciclo de vida anterior pode ser atingido. Se isso acontecer, a implantação falhará, mesmo que um processo de implantação esteja em execução conforme o esperado.

> **Note**
>
> O tempo limite padrão para um script em um evento de ciclo de vida é de 30 minutos. Você pode alterar o tempo limite para um valor diferente no AppSpec arquivo. Para obter mais informações, consulte Adicionar um AppSpec arquivo para uma EC2/On-Premises implantação.

Não é possível controlar a ordem em que as implantações ocorrem quando mais de uma implantação tenta ser executada ao mesmo tempo.

Por fim, se a implantação em qualquer instância falhar, o Amazon EC2 Auto Scaling encerrará essa instância imediatamente. Quando essa primeira instância for encerrada, as outras implantações em execução começarão a falhar. Como CodeDeploy o tempo limite é de uma hora para o CodeDeploy agente responder às implantações pendentes, pode levar até 60 minutos para que cada instância atinja o tempo limite.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte Under the CodeDeploy hood: and Auto Scaling integration.

As instâncias do EC2 em um grupo Amazon EC2 Auto Scaling falham ao ser iniciadas e recebem o erro "Heartbeat Timeout"

Um grupo Amazon EC2 Auto Scaling pode falhar ao inicializar novas instâncias do EC2, gerando uma mensagem semelhante à seguinte:

`Launching a new EC2 instance <instance-Id>. Status Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with token<token-Id> was abandoned: Heartbeat Timeout.`

Em geral, essa mensagem indica uma das seguintes situações:

- O número máximo de implantações simultâneas associadas a uma AWS conta foi atingido. Para obter mais informações sobre limites de implantação, consulte CodeDeploy cotas.
- O grupo do Auto Scaling tentou iniciar muitas instâncias do EC2 muito rapidamente. As chamadas de API para RecordLifecycleActionHeartbeatou CompleteLifecycleActionpara cada nova instância foram limitadas.
- Um aplicativo em CodeDeploy foi excluído antes que seus grupos de implantação associados fossem atualizados ou excluídos.

  Quando você exclui um aplicativo ou grupo de implantação, CodeDeploy tenta limpar qualquer gancho do Amazon EC2 Auto Scaling associado a ele, mas alguns ganchos podem permanecer. Se você executar um comando para excluir um grupo de implantação, os ganchos restantes serão retornados na saída. No entanto, se você executar um comando para excluir um aplicativo, os ganchos restantes não aparecerão na saída.

  Portanto, como prática recomendada, você deve excluir todos os grupos de implantação associados a um aplicativo antes de excluir esse aplicativo. É possível usar a saída do comando para identificar os ganchos de ciclo de vida que devem ser excluídos manualmente.

Se você receber uma mensagem de erro “Heartbeat Timeout”, poderá determinar se os ganchos restantes do ciclo de vida são a causa e resolver o problema fazendo o seguinte:

1. Execute um destes procedimentos:

   - Chame o comando delete-deployment-group para excluir o grupo de implantação associado ao grupo do Auto Scaling que está causando o tempo limite do heartbeat.
   - Chame o comando update-deployment-group com uma lista vazia não nula de nomes de grupos do Auto Scaling para separar todos os ganchos do ciclo de vida do Auto Scaling. CodeDeploy-managed

     Por exemplo, digite o seguinte AWS CLI comando:

```
aws deploy update-deployment-group --application-name my-example-app
--current-deployment-group-name my-deployment-group --auto-scaling-
groups
```

Como outro exemplo, se você estiver usando a CodeDeploy API com Java, chame `UpdateDeploymentGroup` e `autoScalingGroups` defina como`new ArrayList<String>()`. Isso configura o `autoScalingGroups` para uma lista vazia e remove a lista existente. Não use `null`, que é o padrão, pois isso deixa o `autoScalingGroups` como está, e isso não é o que você deseja.

Examine a saída da chamada. Se a saída contiver uma estrutura `hooksNotCleanedUp` com uma lista de ganchos do ciclo de vida de Amazon EC2 Auto Scaling, os ganchos do ciclo de vida restantes são provavelmente a causa do erro.

2. Chame o comando describe-lifecycle-hooks, especificando o nome do grupo Amazon EC2 Auto Scaling associado às instâncias do EC2 que não puderam ser executadas. Na saída, procure qualquer um dos seguintes:

   - Nomes de ganchos do ciclo de vida do Amazon EC2 Auto Scaling que correspondem à estrutura `hooksNotCleanedUp` que você identificou na etapa 1.
   - Nomes de ganchos do ciclo de vida do Amazon EC2 Auto Scaling que contêm o nome do grupo de implantação associado ao grupo do Auto Scaling que está falhando.
   - Nomes de ganchos do ciclo de vida do Amazon EC2 Auto Scaling que podem ter causado o tempo limite de pulsação da implantação. CodeDeploy

3. Se um hook se enquadrar em uma das categorias listadas na etapa 2, chame o comando delete-lifecycle-hook para excluí-lo. Especifique o grupo do Amazon EC2 Auto Scaling e o gancho do ciclo de vida na chamada.

> **Important**
>
> Exclua somente os hooks que estão causando problemas, conforme descrito na etapa 2. Se você excluir ganchos viáveis, suas implantações poderão falhar ou CodeDeploy talvez não consigam implantar suas revisões de aplicativos em instâncias escaláveis do EC2.

4. Chame o comando update-deployment-group ou create-deployment-group com os nomes de grupo do Auto Scaling desejados. CodeDeployreinstala os ganchos do Auto Scaling com novos UUIDs.

> **Note**
>
> Se você separar um grupo do Auto Scaling de CodeDeploy um grupo de implantação, qualquer implantação em andamento no grupo do Auto Scaling poderá falhar, e as novas instâncias do EC2 que forem escaladas pelo grupo do Auto Scaling não receberão as revisões do seu aplicativo. CodeDeploy Para que o Auto Scaling funcione novamente CodeDeploy, você precisará reconectar o grupo do Auto Scaling ao grupo de implantação e chamar um novo para iniciar uma implantação em toda `CreateDeployment` a frota.

Ganchos do ciclo de vida do Amazon EC2 Auto Scaling podem fazer com que as implantações automáticas em grupos do Amazon EC2 Auto Scaling parem ou falhem.

O Amazon EC2 Auto CodeDeploy Scaling usa ganchos de ciclo de vida para determinar quais revisões de aplicativos devem ser implantadas em quais instâncias do EC2 após serem lançadas em grupos do Amazon EC2 Auto Scaling. As implantações automáticas podem parar ou falhar se os ganchos do ciclo de vida e as informações sobre esses ganchos não corresponderem exatamente ao Amazon EC2 Auto Scaling e. CodeDeploy

Se as implantações em um grupo do Amazon EC2 Auto Scaling estiverem falhando, veja se os nomes dos ganchos do ciclo de vida no Amazon EC2 Auto Scaling correspondem. CodeDeploy Caso contrário, use essas chamadas de AWS CLI comando.

Primeiro, obtenha a lista dos nomes de ganchos do ciclo de vida para o grupo de Amazon EC2 Auto Scaling e o grupo de implantação:

1. Chame o comando describe-lifecycle-hooks, especificando o nome do grupo Amazon EC2 Auto Scaling associado ao grupo de implantação em. CodeDeploy Na saída, na lista `LifecycleHooks`, anote cada valor de `LifecycleHookName`.
2. Chame o comando get-deployment-group, especificando o nome do grupo de implantação associado ao grupo do Amazon EC2 Auto Scaling. Na saída, na lista `autoScalingGroups`,

localize cada item cujo valor de nome corresponda ao nome do grupo de Amazon EC2 Auto Scaling e depois anote o valor de hook correspondente.

Agora, compare os dois conjuntos de nomes de ganchos de ciclo de vida. Se eles corresponderem exatamente, caractere por caractere, então este não é o problema. Você pode tentar outras etapas de solução de problemas do Amazon EC2 Auto Scaling descritas em outras partes desta seção.

No entanto, se os dois conjuntos de nomes de ganchos de ciclo de vida não corresponderem exatamente, caractere por caractere, faça o seguinte:

1. Se houver nomes de ganchos de ciclo de vida na saída do comando describe-lifecycle-hooks que também não estão na saída do comando get-deployment-group, faça o seguinte:

   a. Para cada nome de gancho de ciclo de vida na saída do comando describe-lifecycle-hooks, chame o comando delete-lifecycle-hook.

   b. Chame o comando update-deployment-group, especificando o nome do grupo original do Amazon EC2 Auto Scaling. CodeDeploy cria novos ganchos de ciclo de vida de substituição no grupo Amazon EC2 Auto Scaling e associa os ganchos de ciclo de vida ao grupo de implantação. As implantações automáticas serão agora retomadas à medida que novas instâncias forem adicionadas ao grupo de Amazon EC2 Auto Scaling.

2. Se houver nomes de ganchos de ciclo de vida na saída do comando get-deployment-group que não estejam também na saída do comando describe-lifecycle-hooks, faça o seguinte:

   a. Chame o comando update-deployment-group, mas não especifique o nome do grupo de Amazon EC2 Auto Scaling original.

   b. Chame o update-deployment-group comando novamente, mas desta vez especifique o nome do grupo original do Amazon EC2 Auto Scaling. CodeDeploy recria os ganchos do ciclo de vida que faltam no grupo Amazon EC2 Auto Scaling. As implantações automáticas serão agora retomadas à medida que novas instâncias forem adicionadas ao grupo de Amazon EC2 Auto Scaling.

Depois de fazer com que os dois conjuntos de nomes de ganchos do ciclo de vida correspondam exatamente, caractere por caractere, as revisões de aplicativos deverão ser reimplantadas, mas apenas nas novas instâncias à medida que elas forem adicionadas ao grupo de Amazon EC2 Auto Scaling. As implantações não ocorrem automaticamente para as instâncias que já estão no grupo do Amazon EC2 Auto Scaling.

Erro “A implantação falhou porque nenhuma instância foi encontrada para seu grupo de implantação”

Leia esta seção se você ver o seguinte CodeDeploy erro:

```
The deployment failed because no instances were found for your deployment group. Check your deployment group settings to make sure the tags for your EC2 instances or Auto Scaling groups correctly identify the instances you want to deploy to, and then try again.
```

As causas possíveis esse erro são:

1. Suas configurações de grupo de implantação incluem tags para instâncias do EC2, instâncias on-premises ou grupos do Auto Scaling que não estão corretas. Para corrigir esse problema, verifique se suas tags estão corretas e reimplante seu aplicativo.
2. Sua frota aumentou a escala horizontalmente após o início da implantação. Nesse cenário, você vê instâncias íntegras no estado `InService` da sua frota, mas também vê o erro acima. Para corrigir o problema, reimplante o aplicativo.
3. Seu grupo do Auto Scaling não inclui nenhuma instância que esteja no estado `InService`. Nesse cenário, quando você tenta fazer uma implantação em toda a frota, a implantação falha com a mensagem de erro acima porque CodeDeploy precisa de pelo menos uma instância para estar no `InService` estado. Há muitos motivos pelos quais você pode não ter instâncias no estado `InService`. Alguns deles incluem:
   - Você programou (ou configurou manualmente) o tamanho do grupo do Auto Scaling para ser `0`.
   - O ajusta de escala automático detectou instâncias defeituosas do EC2 (por exemplo, as instâncias do EC2 tiveram falhas de hardware), então cancelou todas elas, sem deixar nenhuma no estado `InService`.
   - Durante um evento de expansão horizontal de `0` até`1`, CodeDeploy implantou uma revisão anteriormente bem-sucedida (chamada de última revisão bem-sucedida) que não estava íntegra desde a última implantação. Isso fez com que a implantação na instância escalonada falhasse, o que, por sua vez, fez com que o ajuste de escala automático cancelasse a instância, sem deixar nenhuma instância no estado `InService`.

     Se você achar que não tem instâncias no estado `InService`, solucione o problema conforme descrito no procedimento a seguir, To troubleshoot the error if there are no instances in the InService state.

Para solucionar o erro se não houver instâncias no estado InService

1. No console do Amazon EC2, verifique a configuração de Capacidade desejada. Se for zero, defina-a como um número positivo. Aguarde até que a instância apareça como `InService`, o que significa que a implantação foi bem-sucedida. Você corrigiu o problema e pode pular as etapas restantes desse procedimento de solução de problemas. Para obter informações sobre como definir a configuração de Capacidade desejada, consulte Definindo limites de capacidade em seu grupo do Auto Scaling no Guia do usuário do Amazon EC2 Auto Scaling.
2. Se o ajuste de escala automático continuar tentando lançar novas instâncias do EC2 para atingir a capacidade desejada, mas nunca conseguir atingir a escala horizontal, geralmente é devido a uma falha no gancho do ciclo de vida do ajuste de escala automático. Solucione esse problema da seguinte maneira:

   a. Para verificar qual evento de gancho do ciclo de vida do ajuste de escala automático está falhando, consulte Como verificar uma ação de escala em um grupo do Auto Scaling no Guia do usuário do Amazon EC2 Auto Scaling.

   b. Se o nome do gancho com falha for`CodeDeploy-managed-automatic-launch-deployment-hook-`*`DEPLOYMENT_GROUP_NAME`*, acesse CodeDeploy, encontre o grupo de implantação e encontre a implantação com falha que foi iniciada pelo Auto Scaling. Em seguida, investigue por que a implantação falhou.

   c. Se você entender por que a implantação falhou (por exemplo, CloudWatch alarmes estavam ocorrendo) e puder corrigir o problema sem alterar a revisão, faça isso agora.

   d. Se, após a investigação, você determinar que CodeDeploy a última revisão bem-sucedida não está mais íntegra e que não há nenhuma instância íntegra em seu grupo de Auto Scaling, você está em um cenário de impasse na implantação. Para resolver esse problema, você deve corrigir a CodeDeploy revisão incorreta removendo temporariamente o gancho de ciclo CodeDeploy de vida do grupo Auto Scaling e, em seguida, reinstalando o gancho e reimplantando uma nova (boa) revisão. Para obter instruções, consulte:

      - To fix the deployment deadlock issue (CLI)
      - To fix the deployment deadlock issue (console)

Para corrigir o problema de impasse de implantação (CLI)

1. (Opcional) Bloqueie os CI/CD pipelines que estão causando o CodeDeploy erro para que implantações inesperadas não ocorram enquanto você corrige esse problema.

2. Anote sua configuração atual do Auto Scaling DesiredCapacity:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name ASG_NAME
   ```

   Talvez seja necessário reduzir para esse número no final deste procedimento.

3. Defina sua configuração de Auto Scaling DesiredCapacitycomo. 1 Isso é opcional se a capacidade desejada for maior do que 1 no início. Ao reduzi-la para 1, a instância levará menos tempo para provisionar e implantar posteriormente, o que acelera a solução de problemas. Se a capacidade desejada do ajuste de escala automático foi originalmente definida como `0`, você deve aumentá-la para 1. Isso é obrigatório.

   conjunto de escalabilidade automática aws --auto-scaling-group-name --desired-capacity 1 *ASG_NAME*

   > **Note**
   >
   > As etapas restantes desse procedimento pressupõem que você tenha definido seu DesiredCapacity1.

   Nesse momento, o ajuste de escala automático tenta escalar para uma instância. Então, como o gancho CodeDeploy adicionado ainda está presente, CodeDeploy tenta implantar; a implantação falha; o Auto Scaling cancela a instância; e o Auto Scaling tenta relançar uma instância para atingir a capacidade desejada de uma, o que novamente falha. Você está em um ciclo de cancelamento e reinicialização.

4. De-register o grupo Auto Scaling do grupo de implantação:

   > **Warning**
   >
   > O comando a seguir iniciará uma nova instância do EC2 sem nenhum software nela. Antes de executar o comando, verifique se uma instância `InService` do ajuste de escala automático que não executa nenhum software é aceitável. Por exemplo, certifique-se de que o balanceador de carga associado à instância não esteja enviando tráfego para esse host sem software.

> **Important**
>
> Use o CodeDeploy comando mostrado abaixo para remover o gancho. Não remova o gancho por meio do serviço Auto Scaling, pois a remoção não será reconhecida pelo. CodeDeploy

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups
```

Depois que o comando é executado, ocorre o seguinte:

a. CodeDeploy cancela o registro do grupo Auto Scaling do grupo de implantação.

b. CodeDeploy remove o gancho do ciclo de vida do Auto Scaling do grupo Auto Scaling.

c. Como o hook que estava causando uma falha na implantação não está mais presente, o ajuste de escala automático cancela a instância EC2 existente e inicia imediatamente uma nova para escalar até a capacidade desejada. A nova instância deve entrar no estado `InService` em breve. A nova instância não inclui software.

5. Espere que a instância EC2 entre em um estado `InService`. Para verificar seu estado, use o seguinte comando:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
```

6. Adicione o hook de volta à instância do EC2:

> **Important**
>
> Use o CodeDeploy comando mostrado abaixo para adicionar o gancho. Não use o serviço Auto Scaling para adicionar o gancho, pois a adição não será reconhecida pelo. CodeDeploy

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups ASG_NAME
```

Depois que o comando é executado, ocorre o seguinte:

a. CodeDeploy reinstala o gancho do ciclo de vida do Auto Scaling na instância do EC2

b. CodeDeploy registra novamente o grupo Auto Scaling com o grupo de implantação.

7. Crie uma implantação em toda a frota com o Amazon S3 GitHub ou a revisão que você sabe que está íntegra e deseja usar.

   Por exemplo, se a revisão for um arquivo.zip em um bucket do Amazon S3 chamado `my-revision-bucket` com uma chave de objeto `httpd_app.zip`, digite o seguinte comando:

   ```
   aws deploy create-deployment --application-name APPLICATION_NAME
   --deployment-group-name DEPLOYMENT_GROUP_NAME --
   revision "revisionType=S3,s3Location={bucket=my-revision-
   bucket,bundleType=zip,key=httpd_app.zip}"
   ```

   Como agora há uma instância `InService` no grupo do Auto Scaling, essa implantação deve funcionar e você não deve mais ver o erro A implantação falhou porque nenhuma instância foi encontrada para seu grupo de implantação.

8. Depois que a implantação for bem-sucedida, escale seu grupo do Auto Scaling de volta para a capacidade original, caso você tenha escalado anteriormente:

   ```
   aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME
   --desired-capacity ORIGINAL_CAPACITY
   ```

Para corrigir o problema de impasse de implantação (console)

1. (Opcional) Bloqueie os CI/CD pipelines que estão causando o CodeDeploy erro para que implantações inesperadas não ocorram enquanto você corrige esse problema.
2. Acesse o console do Amazon EC2 e anote a configuração de Capacidade desejada do ajuste de escala automático. Talvez seja necessário reduzir para esse número no final deste procedimento. Para obter informações sobre como encontrar essa configuração, consulte Como definir limites de capacidade em seu grupo do Auto Scaling.
3. Defina o número desejado de instâncias do EC2 como 1:

   Isso é opcional se a capacidade desejada for maior do que 1 no início. Ao reduzi-la para 1, a instância levará menos tempo para provisionar e implantar posteriormente, o que acelera

a solução de problemas. Se a Capacidade desejada do ajuste de escala automático foi originalmente definida como `0`, você deve aumentá-la para 1. Isso é obrigatório.

> **Note**
>
> As etapas restantes desse procedimento pressupõem que você tenha definido a capacidade desejada como 1.

a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/e escolha Auto Scaling Groups no painel de navegação.

b. Escolha a região apropriada.

c. Acesse o grupo do Auto Scaling problemático.

d. Em Detalhes do grupo, escolha Editar.

e. Defina a Capacidade desejada como **1**.

f. Selecione Atualizar.

4. De-register o grupo Auto Scaling do grupo de implantação:

> **Warning**
>
> As subetapas a seguir iniciarão uma nova instância do EC2 sem nenhum software nela. Antes de executar o comando, verifique se uma instância `InService` do ajuste de escala automático que não executa nenhum software é aceitável. Por exemplo, certifique-se de que o balanceador de carga associado à instância não esteja enviando tráfego para esse host sem software.

a. Abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.

b. Escolha a região apropriada.

c. No painel de navegação, escolha Aplicativos.

d. Escolha o nome do seu CodeDeploy aplicativo.

e. Escolha o nome do seu grupo CodeDeploy de implantação.

f. Escolha Editar.

g. Em Configuração do ambiente, desmarque grupos do Amazon EC2 Auto Scaling.

> **Note**
>
> O console não permite que você salve a configuração se não houver uma configuração de ambiente definida. Para ignorar a verificação, adicione temporariamente uma tag de `EC2` ou `On-premises` que você sabe que não será resolvida em nenhum host. Para adicionar uma tag, selecione instâncias ou On-premises instâncias do Amazon EC2 e adicione uma tag Key de **`EC2`** ou. **`On-premises`** Você pode deixar a tag Value vazia.

h. Escolha Salvar alterações.

Depois de concluir essas subetapas, o seguinte vai acontecer:

i. CodeDeploy cancela o registro do grupo Auto Scaling do grupo de implantação.

ii. CodeDeploy remove o gancho do ciclo de vida do Auto Scaling do grupo Auto Scaling.

iii. Como o hook que estava causando uma falha na implantação não está mais presente, o ajuste de escala automático cancela a instância EC2 existente e inicia imediatamente uma nova para escalar até a capacidade desejada. A nova instância deve entrar no estado `InService` em breve. A nova instância não inclui software.

5. Espere que a instância EC2 entre em um estado `InService`. Para verificar seu estado:

a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.

b. No painel de navegação, escolha Groups Auto Scaling.

c. Escolha seu grupo do Auto Scaling.

d. No painel de conteúdo, escolha a guia Gerenciamento de instância.

e. Em Instâncias, verifique se a coluna Ciclo de vida indica InServiceao lado da instância.

6. Re-register o grupo Auto Scaling com o grupo de CodeDeploy implantação usando o mesmo método usado para removê-lo:

a. Abra o CodeDeploy console em https://console.aws.amazon.com/codedeploy/.

b. Escolha a região apropriada.

c. No painel de navegação, escolha Aplicativos.

d. Escolha o nome do seu CodeDeploy aplicativo.

e. Escolha o nome do seu grupo CodeDeploy de implantação.

f. Escolha Editar.

g. Em Configuração do ambiente, selecione grupos do Amazon EC2 Auto Scaling e selecione seu grupo do Auto Scaling na lista.

h. Em Instâncias ou On-premises instâncias do Amazon EC2, encontre a tag que você adicionou e remova-a.

i. Desmarque a caixa de seleção ao lado das instâncias ou instâncias On-premises do Amazon EC2.

j. Escolha Salvar alterações.

Essa configuração reinstala o gancho do ciclo de vida no grupo do Auto Scaling.

7. Crie uma implantação em toda a frota com o Amazon S3 GitHub ou a revisão que você sabe que está íntegra e deseja usar.

   Por exemplo, se a revisão for um arquivo.zip em um bucket do Amazon S3 chamado `my-revision-bucket` com uma chave de objeto de `httpd_app.zip`, faça o seguinte:

   a. No CodeDeploy console, na página Grupo de Implantação, escolha Criar implantação.

   b. Em Tipo de revisão, escolha Meu aplicativo está armazenado no Amazon S3.

   c. Para Local de revisão, escolha `s3://my-revision-bucket/httpd_app.zip`.

   d. Em Tipo de arquivo de revisão, selecione `.zip`.

   e. Escolha Criar implantação.

   Como agora há uma instância `InService` no grupo do Auto Scaling, essa implantação deve funcionar e você não deve mais ver o erro A implantação falhou porque nenhuma instância foi encontrada para seu grupo de implantação.

8. Depois que a implantação for bem-sucedida, escale seu grupo do Auto Scaling de volta para a capacidade original, caso você tenha escalado anteriormente:

   a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/e escolha Auto Scaling Groups no painel de navegação.

   b. Escolha a região apropriada.

   c. Vá para seu grupo do Auto Scaling.

   d. Em Detalhes do grupo, escolha Editar.

   e. Configure a Capacidade desejada de volta ao valor original.

   f. Selecione Atualizar.

Códigos de erro para AWS CodeDeploy

Este tópico fornece informações de referência sobre CodeDeploy erros.

Código de erro	Description
`AGENT_ISSUE`	A implantação falhou devido a um problema com o CodeDeploy agente. Certifique-se de que o agente esteja instalado e em execução em todas as instâncias neste grupo de implantação. Saiba mais: • Verifique se o CodeDeploy agente está em execução • Instale o CodeDeploy agente • Trabalhando com o CodeDeploy agente
`AUTO_SCALING_IAM_ROLE_PERMISSIONS`	O perfil de serviço associada ao seu grupo de implantação não tem a permissão necessária para realizar operações no seguinte serviço da AWS . Saiba mais: • Etapa 2: criar uma função de serviço para CodeDeploy • Criar um perfil para delegar permissões a um serviço da AWS
`HEALTH_CONSTRAINTS`	A implantação geral falhou porque muitas instâncias individuais falharam na implantação, poucas instâncias íntegras estão disponíveis para implantação ou algumas instâncias no seu grupo de implantação estão apresentando problemas.

Código de erro	Description
	Saiba mais: • Instance Health • Solução de problemas de instância • Solucionar problemas de EC2/On-Premises implantação
HEALTH_CONSTRAINTS_INVALID	A implantação não pode ser iniciada porque o número mínimo de instâncias íntegras, conforme definido pela sua configuração de implantação, não está disponível. Você pode reduzir o número necessário de instâncias íntegras, atualizando sua configuração de implantação ou aumentando o número de instâncias nesse grupo de implantação. Saiba mais: • Instance Health • Trabalhando com instâncias para CodeDeploy
IAM_ROLE_MISSING	A implantação falhou porque nenhuma função de serviço existe com o nome de função de serviço especificado para o grupo de implantação. Verifique se você está usando o nome da função de serviço correto. Saiba mais: • Etapa 2: criar uma função de serviço para CodeDeploy • Altere as configurações do grupo de implantação com CodeDeploy

Código de erro	Description
`IAM_ROLE_PERMISSIONS`	CodeDeploy não tem as permissões necessárias para assumir uma função, ou a função do IAM que você está usando não lhe dá permissão para realizar operações em um AWS serviço. Saiba mais: • Etapa 1: configuração • Etapa 2: criar uma função de serviço para CodeDeploy • Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2

Código de erro	Description
NO_INSTANCES	Isso pode ser causado por uma das situações a seguir. • Para uma EC2/On-Premises blue/green implantação, se você usar tags do Amazon EC2, elas podem não estar configuradas corretamente. Em seu grupo de CodeDeplo y implantação, certifique-se de que eles estejam incluídos em suas instâncias azuis e verdes. Você pode confirmar se as suas instâncias estão marcadas corretamente usando o console do Amazon EC2. • Se você usar um grupo de Amazon EC2 Auto Scaling, seu grupo do Auto Scaling talvez não tenha capacidade suficiente. Verifique se o seu grupo do Auto Scaling tem capacidade suficiente para a implantaç ão. Você pode visualizar a capacidade do seu grupo de grupo do Amazon EC2 Auto Scaling observando o número de instâncias íntegras usando o console do Amazon EC2. • Para uma EC2/On-Premises blue/green implantação, as frotas azul e verde podem não ter o mesmo tamanho. Verifique se ambas as frotas têm o mesmo tamanho. Saiba mais: • Tagging Instances for Deployments • Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling • Criar um aplicativo para uma blue/green implantação (console)

Código de erro	Description
OVER_MAX_INSTANCES	A implantação falhou porque mais instâncias estão direcionadas para implantação do que o permitido para a sua conta. Para reduzir o número de instâncias direcionadas para essa implantação, atualize as configurações de tag desse grupo de implantação ou exclua algumas das instâncias direcionadas. Como alternati va, você pode entrar em contato AWS Support para solicitar um aumento de limite. Saiba mais: • Altere as configurações do grupo de implantação com CodeDeploy • CodeDeploy cotas • Solicitar um aumento de limite
THROTTLED	A implantação falhou porque foram feitas mais solicitações do que as permitidas AWS CodeDeploy por uma função do IAM. Tente reduzir o número de solicitações. Saiba mais: • Taxa de solicitações de APIs de consulta

Código de erro	Description
UNABLE_TO_SEND_ASG	A implantação falhou porque o grupo de implantação não está configurado corretamente com seu grupo Amazon EC2 Auto Scaling. No CodeDeploy console, exclua o grupo Amazon EC2 Auto Scaling do grupo de implantação e adicione-o novamente. Saiba mais: • Sob o capô: CodeDeploy e integração com o Auto Scaling

Tópicos relacionados

Solução de problemas CodeDeploy

CodeDeploy recursos

Os seguintes recursos relacionados podem ajudá-lo a trabalhar com CodeDeploy.

Guias de referência e recursos de suporte

- AWS CodeDeploy Referência da API — descrições, sintaxe e exemplos de uso sobre CodeDeploy ações e tipos de dados, incluindo parâmetros comuns e códigos de erro.
- CodeDeploy Perguntas frequentes técnicas — Principais perguntas dos clientes sobre CodeDeploy.
- AWS centro de suporte — O hub para criar e gerenciar seus AWS Support casos. Também inclui links para outros recursos, como fóruns, perguntas frequentes técnicas, status de integridade do serviço e AWS Trusted Advisor.
- AWS planos de suporte — A página principal da web para obter informações sobre AWS Support planos.
- Fale conosco — Um ponto de contato central para consultas sobre AWS faturamento, conta, eventos, abuso e outros problemas.
- AWS termos do site — Informações detalhadas sobre nossos direitos autorais e nossa marca registrada; sua conta, licença e acesso ao site; e outros tópicos.

Amostras

- CodeDeploy amostras em GitHub — Amostras e modelos de cenários para CodeDeploy.
- CodeDeploy Plugin Jenkins — Plugin Jenkins para. CodeDeploy
- CodeDeploy agente — Open-source versão do CodeDeploy agente.

Blogs

- AWS DevOps blog — Insights para desenvolvedores, administradores de sistemas e arquitetos.

AWS kits e ferramentas de desenvolvimento de software

Os seguintes AWS SDKs e ferramentas oferecem suporte ao desenvolvimento de soluções com CodeDeploy:

- AWS SDK para .NET
- AWS SDK para Java
- AWS SDK para JavaScript
- AWS SDK para PHP
- AWS SDK para Python (Boto)
- AWS SDK para Ruby
- AWS Toolkit for Eclipse — Partes 1, 2 e 3.
- AWS Tools for Windows PowerShell— Um conjunto de PowerShell cmdlets do Windows que expõe a funcionalidade do AWS SDK para .NET no ambiente. PowerShell
- CodeDeploy cmdlets no Ferramentas da AWS para PowerShell — Um conjunto de PowerShell cmdlets do Windows que expõe a funcionalidade do ambiente. CodeDeploy PowerShell
- AWS Command Line Interface— Uma sintaxe de linha de comando uniforme para acessar AWS serviços. O AWS CLI usa um único processo de configuração para permitir o acesso a todos os serviços compatíveis.
- AWS ferramentas para desenvolvedores — Links para ferramentas e recursos para desenvolvedores que fornecem documentação, exemplos de código, notas de versão e outras informações para ajudá-lo a criar aplicativos inovadores com CodeDeploy AWS e.

Histórico do documento

A tabela a seguir descreve as principais mudanças feitas neste guia do usuário para dar suporte as novas e aprimoradas funcionalidades desde a última versão do CodeDeploy User guide (Guia do usuário).

- Versão da API: 06-10-2014

Alteração	Descrição	Data
CodeDeploy versão v2.1.2 do agente com suporte para macOS	O AWS CodeDeploy agente foi atualizado para a versão 2.1.2, que adiciona suporte para macOS em instância s EC2 Mac. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente e Instalar o CodeDeploy agente para macOS.	13 de outubro de 2026
CodeDeploy versão v2.1.1 do agente	O AWS CodeDeploy agente foi atualizado para a versão 2.1.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	2 de outubro de 2026
CodeDeploy versão v2.1.0 do agente	O AWS CodeDeploy agente foi atualizado para a versão 2.1.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	7 de setembro de 2026
CodeDeploy versão v2.0.1 do agente	O AWS CodeDeploy agente foi atualizado para a versão	24 de agosto de 2026

	2.0.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	
CodeDeploy versão v2.0.0 do agente	O AWS CodeDeploy agente foi atualizado para a versão 2.0.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	15 de julho de 2026
CodeDeploy versão v1.8.1 do agente	O AWS CodeDeploy agente foi atualizado para a versão 1.8.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	3 de fevereiro de 2026
CodeDeploy versão do agente v1.8.0	O AWS CodeDeploy agente foi atualizado para a versão 1.8.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	31 de julho de 2025
CodeDeploy atualizou uma política AWS gerenciada existente	O `AWSCodeDeployDeployerAccess` foi atualizado. Para obter mais informações, consulte Políticas gerenciadas pela AWS : .	16 de dezembro de 2024
CodeDeploy atualizou uma política AWS gerenciada existente	O `AWSCodeDeployReadOnlyAccess` foi atualizado. Para obter mais informações, consulte Políticas gerenciadas pela AWS : .	16 de dezembro de 2024

CodeDeploy atualizou uma política AWS gerenciada existente	O `AWSCodeDeployFullAccess` foi atualizado. Para obter mais informações, consulte Políticas gerenciadas pela AWS : .	16 de dezembro de 2024
CodeDeploy versão do agente v1.7.1	O AWS CodeDeploy agente foi atualizado para a versão 1.7.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	14 de novembro de 2024
Atualização dos nomes de buckets do Amazon S3	O exemplo de buckets do Amazon S3 neste guia foi atualizado para usar nomes reservados por. AWS	17 de junho de 2024
Texto alternativo adicionado (texto alternativo)	Todas as imagens deste guia foram atualizadas para incluir texto alternativo. O texto alternativo é lido por leitores de tela e torna nossa documentação mais acessível para usuários cegos.	22 de maio de 2024
CodeDeploy versão do agente v1.7.0	O AWS CodeDeploy agente foi atualizado para a versão 1.7.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	6 de março de 2024

Comandos alterados	Os `sudo service codedeploy-agent` *`status|start|stop`* comandos não são mais recomendados porque não são usados `systemd` para o gerenciamento de processos do CodeDeploy agente, o que é uma prática recomendada. Para garantir que `systemd` seja utilizado, utilize comandos `systemctl` , conforme mostrado no exemplo a seguir: `sytemctl start codedeploy-agent`. Os tópicos a seguir foram atualizados com `systemctl` comandos: Instalar o CodeDeploy agente para Amazon Linux ou RHEL, Instalar o CodeDeploy agente para Ubuntu Server, Solucionar problemas de erros ignorados em todos os eventos do ciclo de vida e Criar um novo arquivo de CodeDeploy log caso ele tenha sido excluído acidentalmente.	12 de janeiro de 2024
Tópicos adicionados	Foram adicionados os tópicos Gerenciando o processo do CodeDeploy agente e referenciando arquivos em seus scripts de eventos de ciclo de vida.	12 de janeiro de 2024

CodeDeploy agora suporta configurações zonais	Atualizado o CodeDeploy tópico Criar uma configuração de implantação com informaçõ es de configuração zonal.	7 de dezembro de 2023
CodeDeploy agora suporta implantações de terminação	Foi adicionado o tópico Ativar implantações de encerramento durante eventos de redução da escala horizontal do Auto Scaling ao recurso de implantação de encerramento descrito. Também atualizou a seção “ganchosAppSpec ” para uma EC2/On-Premises implantação, os tópicos Criar um grupo de implantação para uma implantação local (console) e Criar um grupo de implantação para uma EC2/On-Premises blue/gree n implantação (console) para explicar o recurso.	7 de dezembro de 2023
Formatação JSON corrigida	Corrigida a formatação das amostras de código JSON no tópico da seção AppSpec “recursos” (somente Amazon ECS e AWS Lambda implantações).	3 de dezembro de 2023
Tópico de solução de problemas adicionado	Foi adicionado um tópico sobre solução de problemas de implantação do Amazon ECS.	24 de outubro de 2023

Atualizou o nome AppSpec do arquivo	A referência do CodeDeploy AppSpec arquivo foi atualizada para indicar que o AppSpec arquivo deve ser nomeado `appspec.yml` para EC2/On-Premises implantações.	5 de outubro de 2023
CodeDeploy agora suporta vários balanceadores de carga	Os tópicos Criar um grupo de implantação para uma implantação local (console), Criar um grupo de implantação para uma EC2/On-Premises blue/green implantação (console) e Configurar um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2 foram atualizados para indicar suporte para vários balanceadores de carga.	26 de setembro de 2023
Atualizou o tópico Regiões no VPC	A tabela no tópico Use CodeDeploy with Amazon Virtual Private Cloud foi atualizada para mostrar suporte adicional à região. Especificamente, as regiões Ásia-Pacífico (Hyderabad), Ásia-Pacífico (Melbourne), Europa (Milão), Europa (Espanha) e Europa (Zurique) foram atualizadas para mostrar suporte ao endpoint do agente.	22 de setembro de 2023

Regiões atualizadas no tópico do kit de recursos	Foram adicionadas as seguintes AWS regiões à seção de nomes de repositórios do kit de recursos por região: Ásia-Pacífico (Osaka), Ásia-Pacífico (Hyderabad), Canadá (Central), Europa (Espanha), Europa (Zurique), Oriente Médio (EAU). Também atualizei as políticas do IAM com essas regiões e quaisquer outras que estavam faltando.	22 de setembro de 2023
Tópicos abreviados de instalação e atualização do agente	Reduziu os tópicos Instalar o CodeDeploy agente para Windows Server e Atualizar o CodeDeploy agente no Windows Server. Foram removidos URLs de bucket redundantes do Amazon S3 e comandos de cópia do Amazon S3.	21 de setembro de 2023
Região Ásia-Pacífico (Jacarta)	Foi adicionada a região Ásia-Pacífico (Jacarta) aos nomes de buckets do kit de recursos por região.	21 de setembro de 2023
CodeDeploy atualizou uma política AWS gerenciada existente	A política AWSCodeDeployRole gerenciada foi atualizada. Para ter mais informações, consulte Atualizações da AWS para políticas gerenciadas pela AWS.	16 de agosto de 2023

Adicionou um limite	Foi adicionado um limite ao tópico de CodeDeploy limites. O limite é Uma lista de alarmes associados ao grupo de implantação.	15 de agosto de 2023
Corrigida uma etapa relaciona da aos balanceadores de carga	Foram corrigidas as instruçõe s em Criar um grupo de implantação para uma EC2/ On-Premises blue/green implantação (console). A etapa do balanceador de carga agora está marcada como opcional.	3 de agosto de 2023
Redação esclarecida no tópico do Amazon ECS	Esclareceu o texto no Tutorial: Implante um aplicativo no Amazon ECS. O texto agora indica que você está implantan do um aplicativo. Anteriorm ente, o texto indicava que você estava implantando um serviço Amazon ECS.	3 de agosto de 2023
CodeDeploy agora está disponível na região de Israel (Tel Aviv)	CodeDeploy agora está disponível na região de Israel (Tel Aviv) (il-central-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	31 de julho de 2023

Atualização de tópico	O tópico Solucionar problemas de EC2/On-Premises implantação foi atualizado com uma dica sobre como usar um runbook para realizar tarefas de solução de problemas automaticamente.	7 de julho de 2023
Atualização de tópico	Atualizou a seção “recursos AppSpec ” para o tópico de implantações do Amazon ECS com mais informações sobre o ARN de definição de tarefa.	7 de julho de 2023
Atualização de tópico	Atualização da Etapa 1: Instalar e configurar o tópico AWS CLI na instância local com informações sobre solução de problemas.	7 de julho de 2023
Atualização de tópico	Atualizou o Cross-ser viceconfuso tópico de prevenção de deputados com informações sobre blue/green implantações do Amazon ECS por meio de. CloudFormation	6 de julho de 2023
Atualização de tópico	O Cross-serviceconfuso tópico de prevenção de deputados foi atualizado com informações sobre as blue/green implantaç ões do Amazon ECS por meio de. CloudFormation	6 de julho de 2023

Atualização de tópico	O tópico Configurações de implantação predefinidas para uma plataforma de EC2/On-Premises computação foi atualizado. Foi adicionada uma observação sobre o comportamento da configuração de implantação `CodeDeployDefault.HalfAtATime` predefinida com grupos de Auto Scaling.	29 de junho de 2023
Atualização de tópico	O AWS CodeDeploy tópico Segurança da infraestrutura foi atualizado para indicar as novas versões mínimas e recomendadas do protocolo Transport Layer Security (TLS).	28 de junho de 2023
Limite atualizado	O seguinte limite foi alterado: “Número máximo de horas que uma implantação EC2/On-Premises local pode ser executada”. Para obter mais informações, consulte Limites	27 de junho de 2023
Atualização de tópico	O tópico Etapa 3: Limitar as permissões do CodeDeploy usuário foi atualizado com instruções detalhadas.	31 de maio de 2023

CodeDeploy atualizou uma política AWS gerenciada existente	A política AWSCodeDeployFullAccess gerenciada foi atualizada. Para ter mais informações, consulte Atualizações da AWS para políticas gerenciadas pela AWS.	16 de maio de 2023
CodeDeploy versão do agente v1.6.0	O AWS CodeDeploy agente foi atualizado para a versão 1.6.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	30 de março de 2023
CodeDeploy versão do agente v1.5.0	O AWS CodeDeploy agente foi atualizado para a versão 1.5.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	3 de março de 2023
Atualização da plataforma computacional Amazon ECS	Implantações em uma plataforma de computação do Amazon ECS são suportadas na região Ásia-Pacífico (Jacarta).	8 de fevereiro de 2023
CodeDeploy atualizou uma política AWS gerenciada existente	A política AWSCodeDeployRole gerenciada foi atualizada. Para ter mais informações, consulte Atualizações da AWS para políticas gerenciadas pela AWS.	3 de fevereiro de 2023

Atualização de tópico	O tópico Uso CodeDeploy com a Amazon Virtual Private Cloud foi atualizado com AWS regiões novas e alteradas.	2 de fevereiro de 2023
Atualizações de tópicos	CodeDeploy agora está disponível na região Ásia-Pacífico (Melbourne) (ap-southeast-4). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessas novas regiões.	26 de janeiro de 2023
Práticas recomendadas de segurança para o	A CodeDeploy seção Introdução e algumas outras seções foram atualizadas para se adequarem às melhores práticas AWS de segurança.	23 de janeiro de 2023
CodeDeploy versão v1.4.1 do agente	O AWS CodeDeploy agente foi atualizado para a versão 1.4.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	6 de dezembro de 2022

Tópico de solução de problemas adicionado	Foi adicionado um tópico sobre como solucionar erros causados por longos caminhos de arquivo usados com o CodeDeploy agente para Windows. Para obter mais informações, consulte Caminhos de arquivo longos causam erros “Esse arquivo ou diretório não existe”.	6 de dezembro de 2022
Alterou um limite	O limite a seguir foi alterado: “Número máximo de configurações de implantação personalizadas associadas a uma AWS conta”. O limite agora é 200. Para saber mais sobre limites, consulte o tópico Limites.	7 de setembro de 2022
CodeDeploy versão do agente v1.4.0	O AWS CodeDeploy agente foi atualizado para a versão 1.4.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	31 de agosto de 2022

Alguns limites foram corrigidos.	Os seguintes limites foram corrigidos: “O número máximo de implantações simultâneas associadas a uma AWS conta” agora é 1000. ‘O número máximo de instâncias em uma única implantação’ agora é 1000. O ‘número máximo de instâncias que pode ser usado por implantações simultâneas que estão em andamento e associadas a uma conta’ agora é 1000. “O número máximo de configurações de implantação personalizadas associadas a uma AWS conta” agora é 100. Para saber mais sobre limites, consulte o tópico Limites.	8 de agosto de 2022
Foi adicionada uma tabela que mostra os CodeDeploy endpoints suportados em cada região.	Para obter mais informações, consulte Usar CodeDeploy com a Amazon Virtual Private Cloud.	20 de abril de 2022
Foi adicionado um novo limite para blue/green implantações do Amazon ECS.	O número máximo de horas entre a implantação de uma revisão e a transferência do tráfego para o ambiente substituto durante uma blue/green implantação do Amazon ECS agora é 120. Para obter mais informações, consulte Implantações no tópico Limites.	12 de abril de 2022

Foi adicionado um tópico sobre como evitar o problema do deputy confuso	Para obter mais informaçõ es, consulte AWS Identity and Access Management para AWS CodeDeploy.	14 de março de 2022
CodeDeploy atualizou uma política AWS gerenciada existente	O perfil `AmazonEC2 RoleforAWSCodeDepl oyLimited` foi atualizado. Para obter mais informações, consulte Políticas gerenciadas pela AWS : .	22 de novembro de 2021
CodeDeploy atualizou uma política AWS gerenciada existente	O `AWS CodeDeployRole` foi atualizado. Para obter mais informações, consulte Políticas gerenciadas pela AWS : .	18 de maio de 2021
CodeDeploy versão v1.3.2 do agente	O AWS CodeDeploy agente foi atualizado para a versão 1.3.2. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	6 de maio de 2021
CodeDeploy suporta a atualização de instâncias desatualizadas do Amazon EC2	CodeDeploy agora suporta a atualização automática de instâncias desatualizadas do Amazon EC2. Para obter mais informações, consulte Configurar opções avançadas para um grupo de implantaç ão.	23 de fevereiro de 2021

CodeDeploy versão do agente v1.3.1	O AWS CodeDeploy agente foi atualizado para a versão 1.3.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	22 de dezembro de 2020
CodeDeploy versão do agente v1.3.0	O AWS CodeDeploy agente foi atualizado para a versão 1.3.0. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	10 de novembro de 2020
CodeDeploy versão v1.2.1 do agente	O AWS CodeDeploy agente foi atualizado para a versão 1.2.1. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	23 de setembro de 2020
CodeDeploy oferece suporte a endpoints Amazon VPC desenvolvidos por AWS PrivateLink	Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e. CodeDeploy Você pode usar essa conexão para habilitar o CodeDeploy a se comunicar com os seus recursos na VPC sem passar pela Internet pública. Para obter mais informações, consulte Usar CodeDeploy com a Amazon Virtual Private Cloud.	11 de agosto de 2020

Limites CodeDeploy de serviço atualizados	Atualizou o limite do número de aplicativos por conta e do número de grupos de implantação por aplicativo para 1000. Para obter mais informações sobre limites CodeDeploy de serviço, consulte CodeDeploy limites.	6 de agosto de 2020
CodeDeploy versão v1.1.2 do agente	O AWS CodeDeploy agente foi atualizado para a versão 1.1.2. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	4 de agosto de 2020
CodeDeploy lançamento do agente 1.1.0 e integração com o Amazon EC2 Systems Manager	A versão 1.1.0 do CodeDeploy agente já está disponível. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente. Agora você pode usar o Amazon EC2 Systems Manager para gerenciar automaticamente a instalação e as atualizações do CodeDeploy seu agente no Amazon EC2 ou em instâncias locais. Para obter mais informações, consulte Instalar o CodeDeploy agente usando o Amazon EC2 Systems Manager.	30 de junho de 2020

CodeDeploy suporta o gerenciamento de blue/green implantações do Amazon ECS com CloudFormation	Agora você pode usar CloudFormation para gerenciar blue/green implantações do Amazon ECS por meio de. CodeDeploy Você gera a implantação definindo seus recursos verdes e azuis e especificando as configurações de roteamento e estabilização de tráfego a serem usadas no CloudFormation. Para obter mais informações, consulte Criar uma blue/green implantação do Amazon ECS por meio CloudFormation de.	19 de maio de 2020
CodeDeploy suporta a mudança ponderada de tráfego para implantações do Amazon ECS blue/green	CodeDeploy agora oferece suporte à mudança ponderada de tráfego para implantações do Amazon ECS. blue/green Você seleciona ou cria uma configuração de implantação para especificar o número de intervalos de deslocamento de tráfego na implantação e a porcentagem de tráfego a ser deslocada em cada intervalo. O tópico a seguir foi atualizado para refletir essa alteração: Configurações de implantação em uma plataforma de computação do Amazon ECS.	6 de fevereiro de 2020

Tópicos atualizados de segurança, autenticação e controle de acesso	As informações de segurança, autenticação e controle de acesso CodeDeploy foram organizadas em um novo capítulo de Segurança. Para obter mais informações, consulte Segurança.	26 de novembro de 2019
CodeDeploy suporta regras de notificação	Agora é possível usar regras de notificação para notificar os usuários sobre alterações importantes nas implantações. Para obter mais informações, consulte Criar uma regra de notificação.	5 de novembro de 2019
Tópicos atualizados	CodeDeploy agora está disponível na região Ásia-Pacífico (Hong Kong) (ap-east-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região. Você deve habilitar explicitamente o acesso a essa região. Para obter mais informações, consulte Gerenciando AWS regiões.	25 de abril de 2019

Tópicos atualizados	AWS CodeDeploy agora oferece suporte a blue/green implantações de um aplicativo em contêineres em um serviço do Amazon ECS. Um CodeDeploy aplicativo que usa a nova plataforma computacional do Amazon ECS implanta um aplicativo em contêineres em um novo conjunto de tarefas de substituição no mesmo serviço do Amazon ECS. Vários tópicos foram adicionados e atualizados para refletir essa mudança, incluindo visão geral das plataformas AWS CodeDeploy computacionais, implantações em uma plataforma computacional do Amazon ECS, AppSpec estrutura de arquivos para implantações do Amazon ECS e criação de um aplicativo para a implantação de um serviço do Amazon ECS (console).	27 de novembro de 2018
CodeDeploy Agente atualizado	O AWS CodeDeploy agente foi atualizado para a versão 1.0.1.1597. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	15 de novembro de 2018

Console atualizado	Os procedimentos neste guia foram atualizados para corresponder ao novo design do console do CodeDeploy.	30 de outubro de 2018
Nova versão mínima suportada do CodeDeploy agente	A versão mínima suportada do AWS CodeDeploy agente agora é 1.8.x. Para obter mais informações, consulte Histórico de versões do CodeDeploy agente.	7 de agosto de 2018

Atualizações anteriores

A tabela a seguir descreve alterações importantes em cada versão do Guia do usuário do AWS CodeDeploy antes de junho de 2018.

Alteração	Descrição	Alterado em
Atualizações de tópicos	CodeDeploy agora está disponível na região da Europa (Paris) (eu-west-3). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	19 de dezembro de 2017
Tópicos atualizados	CodeDeploy agora está disponível na região da China (Ningxia). Para usar serviços na região China (Pequim) ou China (Ningxia), é necessário ter uma conta e credenciais específicas para essas regiões. As contas e credenciais de outras AWS regiões não funcionam para as regiões de Pequim e Ningxia e vice-versa. Informações sobre alguns recursos para as regiões da China, como nomes de buckets do CodeDeploy Resource Kit e procedimentos de instalação de CodeDeploy agentes,	11 de dezembro de 2017

Alteração	Descrição	Alterado em
	não estão incluídas nesta edição do Guia CodeDeploy do usuário. Para obter mais informações: • CodeDeployem Primeiros passos AWS na região da China (Pequim) • CodeDeploy Guia do usuário para as regiões da China (versão em inglês | versão em chinês)	
Tópicos atualizados	CodeDeploy agora suporta a implantação de uma função Lambda. Uma implantação do AWS Lambda muda o tráfego recebido de uma função do Lambda existente para uma versão de função do Lambda atualizada. Você escolhe ou cria uma configuração de implantação para especificar o número de intervalos de mudança de tráfego na implantação e a porcentagem de tráfego a ser deslocada em cada intervalo. AWS Lambda as implantaç ões são suportadas pelo AWS Serverless Applicati on Model (AWS SAM) para que você possa usar uma preferência de implantação do AWS SAM para gerenciar a forma como o tráfego é deslocado durante uma implantaç ão. AWS Lambda Diversos tópicos foram adicionad os e atualizados para refletir essa alteração, incluindo Visão geral das CodeDeploy plataformas de computaçã o, Implantações em um AWS Plataforma de computação Lambda, Crie um AWS Implantação da Lambda Compute Platform (console), Crie um aplicativo para um AWS Lambda implantação de funções (console) e Adicionar um AppSpec arquivo para um AWS implantação do Lambda.	28 de novembro de 2017

Alteração	Descrição	Alterado em
Novo tópico	CodeDeploy agora oferece suporte a implantações diretamente em uma máquina ou instância local em que o CodeDeploy agente está instalado. Você pode testar localmente uma implantação e, se ela tiver erros, usar os registros de erros do CodeDeploy agente para depurá-los. Você também pode usar uma implantação local para testar a integridade de uma revisão do aplicativo, o conteúdo de um AppSpec arquivo e muito mais. Para obter mais informações, consulte Use o CodeDeploy agente para validar um pacote de implantação em uma máquina local.	16 de novembro de 2017
Tópicos atualizados	CodeDeploy o suporte para balanceadores de carga do Elastic Load Balancing em grupos de implantação foi expandido para incluir balanceadores de carga de rede para blue/green implantações e implantações locais. Agora, você pode escolher um Application Load Balancer, um Classic Load Balancer ou um Network Load Balancer para seu grupo de implantação. Os balanceadores de carga são necessários para blue/green implantações e opcionais para implantações locais. Diversos tópicos foram atualizados para refletir esse suporte, incluindo Integrating CodeDeploy with Elastic Load Balancing, Criar um aplicativo para uma implantação no local (console), Pré-requisitos de implantação, Integrating CodeDeploy with Elastic Load Balancing e Criar um aplicativo para uma implantação no local (console).	12 de setembro de 2017

Alteração	Descrição	Alterado em
Tópicos atualizados	CodeDeploy o suporte para balanceadores de carga do Elastic Load Balancing em grupos de implantação foi expandido para incluir balanceadores de carga de aplicativos para blue/green implantações e implantaç ões locais. Agora, você pode escolher entre um Applicati on Load Balancer e um Classic Load Balancer para seu grupo de implantação. Os balanceadores de carga são necessários para blue/green implantações e opcionais para implantações locais. Os tópicos incluindo Integrating CodeDeploy with Elastic Load Balancing, Crie um aplicativ o com CodeDeploy e Crie um grupo de implantação com CodeDeploy foram atualizados para refletir esse suporte adicional.	10 de agosto de 2017
Tópicos novos e atualizados	CodeDeploy agora oferece suporte ao uso de vários grupos de tags para identificar uniões e interseções de instâncias a serem incluídas em um grupo de implantação. Se você usar um grupo de tag única, qualquer instância identificada por pelo menos uma tag no grupo é incluída no grupo de implementação. Se você usar grupos de várias tags, apenas as instâncias que são identificadas por pelo menos uma tag em cada um dos grupos de tags serão incluídas. Para obter informações sobre o novo método para adicionar instâncias a um grupo de implantaç ão, consulte Tagging Instances for Deployments. Outros tópicos atualizados para refletir este suporte incluem Criar um aplicativo para uma implantação no local (console) , Criar um aplicativo para uma blue/green implantaç ão (console), Criar um grupo de implantação para uma implantação no local (console), Crie um grupo de implantaç ão para uma EC2/On-Premises blue/green implantação (console), Deployments e Etapa 5: criar um aplicativo e um grupo de implantação em Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub.	31 de julho de 2017

Alteração	Descrição	Alterado em
Tópico atualizado	Dois métodos adicionais para instalar o CodeDeploy agente em instâncias do Windows Server foram adicionados Instale o CodeDeploy agente para Windows Server. Além dos PowerShell comandos do Windows, agora estão disponíveis instruções para baixar o arquivo de instalação usando um link HTTPS direto e usando um comando de cópia do Amazon S3. Depois que o arquivo é baixado ou copiado para uma instância, é possível executar a instalação manualmente.	12 de julho de 2017
Tópicos atualizados	CodeDeploy melhorou a forma como gerencia conexões com GitHub contas e repositórios. Agora você pode criar e armazenar até 25 conexões com GitHub contas para associar CodeDeploy aplicativos a GitHub repositórios. Cada conexão pode dar suporte a vários repositórios. Você pode criar conexões com até 25 GitHub contas diferentes ou criar mais de uma conexão com uma única conta. Depois de conectar um aplicativo a uma GitHub conta, CodeDeploy gerencia as permissões de acesso necessárias sem exigir nenhuma ação adicional de sua parte. Atualizações para refletir esse suporte foram feitas em Especificar informações sobre uma revisão armazenada em um GitHub repositório, Integrando com CodeDeploy GitHub e Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub.	30 de maio de 2017

Alteração	Descrição	Alterado em
Tópicos atualizados	No passado, se o CodeDeploy agente detectasse arquivos em um local de destino que não faziam parte da revisão do aplicativo da implantação bem-sucedida mais recente, ele falharia na implantação atual por padrão. CodeDeploy agora fornece opções de como o agente manipula esses arquivos: falha na implantação, retém o conteúdo ou sobrescreve o conteúdo. Crie uma implantação com CodeDeployfoi atualizado para refletir esse suporte, e a nova seção Comportamento de reversão com conteúdo existente foi adicionadaReimplante e reverta uma implantação com CodeDeploy.	16 de maio de 2017
Tópicos atualizados	Um Classic Load Balancer no Elastic Load Balancing agora pode ser atribuído a um grupo de implantação usando o CodeDeploy console ou. AWS CLI Em uma implantação no local, um balanceador de carga impede que o tráfego da Internet seja direcionado para uma instância durante o processo, tornando-a novamente disponível para tráfego assim que essa implantação for concluída. Vários tópicos foram atualizados para refletir esse novo suporte, incluindo Integração com outros AWS serviços, Integrating CodeDeploy with Elastic Load Balancing, Criar um aplicativo para uma implantação no local (console), Criar um grupo de implantação para uma implantação no local (console) e AppSpec seção 'ganchos'. Foi adicionada uma nova seção ao guia de solução de problemas, Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação.	27 de abril de 2017

Alteração	Descrição	Alterado em
Tópicos atualizados	Um Classic Load Balancer no Elastic Load Balancing agora pode ser atribuído a um grupo de implantação usando o CodeDeploy console ou. AWS CLI Em uma implantação no local, um balanceador de carga impede que o tráfego da Internet seja direcionado para uma instância durante o processo, tornando-a novamente disponível para tráfego assim que essa implantação for concluída. Vários tópicos foram atualizados para refletir esse novo suporte, incluindo Integração com outros AWS serviços, Integrating CodeDeploy with Elastic Load Balancing, Criar um aplicativo para uma implantação no local (console), Criar um grupo de implantação para uma implantação no local (console) e AppSpec seção 'ganchos'. Foi adicionada uma nova seção ao guia de solução de problemas, Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação.	1 de maio de 2017

Alteração	Descrição	Alterado em
Tópicos atualizados	CodeDeploy agora está disponível na região da China (Pequim). Para usar serviços na região China (Pequim) ou China (Ningxia), é necessário ter uma conta e credenciais específicas para essas regiões. As contas e credenciais de outras AWS regiões não funcionam para as regiões de Pequim e Ningxia e vice-versa. Informações sobre alguns recursos para as regiões da China, como nomes de buckets do CodeDeploy Resource Kit e procedimentos de instalação de CodeDeploy agentes, não estão incluídas nesta edição do Guia CodeDeploy do usuário. Para obter mais informações: • CodeDeployem Introdução à AWS região da China (Pequim) • CodeDeploy Guia do usuário para as regiões da China (versão em inglês | versão em chinês)	29 de março de 2017

Alteração	Descrição	Alterado em
Tópicos novos e atualizados	Vários novos tópicos foram introduzidos para refletir o novo CodeDeploy suporte para blue/green implantações, um método de implantação no qual as instâncias em um grupo de implantação (o ambiente original) são substituídas por um conjunto diferente de instâncias (o ambiente substitut o). Visão geral de uma blue/green implantaçãofornece uma explicação de alto nível da blue/green metodologia usada por CodeDeploy. Novos tópicos adicionais incluem Criar um aplicativo para uma blue/green implantação (console) . Crie um grupo de implantação para uma EC2/On-Pr emises blue/green implantação (console) e Configure um balanceador de carga no Elastic Load Balancing para implantações do Amazon CodeDeploy EC2. Vários tópicos foram atualizados também, incluindo Crie uma implantação com CodeDeploy, Trabalhando com configurações de implantação em CodeDeploy, Crie um aplicativo com CodeDeploy, Trabalhando com grupos de implantação em CodeDeploy , Trabalhando com implantaç ões em CodeDeploy e AppSpec seção 'ganchos'.	25 de janeiro de 2017
Tópicos novos e atualizados	Um novo tópico,Usar o comando register-on-premises-instance (ARN de sessão do IAM) para registrar uma instância on-premises, descreve como autenticar e registrar instâncias locais usando credenciais temporári as atualizadas periodicamente geradas por meio de. AWS Security Token Service Essa abordagem fornece melhor suporte para grandes frotas de instâncias no local do que o uso exclusivo das credenciais de um usuário do IAM estático em cada instância. O Working with On-Premises Instances foi atualizado para refletir esse novo suporte.	28 de dezembro de 2016

Alteração	Descrição	Alterado em
Tópicos atualizados	CodeDeploy agora está disponível na região Europa (Londres) (eu-west-2). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	13 de dezembro de 2016
Tópicos atualizados	CodeDeploy agora está disponível na região do Canadá (Central) (ca-central-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	8 de dezembro de 2016
Tópicos atualizados	CodeDeploy agora está disponível na região Leste dos EUA (Ohio) (us-east-2). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	17 de outubro de 2016
Novos tópicos	Uma nova seção, Autenticação e Controle de Acesso, fornece informações abrangentes sobre o uso AWS Identity and Access Management (IAM) e ajuda CodeDeploy a proteger o acesso aos seus recursos por meio do uso de credenciais. Essas credenciais fornecem as permissões necessárias para acessar AWS recursos, como recuperar revisões de aplicativos dos buckets do Amazon S3 e ler as tags nas instâncias do Amazon EC2.	11 de outubro de 2016
Tópico atualizado	Atualize o CodeDeploy agente no Windows Serverfoi atualizado para refletir a disponibilidade do novo atualizador de CodeDeploy agentes para Windows Server. Quando instalado em uma instância do Windows Server, o atualizador verificará periodicamente se há novas versões. Quando uma nova versão for detectada, o atualizador desinstalará a versão atual do agente, se instalada, antes de instalar a versão mais recente.	4 de outubro de 2016

Alteração	Descrição	Alterado em
Tópicos atualizados	CodeDeploy agora se integra aos CloudWatch alarmes da Amazon, possibilitando interromper uma implantação se houver uma alteração no estado de um alarme especific ado por vários períodos consecutivos, conforme especific ado no limite do alarme. CodeDeploy agora também suporta a reversão automátic a de uma implantação se determinadas condições forem atendidas, como uma falha na implantação ou um alarme ativado. Uma série de tópicos foram atualizados para refletir essas alterações, incluindo Crie um aplicativo com CodeDeploy, Crie um grupo de implantação com CodeDeploy, Altere as configurações do grupo de implantação com CodeDeplo y, Deployments, Reimplante e reverta uma implantação com CodeDeploy e Integrações de produtos e serviços com CodeDeploy, junto com um novo tópico, Monitorando implantações com CloudWatch alarmes em CodeDeploy.	15 de setembro de 2016
Tópicos novos e atualizados	CodeDeploy agora fornece integração com a Amazon CloudWatch Events. Agora você pode usar CloudWatc h Eventos para iniciar uma ou mais ações quando forem detectadas alterações no estado de uma implantação ou no estado de uma instância que pertence a um grupo de CodeDeploy implantação. Você pode incorporar ações que invocam AWS Lambda funções; que publicam nos streams do Kinesis ou nos tópicos do Amazon SNS; que enviam mensagens para as filas do Amazon SQS; ou que, por sua vez, acionam ações de alarme. CloudWatch Para obter mais informações, consulte Monitoramento de implantaç ões com Amazon Events CloudWatch.	9 de setembro de 2016

Alteração	Descrição	Alterado em
Atualizações de tópicos	Os tópicos Integrating CodeDeploy with Elastic Load Balancing Integração com outros AWS serviços foram atualizados para refletir uma opção adicional de balanceam ento de carga. CodeDeploy agora oferece suporte ao Classic Load Balancer e ao Application Load Balancer disponíveis no Elastic Load Balancing.	11 de agosto de 2016
Atualizações de tópicos	CodeDeploy agora está disponível na região Ásia-Pacífico (Mumbai) (ap-south-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	27 de junho de 2016
Atualizações de tópicos	CodeDeploy agora está disponível na região Ásia-Pacífico (Seul) (ap-northeast-2). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região. O sumário foi reorganizada para incluir seções para instâncias, configurações de implantação, aplicativos, grupos de implantação, revisões e implantações. Uma nova seção foi adicionada para tutoriais do CodeDeplo y. Para facilitar o uso, vários tópicos longos, incluindo CodeDeploy AppSpec referência de arquivo e Solução de problemas CodeDeploy, foram divididos em tópicos mais curtos. Informações de configuração para o agente CodeDeploy foi movida para um novo tópico, CodeDeploy referência de configuração do agente	15 de junho de 2016

Alteração	Descrição	Alterado em
Tópicos novos e atualizados	Códigos de erro para AWS CodeDeploy fornece informaçõ es sobre algumas das mensagens de erro que podem ser exibidas quando a implantação do CodeDeploy falhar. As seguintes seções em Solução de problemas CodeDeplo y foram atualizadas para ajudar na resolução de problemas de implantação: • As instâncias do EC2 em um grupo Amazon EC2 Auto Scaling falham ao ser iniciadas e recebem o erro "Heartbeat Timeout" • Evite associar vários grupos de implantação a um único grupo de Amazon EC2 Auto Scaling	20 de abril de 2016
Atualizações de tópicos	CodeDeploy agora está disponível na região da América do Sul (São Paulo) (sa-east-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região. Trabalhando com o CodeDeploy agentefoi atualizado para refletir a nova opção de configuração:max_revisions: , que você usa para especificar o número de revisões de aplicativos para um grupo de implantação que você deseja que o agente arquive. CodeDeploy	10 de março de 2016

Alteração	Descrição	Alterado em
Tópicos novos e atualizados	CodeDeploy agora oferece suporte à adição de gatilhos a um grupo de implantação para receber notificações sobre eventos relacionados a implantações ou instâncias nesse grupo de implantação. Essas notificações são enviadas aos destinatários inscritos em um tópico do Amazon Simple Notification Service que você incluiu como parte da ação do gatilho. Você também pode usar dados JSON que são criados quando um gatilho é disparado no seu próprio fluxo de trabalho de notificação personalizado. Para obter mais informações, consulte Monitoring Deployments with Amazon SNS Event Notifications. Procedimentos foram atualizados para refletir o novo design da página Application details. A seção As implantações não falham por até uma hora quando uma instância é encerrada durante uma implantação em Solução de problemas CodeDeploy foi atualizada. O CodeDeploy cotas foi atualizado para refletir limites revisados para o número de grupos de implantação que podem ser associados a um único aplicativo, os valores permitidos para configurações mínimas de instâncias íntegras; e versões necessárias do AWS SDK para Ruby.	17 de fevereiro de 2016

Alteração	Descrição	Alterado em
Tópicos novos e atualizados	CodeDeploy agora está disponível na região Oeste dos EUA (Norte da Califórnia) (us-west-1). Vários tópicos, incluindo aqueles que contêm instruções para configura r o CodeDeploy agente, foram atualizados para refletir a adição dessa nova região. O Escolha um tipo de CodeDeploy repositório lista e descreve os tipos de repositórios agora com suporte pelo CodeDeploy. Esse novo tópico será atualizado à medida que houver suporte para outros tipos de repositório. Gerenciando as operações CodeDeploy do agentefoi atualizado com informações sobre o novo `.version` arquivo adicionado às instâncias para relatar a versão atual do CodeDeploy agente, bem como informações sobre as versões suportadas do agente. Foi adicionado realce de sintaxe para amostras de código, incluindo exemplos JSON e YAML, ao guia do usuário. Adicionar um arquivo de especificação do aplicativo a uma revisão do CodeDeploy foi reorganizado como instruções passo a passo.	20 de janeiro de 2016
Novo tópico	Implemente um aplicativo em outro AWS accountdescreve os requisitos de configuração e o processo para iniciar implantações que pertencem a outra conta da sua organização, sem precisar de um conjunto completo de credenciais para essa outra conta. Isso é muito útil para organizações que usam várias contas para diferente s finalidades, como uma associada a ambientes de desenvolvimento e teste e outra associada ao ambiente de produção.	30 de dezembro de 2015

Alteração	Descrição	Alterado em
Atualização de tópico	O tópico Integrações de produtos e serviços com CodeDeploy foi reprojetado. Agora, ele inclui uma seção para exemplos de integração da comunidade, com listas de postagens de blog e exemplos de vídeos relacionados a integrações com o CodeDeploy.	16 de dezembro de 2015
Atualizações de tópicos	CodeDeploy agora está disponível na região Ásia-Pacífico (Cingapura) (ap-southeast-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	9 de dezembro de 2015
Atualizações de tópicos	O Trabalhando com o CodeDeploy agente foi atualizado para refletir a nova opção `:proxy_uri:` no arquivo de configuração do agente CodeDeploy. Foram feitas atualizações em CodeDeploy AppSpec referência de arquivo com informações sobre como usar uma nova variável de ambiente, `DEPLOYMENT_GROUP_ID`, que scripts de gancho podem acessar durante um evento de ciclo de vida de implantação.	1 de dezembro de 2015
Atualização de tópico	Etapa 2: criar uma função de serviço para CodeDeployfoi atualizado para refletir um novo procedimento para criar uma função de serviço CodeDeploy e incorporar outras melhorias.	13 de novembro de 2015
Atualizações de tópicos	CodeDeploy agora está disponível na região da Europa (Frankfurt) (eu-central-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região. O tópico Solução de problemas CodeDeploy foi atualizado com informações sobre como garantir que as configurações de tempo nas instâncias estejam precisas.	19 de outubro de 2015

Alteração	Descrição	Alterado em
Novos tópicos	CloudFormation modelos para CodeDeploy referência foi publicado para refletir o novo CloudFormation apoio CodeDeploy às ações. Criação de um tópico Primary Components e introdução da definição de uma revisão de destino.	1 de outubro de 2015
Atualizações de tópicos	Atualização de Crie um grupo de implantação com CodeDeploy para refletir a capacidade de localizar instâncias para um grupo de implantação usando pesquisas com curingas. Atualização de Instance Health para esclarecer o conceito de instâncias íntegras mínimas.	31 de agosto de 2015
Atualizações de tópicos	CodeDeploy agora está disponível na região Ásia-Pacífico (Tóquio) (ap-northeast-1). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessa nova região.	19 de agosto de 2015
Atualizações de tópicos	CodeDeploy agora oferece suporte a implantações em instâncias locais do Red Hat Enterprise Linux (RHEL) e instâncias do Amazon EC2. Para saber mais, consulte os seguintes tópicos: • Sistemas operacionais suportados pelo CodeDeploy agente • Trabalhando com instâncias para CodeDeploy • Tutorial: Implante WordPress em uma instância do Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux e Linux, macOS ou Unix) • Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux)	23 de junho de 2015

Alteração	Descrição	Alterado em
Atualização de tópico	CodeDeploy agora fornece um conjunto de variáveis de ambiente que seus scripts de implantação podem usar durante as implantações. Essas variáveis de ambiente incluem informações como o nome do CodeDeploy aplicativo atual, grupo de implantação e evento do ciclo de vida da implantação, bem como o identificador de CodeDeploy implantação atual. Para obter mais informaçõ es, consulte o final da seção AppSpec seção 'ganchos' no CodeDeploy AppSpec referência de arquivo.	29 de maio de 2015
Atualizações de tópicos	CodeDeploy agora fornece um conjunto de políticas AWS gerenciadas no IAM que você pode usar em vez de criar manualmente as políticas equivalentes por conta própria. Isso inclui: • Uma política para permitir que um usuário registre revisões CodeDeploy somente com e depois as implante por meio CodeDeploy de. • Uma política para fornecer ao usuário acesso total aos CodeDeploy recursos. • Uma política para fornecer ao usuário acesso somente para CodeDeploy leitura aos recursos. • Uma política a ser anexada a uma função de serviço para que CodeDeploy possa identificar instâncias do Amazon EC2 por meio de suas tags do Amazon EC2, tags de instância local ou nomes de grupos do Amazon EC2 Auto Scaling e implantar revisões de aplicativos nelas adequadamente. Para obter mais informações, consulte a seção Customer-managed exemplos de políticas em Autenticação e controle de acesso.	29 de maio de 2015

Alteração	Descrição	Alterado em
Atualizações de tópicos	CodeDeploy agora está disponível na região da Europa (Irlanda) (eu-west-1) e na região Ásia-Pacífico (Sydney) (ap-southeast-2). Vários tópicos, incluindo aqueles que contêm instruções para configurar o CodeDeploy agente, foram atualizados para refletir a disponibilidade dessas novas regiões.	7 de maio de 2015
Novos tópicos	CodeDeploy agora oferece suporte a implantações em instâncias locais e instâncias do Amazon EC2. Os seguintes tópicos foram adicionados para descrever esse novo suporte: • Working with On-Premises Instances • Tutorial: Implantar um aplicativo em uma instância local com CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux) • Working with On-Premises Instances	2 de abril de 2015
Novo tópico	CodeDeploy recursos foi adicionado.	2 de abril de 2015

Alteração	Descrição	Alterado em
Atualização de tópico	Atualização de Solução de problemas CodeDeploy: • Uma nova seção Long-running processos podem fazer com que as implantações falhem descreve as etapas que você pode realizar para identificar e solucionar falhas de implantação devido a processos de execução longa. • A Solução de problemas gerais do Amazon EC2 Auto Scaling seção foi atualizada para mostrar que CodeDeploy aumentou sua lógica de tempo limite do Amazon EC2 Auto Scaling CodeDeploy para o agente de cinco minutos para uma hora. • Uma nova seção Ganchos do ciclo de vida do Amazon EC2 Auto Scaling podem fazer com que as implantações automáticas em grupos do Amazon EC2 Auto Scaling parem ou falhem. descreve as etapas que você pode seguir para identificar e resolver implantações automátic as com falhas em grupos do Amazon EC2 Auto Scaling.	2 de abril de 2015

Alteração	Descrição	Alterado em
Atualizações de tópicos	Os seguintes tópicos foram atualizados para refletir novas recomendações para criar suas próprias políticas personalizadas e, em seguida, anexá-las a usuários e funções no IAM;: • Configurar uma instância do Amazon EC2 para trabalhar com CodeDeploy • Etapa 4: criar um perfil de instância do IAM para as suas instâncias do Amazon EC2 • Etapa 2: criar uma função de serviço para CodeDeploy Duas seções foram adicionadas a Solução de problemas CodeDeploy: • Lista de verificação geral para solução de problemas • PowerShell Os scripts do Windows não usam a versão de 64 bits do Windows PowerShell por padrão Atualização da seção AppSpec seção 'ganchos' no CodeDeploy AppSpec referência de arquivo para descrever com mais precisão os eventos de ciclo de vida de implantação disponíveis.	12 de fevereiro de 2015
Atualizações de tópicos	Uma nova seção foi adicionada a Solução de problemas CodeDeploy: As instâncias do EC2 em um grupo Amazon EC2 Auto Scaling falham ao ser iniciadas e recebem o erro "Heartbeat Timeout". Uma CloudBees seção foi adicionada Integrações de produtos e serviços com CodeDeploy a.	28 de janeiro de 2015

Alteração	Descrição	Alterado em
Atualizações de tópicos	As seguintes seções foram adicionadas a Solução de problemas CodeDeploy: • Usar alguns editores de texto para criar AppSpec arquivos e scripts de shell pode causar falhas nas implantações • O uso do Finder no macOS para agrupar uma revisão de aplicativo pode causar falhas na implantação • Solução de problemas de falha ApplicationStop ou BeforeBlockTraffic evento do ciclo AfterBlockTraffic de vida da implantação • Solução de problemas em um evento DownloadB undle de ciclo de vida de implantação com falha com UnknownError: não aberto para leitura • Solução de problemas gerais do Amazon EC2 Auto Scaling	20 de janeiro de 2015
Novos tópicos	Atualização da seção Integrações de produtos e serviços com CodeDeploy para incluir os seguintes tópicos: • Integração CodeDeploy com o Amazon EC2 Auto Scaling • Tutorial: Use CodeDeploy para implantar um aplicativo em um grupo do Auto Scaling • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Integrando com CodeDeploy GitHub • Tutorial: Use CodeDeploy para implantar um aplicativo a partir de GitHub	9 de janeiro de 2015

Alteração	Descrição	Alterado em
Atualizações de tópicos	• A seção Implemente automaticamente a partir de CodePipeline com CodeDeploy foi adicionada a Integrando com CodeDeploy GitHub. Agora você pode acionar automaticamente uma implantação a partir de um GitHub repositório sempre que o código-fonte desse repositório for alterado. • A seção Solucionar problemas do Amazon EC2 Auto Scaling foi adicionada a Solução de problemas CodeDeploy. Essa nova seção descreve como soluciona r problemas comuns com a implantação em grupos do Amazon EC2 Auto Scaling . • A nova subseção "Exemplos de arquivos" foi adicionada à seção AppSpec seção 'arquivos' (somente EC2/On-Pr emises implantações) de CodeDeploy AppSpec referênci a de arquivo. Essa nova subseção inclui vários exemplos de como usar a `files` seção de um AppSpec arquivo para CodeDeploy instruir a cópia de arquivos ou pastas específicos para locais específicos em uma instância do Amazon EC2 durante uma implantação.	8 de janeiro de 2015
Novo tópico	Monitoring Deploymentsfoi adicionado. CodeDeploy é integrado com AWS CloudTrail, um serviço que captura chamadas de API feitas por ou em nome de CodeDeplo y sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar.	17 de dezembro de 2014
Lançamento público inicial	Essa é o lançamento inicial público do CodeDeploy User guide (Guia de usuário).	12 de novembro de 2014

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.