Panduan Pengguna

AWS CodeDeploy



Versi API 2014-10-06

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

AWS CodeDeploy: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu CodeDeploy? 1
Manfaat dari AWS CodeDeploy 2
Ikhtisar CodeDeploy platform komputasi 3
Ikhtisar jenis CodeDeploy penerapan 10
Ikhtisar penerapan di tempat 12
Ikhtisar blue/green penerapan 13
Kami ingin mendengar pendapat Anda 17
Komponen utama 17
Aplikasi 18
Platform komputasi 18
Konfigurasi deployment 19
Grup deployment 20
Tipe deployment 20
Tipe deployment 21
Revisi 21
Peran layanan 22
Revisi target 22
Komponen lainnya 22
Deployment 23
Penerapan pada AWS Platform Komputasi Lambda 23
Deployment di Platform Komputasi Amazon ECS 26
Penerapan pada Platform Komputasi EC2/On-Premises 39
File spesifikasi aplikasi 46
AppSpec file di Platform Komputasi Amazon ECS 46
AppSpec berkas pada sebuah AWS Lambda platform komputasi 47
AppSpec file pada platform EC2/on-premises komputasi 47
Bagaimana CodeDeploy agen menggunakan AppSpec file 47
Memulai 49
Langkah 1: Menyiapkan 49
Mendaftar untuk Akun AWS 49
Langkah 2: Buat peran layanan 50
Buat peran layanan (konsol) 52
Buat peran layanan (CLI) 55
Dapatkan peran layanan ARN (konsol) 58

Dapatkan peran layanan ARN (CLI) 58
Langkah 3: Batasi izin CodeDeploy pengguna 58
Langkah 4: Buat profil instans IAM 62
Buat profil instans IAM untuk instans Amazon EC2 (CLI) 63
Membuat profil instans IAM untuk instans Amazon EC2 (konsol) 67
Integrasi produk dan layanan 71
Integrasi dengan AWS layanan lain 71
Amazon EC2 Auto Scaling 79
Elastic Load Balancing 88
Integrasi dengan produk dan layanan mitra 92
GitHub 98
Contoh integrasi dari komunitas 102
Unggahan blog 102
Tutorial 104
Tutorial: Terapkan WordPress ke instance non-Windows 104
Langkah 1: Luncurkan instans Amazon EC2 105
Langkah 2: Konfigurasikan konten sumber Anda 108
Langkah 3: Unggah aplikasi Anda ke Amazon S3 113
Langkah 4: Menyebarkan aplikasi Anda 118
Langkah 5: Perbarui dan terapkan ulang aplikasi Anda 124
Langkah 6: Bersihkan 128
Tutorial: Menyebarkan aplikasi Hello World ke instance Windows Server 132
Langkah 1: Luncurkan instans Amazon EC2 133
Langkah 2: Konfigurasikan konten sumber Anda 135
Langkah 3: Unggah aplikasi Anda ke Amazon S3 138
Langkah 4: Menyebarkan aplikasi Anda 143
Langkah 5: Perbarui dan terapkan ulang aplikasi Anda 148
Langkah 6: Bersihkan 151
Tutorial: Menerapkan aplikasi ke instans lokal 155
Prasyarat 155
Langkah 1: Konfigurasikan instans lokal 156
Langkah 2: Buat contoh revisi aplikasi 156
Langkah 3: Bundel dan unggah revisi aplikasi Anda ke Amazon S3 161
Langkah 4: Terapkan revisi aplikasi Anda 161
Langkah 5: Verifikasi penerapan Anda 161
Langkah 6: Bersihkan Sumber Daya 162

Tutorial: Terapkan ke grup Auto Scaling 164
Prasyarat 165
Langkah 1: Buat dan konfigurasikan grup Auto Scaling 165
Langkah 2: Menyebarkan aplikasi ke grup Auto Scaling 171
Langkah 3: Periksa hasil Anda 182
Langkah 4: Tingkatkan jumlah instans Amazon EC2 di grup Auto Scaling 184
Langkah 5: Periksa kembali hasil Anda 185
Langkah 6: Bersihkan 187
Tutorial: Menyebarkan aplikasi dari GitHub 189
Prasyarat 190
Langkah 1: Siapkan GitHub akun 191
Langkah 2: Buat GitHub repositori 191
Langkah 3: Unggah contoh aplikasi ke GitHub repositori Anda 193
Langkah 4: Menyediakan sebuah instance 198
Langkah 5: Buat grup aplikasi dan penyebaran 199
Langkah 6: Menyebarkan aplikasi ke instance 201
Langkah 7: Pantau dan verifikasi penyebaran 205
Langkah 8: Membersihkan 206
Tutorial: Menyebarkan aplikasi ke Amazon ECS 208
Prasyarat 210
Langkah 1: Perbarui aplikasi Amazon ECS Anda 211
Langkah 2: Buat AppSpec file 212
Langkah 3: Gunakan CodeDeploy konsol untuk menyebarkan aplikasi Anda 213
Langkah 4: Membersihkan 217
Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi 217
Prasyarat 220
Langkah 1: Buat pendengar tes 220
Langkah 2: Perbarui aplikasi Amazon ECS Anda 221
Langkah 3: Buat fungsi Lambda kait siklus hidup 221
Langkah 4: Perbarui AppSpec file Anda 224
Langkah 5: Gunakan CodeDeploy konsol untuk menyebarkan layanan Amazon ECS Anda 225
Langkah 6: Lihat output fungsi kait Lambda Anda di Log CloudWatch 227
Langkah 7: Bersihkan 228
Tutorial: Menyebarkan fungsi Lambda menggunakan SAM AWS 229
Prasyarat 230

Langkah 1: Siapkan infrastruktur Anda 230
Langkah 2: Perbarui fungsi Lambda 245
Langkah 3: Menyebarkan fungsi Lambda yang diperbarui 248
Langkah 4: Lihat hasil penerapan 250
Langkah 5: Bersihkan 253
Bekerja dengan CodeDeploy agen 254
Sistem operasi yang didukung oleh CodeDeploy agen 255
Sistem operasi Amazon EC2 AMI yang didukung 255
Sistem operasi lokal yang didukung 255
Protokol komunikasi dan port untuk CodeDeploy agen 256
Riwayat versi CodeDeploy agen 257
Mengelola CodeDeploy proses 275
Revisi aplikasi dan pembersihan file log 275
File yang diinstal oleh CodeDeploy agen 276
Mengelola operasi CodeDeploy agen 279
Verifikasi CodeDeploy agen sedang berjalan 280
Tentukan versi CodeDeploy agen 283
Instal CodeDeploy agen 285
Perbarui CodeDeploy agen 299
Copot instalan agen CodeDeploy 302
Kirim log CodeDeploy agen ke CloudWatch 303
Bekerja dengan instans 309
Membandingkan instans Amazon EC2 dengan instans lokal 309
Tugas instans untuk CodeDeploy 311
Menandai instance untuk penerapan CodeDeploy 312
Contoh 1: Grup tag tunggal, tag tunggal 313
Contoh 2: Grup tag tunggal, beberapa tag 314
Contoh 3: Beberapa grup tag, tag tunggal 316
Contoh 4: Beberapa grup tag, beberapa tag 318
Bekerja dengan instans Amazon EC2 322
Buat instans Amazon EC2 untuk CodeDeploy 322
Buat instans Amazon EC2 (template)CloudFormation 329
Konfigurasikan instans Amazon EC2 340
Bekerja dengan instans on-premise 344
Prasyarat untuk mengonfigurasi instans lokal 345
Mendaftarkan instans on-premise 347

Mengelola operasi instans lokal 382
Tampilkan detail instans 390
Lihat detail contoh (konsol) 390
Lihat detail contoh (CLI) 391
Kesehatan contoh 391
Status kondisi 392
Tentang jumlah minimum contoh sehat 393
Tentang jumlah minimum instans sehat per Availability Zone 398
Bekerja dengan konfigurasi deployment 400
Konfigurasi penerapan pada platform komputasi EC2/on-premises 400
Konfigurasi penerapan yang telah ditentukan sebelumnya 400
Konfigurasi deployment pada platform komputasi Amazon ECS 405
Konfigurasi penerapan yang telah ditentukan sebelumnya untuk Amazon ECS 405
Konfigurasi penerapan untuk CloudFormation blue/green penerapan (Amazon ECS) 406
Konfigurasi penerapan pada sebuah AWS Lambda platform komputasi 407
Konfigurasi penerapan yang telah ditentukan untuk Lambda 407
Topik 408
Buat konfigurasi penerapan 408
Buat konfigurasi penerapan (konsol) 409
Buat konfigurasi penerapan ()AWS CLI 411
Lihat detail konfigurasi penerapan 413
Lihat detail konfigurasi penerapan (konsol) 413
Lihat konfigurasi penerapan (CLI) 414
Hapus konfigurasi penerapan 414
Bekerja dengan aplikasi 415
Membuat aplikasi 416
Buat aplikasi untuk penyebaran di tempat (konsol) 418
Buat aplikasi untuk blue/green penyebaran (konsol) 421
Membuat aplikasi untuk penyebaran layanan Amazon ECS (konsol) 426
Buat aplikasi untuk AWS Lambda penyebaran fungsi (konsol) 428
Buat aplikasi (CLI) 430
Lihat detail aplikasi 430
Lihat detail aplikasi (konsol) 430
Lihat detail aplikasi (CLI) 431
Membuat aturan notifikasi 431
Ganti nama aplikasi 434

Menghapus sebuah aplikasi ... 434
Hapus aplikasi (konsol) ... 435
Hapus aplikasi (AWS CLI) ... 435
Bekerja dengan grup deployment ... 436
Grup deployment dalam deployment platform komputasi Amazon ECS ... 436
Grup penyebaran di AWS Lambda komputasi penyebaran platform ... 436
Grup penyebaran dalam penerapan EC2/On-Premises Compute Platform ... 436
Topik ... 437
Buat grup penyebaran ... 438
Buat grup penerapan untuk penerapan di tempat (konsol) ... 438
Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol) ... 442
Membuat grup penerapan untuk penyebaran Amazon ECS (konsol) ... 448
Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2 ... 449
Siapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS ... 451
Buat grup penyebaran (CLI) ... 456
Lihat detail grup penyebaran ... 457
Lihat detail grup penerapan (konsol) ... 457
Lihat detail grup penerapan (CLI) ... 458
Ubah setelan grup penerapan ... 459
Ubah pengaturan grup penerapan (konsol) ... 459
Ubah pengaturan grup penyebaran (CLI) ... 460
Mengonfigurasi opsi lanjutan untuk grup deployment ... 461
Menghapus grup penyebaran ... 465
Menghapus grup penyebaran (konsol) ... 465
Hapus grup penyebaran (CLI) ... 466
Bekerja dengan revisi aplikasi ... 467
Rencanakan revisi ... 467
Menambahkan AppSpec File ... 468
Menambahkan AppSpec file untuk penyebaran Amazon ECS ... 468
Menambahkan AppSpec file untuk AWS Penyebaran Lambda ... 471
Menambahkan AppSpec file untuk EC2/On-Premises penerapan ... 473
Pilih jenis repositori ... 477
Dorong revisi ... 480
Dorong revisi menggunakan AWS CLI ... 482

Lihat detail revisi aplikasi ... 484
Lihat detail revisi aplikasi (konsol) ... 485
Lihat detail revisi aplikasi (CLI) ... 485
Daftarkan revisi aplikasi ... 486
Daftarkan revisi di Amazon S3 dengan CodeDeploy (CLI) ... 487
Daftarkan revisi GitHub dengan CodeDeploy (CLI) ... 488
Bekerja dengan deployment ... 489
Membuat deployment ... 490
Prasyarat penyebaran ... 491
Membuat penerapan Platform Komputasi Amazon ECS (konsol) ... 494
Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol) ... 495
Membuat penerapan Platform EC2/On-Premises Komputasi (konsol) ... 497
Membuat penerapan Platform Komputasi Amazon ECS (CLI) ... 502
Buat sebuah AWS Penerapan Platform Komputasi Lambda (CLI) ... 503
Membuat penerapan Platform EC2/On-Premises Komputasi (CLI) ... 505
Buat blue/green penyebaran Amazon ECS melalui CloudFormation ... 509
Lihat detail penerapan ... 512
Lihat detail penerapan (konsol) ... 513
Lihat detail penerapan (CLI) ... 514
Lihat data log penyebaran ... 514
Lihat data file log di CloudWatch konsol Amazon ... 515
Lihat file log pada sebuah instance ... 515
Hentikan penerapan ... 518
Hentikan penerapan (konsol) ... 519
Hentikan penerapan (CLI) ... 520
Menerapkan ulang dan memutar kembali penerapan ... 520
Rollback otomatis ... 521
Rollback manual ... 521
Alur kerja rollback dan redeployment ... 521
Perilaku rollback dengan konten yang ada ... 523
Menyebarkan aplikasi di akun yang berbeda AWS ... 527
Langkah 1: Buat bucket S3 di salah satu akun ... 527
Langkah 2: Berikan izin bucket Amazon S3 ke profil instans IAM akun produksi ... 527
Langkah 3: Buat sumber daya dan peran lintas akun di akun produksi ... 529
Langkah 4: Unggah revisi aplikasi ke bucket Amazon S3 ... 530
Langkah 5: Asumsikan peran lintas akun dan gunakan aplikasi ... 530

Validasi paket penerapan pada mesin lokal ... 531
Prasyarat ... 531
Buat penerapan lokal ... 534
Contoh ... 537
Memantau deployment ... 539
Alat otomatis ... 539
Alat manual ... 541
Memantau penerapan dengan alat Amazon CloudWatch ... 542
Memantau penyebaran dengan alarm CloudWatch ... 542
Memantau penerapan dengan Amazon Events CloudWatch ... 544
Memantau penyebaran dengan AWS CloudTrail ... 547
CodeDeploy informasi di CloudTrail ... 547
Memahami entri file CodeDeploy log ... 548
Memantau penerapan dengan notifikasi acara Amazon SNS ... 549
Berikan izin Amazon SNS ke peran layanan ... 551
Buat pemicu untuk suatu CodeDeploy acara ... 552
Mengedit pemicu dalam grup penerapan ... 559
Menghapus pemicu dari grup penerapan ... 561
Format data JSON untuk pemicu ... 562
Keamanan ... 564
Perlindungan Data ... 565
Privasi lalu lintas antarjaringan ... 566
Enkripsi Saat Diam ... 566
Enkripsi Saat Bergerak ... 566
Manajemen kunci enkripsi ... 567
Manajemen Identitas dan Akses ... 567
Audiens ... 567
Mengautentikasi dengan Identitas ... 568
Mengelola Akses menggunakan Kebijakan ... 569
Bagaimana AWS CodeDeploy bekerja dengan IAM ... 570
AWS kebijakan terkelola (standar) untuk CodeDeploy ... 575
CodeDeploy update ke AWS kebijakan terkelola ... 584
Identity-based contoh kebijakan ... 587
Pemecahan Masalah ... 595
CodeDeploy referensi izin ... 596
Cross-service pencegahan wakil bingung ... 605

Respons Insiden ... 608
Audit semua interaksi dengan CodeDeploy ... 608
Peringatan dan manajemen insiden ... 608
Validasi Kepatuhan ... 609
Ketahanan ... 609
Keamanan Infrastruktur ... 610
Referensi ... 611
AppSpec referensi file ... 611
AppSpec file pada platform komputasi Amazon ECS ... 612
AppSpec berkas pada sebuah AWS Lambda platform komputasi ... 612
AppSpec file pada platform EC2/on-premises komputasi ... 612
AppSpec Struktur berkas ... 613
AppSpec Contoh berkas ... 659
AppSpec Spasi file ... 665
Validasi AppSpec File dan lokasi file Anda ... 666
Referensi konfigurasi agen ... 667
Topik terkait ... 678
CloudFormation referensi template ... 678
Gunakan CodeDeploy dengan Amazon Virtual Private Cloud ... 681
Ketersediaan ... 682
Buat titik akhir VPC untuk CodeDeploy ... 685
Konfigurasikan CodeDeploy izin agen dan IAM ... 686
Referensi kit sumber daya ... 686
Nama bucket kit sumber daya menurut Wilayah ... 687
Isi kit sumber daya ... 689
Menampilkan daftar file kit sumber daya ... 691
Unduh file kit sumber daya ... 693
Kuota ... 696
Pemecahan masalah ... 704
Pemecahan masalah umum ... 704
Daftar periksa pemecahan masalah umum ... 705
CodeDeploy sumber daya penerapan hanya didukung di beberapa AWS Wilayah ... 706
Prosedur dalam panduan ini tidak cocok dengan CodeDeploy konsol ... 707
Peran IAM yang diperlukan tidak tersedia ... 707
Menggunakan beberapa editor teks untuk membuat AppSpec file dan skrip shell dapat menyebabkan penerapan gagal ... 707

Menggunakan Finder di macOS untuk menggabungkan revisi aplikasi dapat menyebabkan penerapan gagal 708
Memecahkan masalah EC2/On-Premises penerapan 708
CodeDeploy CommandPoller kesalahan kredensialnya hilang plugin 709
Penerapan gagal dengan pesan “Validasi pesan yang ditandatangani PKCS7 gagal” 710
Penerapan atau pemindahan file yang sama ke lokasi instance yang sama gagal dengan kesalahan “Penerapan gagal karena file tertentu sudah ada di lokasi ini” 710
Jalur file yang panjang menyebabkan kesalahan “Tidak ada file atau direktori seperti itu” 713
Long-running proses dapat menyebabkan penerapan gagal 714
Memecahkan masalah peristiwa AllowTraffic siklus hidup yang gagal tanpa kesalahan yang dilaporkan dalam log penerapan 715
Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan 716
Memecahkan masalah peristiwa siklus hidup DownloadBundle penerapan yang gagal dengan UnknownError: tidak dibuka untuk dibaca 717
Memecahkan masalah semua peristiwa siklus hidup yang dilewati kesalahan 718
PowerShell Skrip Windows gagal menggunakan versi Windows 64-bit PowerShell secara default 720
Memecahkan masalah penerapan Amazon ECS 721
Waktu tunggu terjadi saat menunggu set tugas pengganti 721
Waktu tunggu terjadi saat menunggu pemberitahuan dilanjutkan 722
Peran IAM tidak memiliki izin yang cukup 723
Waktu penerapan habis saat menunggu panggilan balik status 723
Penerapan gagal karena satu atau lebih fungsi validasi peristiwa siklus hidup gagal 724
ELB tidak dapat diperbarui karena kesalahan berikut: Kelompok target tugas utama harus berada di belakang pendengar 724
Penerapan saya terkadang gagal saat menggunakan Auto Scaling 725
Hanya ALB yang mendukung perutean lalu lintas bertahap, gunakan perutean AllAtOnce lalu lintas sebagai gantinya saat Anda grup create/update Deployment 726
Meskipun penerapan saya berhasil, set tugas pengganti gagal dalam pemeriksaan kesehatan Elastic Load Balancing, dan aplikasi saya mati 727
Dapatkah saya melampirkan beberapa penyeimbang beban ke grup penerapan? 728
Dapatkah saya melakukan CodeDeploy blue/green penerapan tanpa penyeimbang beban? 728
Bagaimana cara memperbarui layanan Amazon ECS saya dengan informasi baru selama penerapan? 728

Pemecahan Masalah AWS Masalah penyebaran Lambda 729
AWS Lambda penerapan gagal setelah menghentikan penerapan Lambda secara manual yang tidak memiliki rollback yang dikonfigurasi 729
Memecahkan masalah grup penyebaran 730
Menandai instance sebagai bagian dari grup penerapan tidak secara otomatis menyebarkan aplikasi Anda ke instance baru 730
Memecahkan masalah contoh 730
Tag harus diatur dengan benar 730
AWS CodeDeploy agen harus diinstal dan berjalan pada instance 731
Penerapan tidak gagal hingga satu jam saat instance dihentikan selama penerapan 731
Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance 731
Buat file CodeDeploy log baru jika tidak sengaja dihapus 732
Pemecahan masalah "InvalidSignatureException - Tanda tangan kedaluwarsa: [waktu] sekarang lebih awal dari [waktu]” kesalahan penerapan 732
Memecahkan masalah GitHub token 733
Token OAuth tidak valid GitHub 733
Jumlah maksimum token GitHub OAuth terlampaui 733
Memecahkan masalah Amazon EC2 Auto Scaling 734
Pemecahan masalah Penskalaan Otomatis Amazon EC2 Umum 734
“CodeDeployRole tidak memberi Anda izin untuk melakukan operasi sebagai berikut AWS layanan: AmazonAutoScaling "kesalahan 735
Instans dalam grup Amazon EC2 Auto Scaling terus disediakan dan dihentikan sebelum revisi dapat diterapkan 736
Mengakhiri atau me-reboot instans Amazon EC2 Auto Scaling dapat menyebabkan penerapan gagal 737
Hindari mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2 737
Instans EC2 dalam grup Amazon EC2 Auto Scaling gagal diluncurkan dan menerima kesalahan “Heartbeat Timeout” 738
Kait siklus hidup Amazon EC2 Auto Scaling yang tidak cocok dapat menyebabkan penerapan otomatis ke grup Amazon EC2 Auto Scaling berhenti atau gagal 741
Kesalahan “Penerapan gagal karena tidak ada instance yang ditemukan untuk grup penerapan Anda” 742
Kode eror 750
Topik terkait 756
Sumber daya 757

Panduan referensi dan sumber daya pendukung 757
Sampel 757
Blog 757
AWS kit dan alat pengembangan perangkat lunak 757
Riwayat dokumen 759
Pembaruan lebih awal 779
AWS Glosarium 801
..... dcccii

Apa itu CodeDeploy?

CodeDeploy adalah layanan penerapan yang mengotomatiskan penerapan aplikasi ke instans Amazon EC2, instans lokal, fungsi Lambda tanpa server, atau layanan Amazon ECS.

Anda dapat menyebarkan berbagai konten aplikasi yang hampir tidak terbatas, termasuk:

- Kode
- Fungsi tanpa server AWS Lambda
- File web dan konfigurasi
- Dapat dieksekusi
- Paket
- Skrip
- File multimedia

CodeDeploy dapat menyebarkan konten aplikasi yang berjalan di server dan disimpan di bucket Amazon S3, repositori GitHub , atau repositori Bitbucket. CodeDeploy juga dapat menerapkan fungsi Lambda tanpa server. Anda tidak perlu membuat perubahan pada kode yang ada sebelum Anda dapat menggunakannya CodeDeploy.

CodeDeploy membuatnya lebih mudah bagi Anda untuk:

- Cepat merilis fitur baru.
- Perbarui versi AWS Lambda fungsi.
- Hindari waktu henti selama penerapan aplikasi.
- Tangani kompleksitas memperbarui aplikasi Anda, tanpa banyak risiko yang terkait dengan penerapan manual yang rawan kesalahan.

Skala layanan dengan infrastruktur Anda sehingga Anda dapat dengan mudah menyebarkan ke satu instans atau ribuan.

CodeDeploy bekerja dengan berbagai sistem untuk manajemen konfigurasi, kontrol sumber, integrasi berkelanjutan, pengiriman berkelanjutan, dan penyebaran berkelanjutan. Untuk informasi selengkapnya, lihat Integrasi produk.

CodeDeploy Konsol juga menyediakan cara untuk mencari sumber daya Anda dengan cepat, seperti repositori, membangun proyek, aplikasi penerapan, dan saluran pipa. Pilih Pergi ke sumber daya atau tekan tombol /, dan kemudian ketik nama sumber daya. Setiap kecocokan akan muncul di daftar. Pencarian peka huruf besar/kecil. Anda hanya melihat sumber daya yang izin untuk menampilkannya Anda memiliki. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk AWS CodeDeploy.

Topik

- Manfaat dari AWS CodeDeploy
- Ikhtisar CodeDeploy platform komputasi
- Ikhtisar jenis CodeDeploy penerapan
- Kami ingin mendengar pendapat Anda
- CodeDeploy komponen utama
- CodeDeploy penyebaran
- CodeDeploy spesifikasi aplikasi (AppSpec) file

Manfaat dari AWS CodeDeploy

CodeDeploy menawarkan manfaat ini:

- Aplikasi server, tanpa server, dan kontainer. CodeDeploymemungkinkan Anda menyebarkan aplikasi tradisional pada server dan aplikasi yang menyebarkan versi AWS Lambda fungsi tanpa server atau aplikasi Amazon ECS.
- Penerapan otomatis. CodeDeploy sepenuhnya mengotomatiskan penerapan aplikasi Anda di seluruh lingkungan pengembangan, pengujian, dan produksi Anda. CodeDeployskala dengan infrastruktur Anda sehingga Anda dapat menyebarkan ke satu instance atau ribuan.
- Minimalkan waktu henti. Jika aplikasi Anda menggunakan platform EC2/On-Premises komputasi, CodeDeploy membantu memaksimalkan ketersediaan aplikasi Anda. Selama penerapan di tempat, CodeDeploy melakukan pembaruan bergulir di seluruh instans Amazon EC2. Anda dapat menentukan jumlah instance yang akan diambil offline pada satu waktu untuk pembaruan. Selama blue/green penerapan, revisi aplikasi terbaru diinstal pada instance pengganti. Lalu lintas dialihkan ke contoh ini ketika Anda memilih, baik segera atau segera setelah Anda selesai menguji lingkungan baru. Untuk kedua jenis penerapan, CodeDeploy melacak kesehatan aplikasi sesuai dengan aturan yang Anda konfigurasikan.

- Berhenti dan putar kembali. Anda dapat secara otomatis atau manual menghentikan dan memutar kembali penerapan jika ada kesalahan.
- Kontrol terpusat. Anda dapat meluncurkan dan melacak status penerapan Anda melalui CodeDeploy konsol atau. AWS CLI Anda menerima laporan yang mencantumkan kapan setiap revisi aplikasi diterapkan dan instans Amazon EC2 mana.
- Mudah diadopsi. CodeDeploy adalah platform-agnostik dan bekerja dengan aplikasi apa pun. Anda dapat dengan mudah menggunakan kembali kode pengaturan Anda. CodeDeploy juga dapat berintegrasi dengan proses rilis perangkat lunak Anda atau rantai alat pengiriman berkelanjutan.
- Penerapan bersamaan. Jika Anda memiliki lebih dari satu aplikasi yang menggunakan platform EC2/On-Premises komputasi, CodeDeploy dapat menerapkannya secara bersamaan ke kumpulan instance yang sama.

Ikhtisar CodeDeploy platform komputasi

CodeDeploy mampu menyebarkan aplikasi ke tiga platform komputasi:

- EC2/On-Premises: Menjelaskan instance server fisik yang dapat berupa instans cloud Amazon EC2, server lokal, atau keduanya. Aplikasi yang dibuat menggunakan platform EC2/On-Premises komputasi dapat terdiri dari file yang dapat dieksekusi, file konfigurasi, gambar, dan banyak lagi.

  Penerapan yang menggunakan platform EC2/On-Premises komputasi mengelola cara lalu lintas diarahkan ke instance dengan menggunakan tipe in-place atau deployment. blue/green Untuk informasi selengkapnya, lihat Ikhtisar jenis CodeDeploy penerapan.
- AWS Lambda: Digunakan untuk menyebarkan aplikasi yang terdiri dari versi terbaru dari fungsi Lambda. AWS Lambda mengelola fungsi Lambda dalam lingkungan komputasi tanpa server yang terdiri dari struktur komputasi ketersediaan tinggi. Semua administrasi sumber daya komputasi dilakukan oleh AWS Lambda. Untuk informasi selengkapnya, lihat Komputasi dan Aplikasi Tanpa Server. Untuk informasi selengkapnya tentang AWS Lambda dan fungsi Lambda, lihat. AWS Lambda

  Anda dapat mengelola cara lalu lintas dialihkan ke versi fungsi Lambda yang diperbarui selama penerapan dengan memilih konfigurasi canary, linear, atau all-at-once.
- Amazon ECS: Digunakan untuk menyebarkan aplikasi kontainer Amazon ECS sebagai set tugas. CodeDeploy melakukan blue/green penyebaran dengan menginstal versi aplikasi yang diperbarui sebagai set tugas pengganti baru. CodeDeploy mengubah rute lalu lintas produksi dari tugas

aplikasi asli yang disetel ke set tugas pengganti. Set tugas asli dihentikan setelah penerapan berhasil. Untuk informasi selengkapnya tentang Amazon ECS, lihat Amazon Elastic Container Service.

Anda dapat mengelola cara lalu lintas dialihkan ke set tugas yang diperbarui selama penerapan dengan memilih konfigurasi canary, linier, atau all-at-once.

> **Note**
>
> blue/green Penerapan Amazon ECS didukung menggunakan keduanya dan. CodeDeploy CloudFormation Detail untuk penerapan ini dijelaskan di bagian berikutnya.

Tabel berikut menjelaskan bagaimana CodeDeploy komponen digunakan dengan setiap platform komputasi. Untuk informasi lebih lanjut, lihat:

- Bekerja dengan grup penyebaran di CodeDeploy
- Bekerja dengan penerapan di CodeDeploy
- Bekerja dengan konfigurasi penerapan di CodeDeploy
- Bekerja dengan revisi aplikasi untuk CodeDeploy
- Bekerja dengan aplikasi di CodeDeploy

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
Grup deployment	Menyebarkan revisi ke serangkaian instance.	Menerapkan versi baru fungsi Lambda tanpa server pada infrastruktur komputasi ketersediaan tinggi.	Menentukan layanan Amazon ECS dengan aplikasi kontainer yang akan diterapkan sebagai set tugas,

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
			pendengar pengujian produksi dan opsional yang digunakan untuk menyajikan lalu lintas ke aplikasi yang diterapkan, kapan mengalihkan lalu lintas dan menghentikan set tugas asli aplikasi yang diterapkan, serta pengaturan pemicu, alarm, dan rollback opsional.

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
Deployment	Menyebarkan revisi baru yang terdiri dari aplikasi dan AppSpec file. AppSpec Menentukan cara menyebarkan aplikasi ke instance dalam grup penyebaran.	Menggeser lalu lintas produksi dari satu versi fungsi Lambda ke versi baru dari fungsi yang sama. AppSpec File menentukan versi fungsi Lambda mana yang akan digunakan.	Menerapkan versi terbaru dari aplikasi kontainer Amazon ECS sebagai set tugas pengganti baru. CodeDeploy mengubah rute lalu lintas produksi dari set tugas dengan versi asli ke set tugas pengganti baru dengan versi yang diperbarui. Ketika penerapan selesai, set tugas asli dihentikan.

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
Konfigurasi deployment	Pengaturan yang menentukan kecepatan penerapan dan jumlah minimum instance yang harus sehat kapan saja selama penerapan.	Pengaturan yang menentukan bagaimana lalu lintas dialihkan ke versi fungsi Lambda yang diperbarui.	Pengaturan yang menentukan bagaimana lalu lintas dialihkan ke set tugas Amazon ECS yang diperbarui.

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
Revisi	Kombinasi AppSpec file dan file aplikasi, seperti executable, file konfigurasi, dan sebagainya.	AppSpec File yang menentukan fungsi Lambda mana yang akan digunakan dan fungsi Lambda yang dapat menjalankan pengujian validasi selama kait peristiwa siklus hidup penerapan.	AppSpec File yang menentukan: • Definisi tugas Amazon ECS untuk layanan Amazon ECS dengan aplikasi kontainer untuk diterapkan. • Kontainer tempat aplikasi Anda yang diperbarui digunakan. • Port untuk kontainer tempat lalu

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
			lintas produksi dialihkan. • Pengaturan konfigurasi jaringan opsional dan fungsi Lambda yang dapat menjalankan pengujian validasi selama kait peristiwa siklus hidup penerapan.

CodeDeploy komponen	EC2/On-Premises	AWS Lambda	Amazon ECS
Aplikasi	Kumpulan grup penyebaran dan revisi. EC2/On-PremisesAplikasi menggunakan platform EC2/On-Premises komputasi.	Kumpulan grup penyebaran dan revisi. Aplikasi yang digunakan untuk AWS Lambda penyebaran menggunakan platform komputasi AWS Lambda tanpa server.	Kumpulan grup penyebaran dan revisi. Aplikasi yang digunakan untuk penyebaran Amazon ECS menggunakan platform komputasi Amazon ECS.

Ikhtisar jenis CodeDeploy penerapan

CodeDeploy menyediakan dua opsi jenis penerapan:

- In-place deployment: Aplikasi pada setiap instance dalam grup penyebaran dihentikan, revisi aplikasi terbaru diinstal, dan versi baru aplikasi dimulai dan divalidasi. Anda dapat menggunakan penyeimbang beban sehingga setiap instance dideregistrasi selama penerapannya dan kemudian dikembalikan ke layanan setelah penerapan selesai. Hanya penerapan yang menggunakan platform EC2/On-Premises komputasi yang dapat menggunakan penerapan di tempat. Untuk informasi selengkapnya tentang penerapan di tempat, lihat. Ikhtisar penerapan di tempat

> **Note**
>
> AWS Penerapan Lambda dan Amazon ECS tidak dapat menggunakan jenis penerapan di tempat.

- Blue/green deployment: Perilaku penerapan Anda bergantung pada platform komputasi yang Anda gunakan:
  - Blue/green pada platform EC2/On-Premises komputasi: Instance dalam grup penerapan (lingkungan asli) digantikan oleh serangkaian instance yang berbeda (lingkungan pengganti) menggunakan langkah-langkah ini:
    - Instans disediakan untuk lingkungan pengganti.
    - Revisi aplikasi terbaru diinstal pada instance pengganti.
    - Waktu tunggu opsional terjadi untuk kegiatan seperti pengujian aplikasi dan verifikasi sistem.
    - Instans di lingkungan penggantian terdaftar dengan satu atau lebih penyeimbang beban Elastic Load Balancing, menyebabkan lalu lintas dialihkan ke sana. Contoh di lingkungan asli dideregistrasi dan dapat dihentikan atau terus berjalan untuk penggunaan lain.

    > **Note**
    >
    > Jika Anda menggunakan platform EC2/On-Premises komputasi, ketahuilah bahwa blue/green penerapan hanya berfungsi dengan instans Amazon EC2.

  - Blue/green pada platform komputasi Amazon ECS AWS Lambda atau Amazon: Lalu lintas digeser secara bertahap sesuai dengan konfigurasi penerapan kenari, linier, atau all-at-once.
  - Blue/green penerapan melalui CloudFormation: Lalu lintas dialihkan dari sumber daya Anda saat ini ke sumber daya yang diperbarui sebagai bagian dari pembaruan CloudFormation tumpukan. Saat ini, hanya blue/green penerapan ECS yang didukung.

Untuk informasi selengkapnya tentang blue/green penerapan, lihat. Ikhtisar blue/green penerapan

> **Note**
>
> Dengan menggunakan CodeDeploy agen, Anda dapat melakukan penyebaran pada instans yang Anda masuki tanpa memerlukan aplikasi, grup penyebaran, atau bahkan akun AWS . Untuk informasi, lihat Gunakan CodeDeploy agen untuk memvalidasi paket penerapan pada mesin lokal.

Topik

- Ikhtisar penerapan di tempat

- Ikhtisar blue/green penerapan

Ikhtisar penerapan di tempat

> **Note**
>
> AWS Penerapan Lambda dan Amazon ECS tidak dapat menggunakan jenis penerapan di tempat.

Inilah cara kerja penerapan di tempat:

1. Pertama, Anda membuat konten deployable pada mesin pengembangan lokal Anda atau lingkungan serupa, dan kemudian Anda menambahkan file spesifikasi aplikasi (AppSpec file). AppSpec File ini unik untuk CodeDeploy. Ini mendefinisikan tindakan penerapan yang CodeDeploy ingin Anda jalankan. Anda menggabungkan konten yang dapat di-deploy dan AppSpec file ke dalam file arsip, lalu mengunggahnya ke bucket Amazon S3 atau repositori. GitHub File arsip ini disebut revisi aplikasi (atau hanya revisi).
2. Selanjutnya, Anda memberikan CodeDeploy informasi tentang penerapan Anda, seperti bucket GitHub atau repositori Amazon S3 untuk menarik revisi dari dan ke set instans Amazon EC2 mana yang akan menyebarkan kontennya. CodeDeploy memanggil satu set instans Amazon EC2 grup penerapan. Grup penerapan berisi instans Amazon EC2 yang ditandai secara individual, instans Amazon EC2 di grup Penskalaan Otomatis Amazon EC2, atau keduanya.

   Setiap kali Anda berhasil mengunggah revisi aplikasi baru yang ingin Anda terapkan ke grup penerapan, bundel tersebut ditetapkan sebagai revisi target untuk grup penerapan. Dengan kata lain, revisi aplikasi yang saat ini ditargetkan untuk penerapan adalah revisi target. Ini juga merupakan revisi yang ditarik untuk penerapan otomatis.
3. Selanjutnya, CodeDeploy agen pada setiap instans melakukan polling CodeDeploy untuk menentukan apa dan kapan harus menarik dari bucket GitHub atau repositori Amazon S3 yang ditentukan.
4. Terakhir, CodeDeploy agen pada setiap instance menarik revisi target dari bucket GitHub atau repositori Amazon S3 dan, menggunakan instruksi dalam file, menyebarkan AppSpec konten ke instance.

CodeDeploy menyimpan catatan penerapan Anda sehingga Anda bisa mendapatkan status penerapan, parameter konfigurasi penerapan, kesehatan instance, dan sebagainya.

Ikhtisar blue/green penerapan

blue/green Penerapan digunakan untuk memperbarui aplikasi Anda sambil meminimalkan gangguan yang disebabkan oleh perubahan versi aplikasi baru. CodeDeploy menyediakan versi aplikasi baru Anda di samping versi lama sebelum mengalihkan lalu lintas produksi Anda.

- AWS Lambda: Lalu lintas digeser dari satu versi fungsi Lambda ke versi baru dari fungsi Lambda yang sama.
- Amazon ECS: Lalu lintas dialihkan dari tugas yang ditetapkan di layanan Amazon ECS Anda ke tugas pengganti yang diperbarui yang disetel dalam layanan Amazon ECS yang sama.
- EC2/On-Premises: Lalu lintas digeser dari satu set instance di lingkungan asli ke serangkaian instance pengganti.

Semua penerapan AWS Lambda dan Amazon ECS ada. blue/green EC2/On-Premises Penyebaran bisa di tempat atau. blue/green blue/green Penerapan menawarkan sejumlah keunggulan dibandingkan penerapan di tempat:

- Anda dapat menginstal dan menguji aplikasi di lingkungan pengganti baru dan menyebarkannya ke produksi hanya dengan mengalihkan lalu lintas.
- Jika Anda menggunakan platform EC2/On-Premises komputasi, beralih kembali ke versi terbaru aplikasi lebih cepat dan lebih dapat diandalkan. Itu karena lalu lintas dapat dialihkan kembali ke instance asli selama mereka belum dihentikan. Dengan penerapan di tempat, versi harus diputar kembali dengan menerapkan kembali versi aplikasi sebelumnya.
- Jika Anda menggunakan platform EC2/On-Premises komputasi, instance baru disediakan untuk blue/green penerapan dan mencerminkan konfigurasi server terbaru. Ini membantu Anda menghindari jenis masalah yang terkadang terjadi pada kasus yang berjalan lama.
- Jika Anda menggunakan platform komputasi AWS Lambda, Anda mengontrol bagaimana lalu lintas digeser dari versi fungsi Lambda asli Anda ke versi fungsi AWS Lambda baru Anda. AWS
- Jika Anda menggunakan platform komputasi Amazon ECS, Anda mengontrol bagaimana lalu lintas digeser dari tugas awal yang disetel ke set tugas baru Anda.

blue/green Penerapan dengan CloudFormation dapat menggunakan salah satu metode berikut:

- CloudFormation template untuk penerapan: Saat Anda mengonfigurasi penerapan dengan CloudFormation templat, penerapan Anda dipicu oleh pembaruan. CloudFormation Saat Anda mengubah sumber daya dan mengunggah perubahan templat, pembaruan tumpukan akan CloudFormation memulai penerapan baru. Untuk daftar sumber daya yang dapat Anda gunakan dalam CloudFormation templat, lihatCloudFormation template untuk CodeDeploy referensi.
- Blue/green penerapan melalui CloudFormation: Anda dapat menggunakan CloudFormation untuk mengelola blue/green penerapan Anda melalui pembaruan tumpukan. Anda menentukan sumber daya biru dan hijau Anda, selain menentukan pengaturan perutean dan stabilisasi lalu lintas, dalam template tumpukan. Kemudian, jika Anda memperbarui sumber daya yang dipilih selama pembaruan tumpukan, CloudFormation menghasilkan semua sumber daya hijau yang diperlukan, menggeser lalu lintas berdasarkan parameter perutean lalu lintas yang ditentukan, dan menghapus sumber daya biru. Untuk informasi selengkapnya, lihat Mengotomatiskan blue/green penerapan Amazon ECS melalui CodeDeploy penggunaan CloudFormation di Panduan Pengguna.AWS CloudFormation

> **Note**
>
> Didukung hanya untuk blue/green penerapan Amazon ECS.

Cara Anda mengonfigurasi blue/green penerapan tergantung pada platform komputasi mana yang digunakan penerapan Anda.

Blue/Green penyebaran pada AWS Lambda atau platform komputasi Amazon ECS

Jika Anda menggunakan platform komputasi Amazon ECS AWS Lambda atau Amazon, Anda harus menunjukkan bagaimana lalu lintas digeser dari fungsi asli AWS Lambda atau tugas Amazon ECS yang disetel ke fungsi atau set tugas baru. Untuk menunjukkan bagaimana lalu lintas digeser, Anda harus menentukan salah satu konfigurasi penerapan berikut:

- kenari
- linier
- semua-sekaligus

Untuk informasi tentang bagaimana lalu lintas digeser dalam konfigurasi penerapan canary, linear, atau all-at-once, lihat. Konfigurasi deployment

Untuk detail tentang konfigurasi penerapan Lambda, lihat. Konfigurasi penerapan pada sebuah AWS Lambda platform komputasi

Untuk detail tentang konfigurasi penerapan Amazon ECS, lihat. Konfigurasi deployment pada platform komputasi Amazon ECS

Blue/Green penyebaran pada platform EC2/on-premises komputasi

> **Note**
>
> Anda harus menggunakan instans Amazon EC2 untuk blue/green penerapan di platform komputasi. EC2/On-Premises On-premises instance tidak didukung untuk jenis blue/green penerapan.

Jika Anda menggunakan platform EC2/On-Premises komputasi, berikut ini berlaku:

Anda harus memiliki satu atau beberapa instans Amazon EC2 dengan mengidentifikasi tag Amazon EC2 atau grup Amazon EC2 Auto Scaling. Contoh harus memenuhi persyaratan tambahan ini:

- Setiap instans Amazon EC2 harus memiliki profil instans IAM yang benar.
- CodeDeploy Agen harus diinstal dan dijalankan pada setiap instance.

> **Note**
>
> Anda biasanya juga memiliki revisi aplikasi yang berjalan pada instance di lingkungan asli Anda, tetapi ini bukan persyaratan untuk penerapan. blue/green

Saat membuat grup penyebaran yang digunakan dalam blue/green penerapan, Anda dapat memilih bagaimana lingkungan pengganti Anda ditentukan:

Salin grup Penskalaan Otomatis Amazon EC2 yang ada: blue/green Selama CodeDeploy penerapan, buat instans untuk lingkungan pengganti Anda selama penerapan. Dengan opsi ini, CodeDeploy gunakan grup Amazon EC2 Auto Scaling yang Anda tentukan sebagai templat untuk lingkungan pengganti, termasuk jumlah instans yang sedang berjalan dan banyak opsi konfigurasi lainnya.

Pilih instans secara manual: Anda dapat menentukan instans yang akan dihitung sebagai pengganti menggunakan tag instans Amazon EC2, nama grup Amazon EC2 Auto Scaling, atau keduanya. Jika

Anda memilih opsi ini, Anda tidak perlu menentukan instance untuk lingkungan pengganti sampai Anda membuat penerapan.

Begini cara kerjanya:

1. Anda sudah memiliki instans atau grup Amazon EC2 Auto Scaling yang berfungsi sebagai lingkungan asli Anda. Saat pertama kali menjalankan blue/green penerapan, Anda biasanya menggunakan instance yang sudah digunakan dalam penerapan di tempat.
2. Di CodeDeploy aplikasi yang ada, Anda membuat grup blue/green penyebaran di mana, selain opsi yang diperlukan untuk penerapan di tempat, Anda menentukan hal berikut:
   - Penyeimbang beban atau penyeimbang beban yang merutekan lalu lintas dari lingkungan asli Anda ke lingkungan pengganti Anda selama proses penyebaran. blue/green
   - Apakah akan mengalihkan lalu lintas ke lingkungan pengganti segera atau menunggu Anda untuk mengubah rute secara manual.
   - Tingkat di mana lalu lintas dialihkan ke instance penggantian.
   - Apakah instance yang diganti dihentikan atau terus berjalan.
3. Anda membuat penerapan untuk grup penerapan ini selama hal berikut terjadi:
   a. Jika Anda memilih untuk menyalin grup Amazon EC2 Auto Scaling, instans akan disediakan untuk lingkungan pengganti Anda.
   b. Revisi aplikasi yang Anda tentukan untuk penerapan diinstal pada instance pengganti.
   c. Jika Anda menentukan waktu tunggu dalam pengaturan grup penerapan, penerapan akan dijeda. Ini adalah waktu ketika Anda dapat menjalankan tes dan verifikasi di lingkungan pengganti Anda. Jika Anda tidak mengubah rute lalu lintas secara manual sebelum akhir periode tunggu, penerapan dihentikan.
   d. Instans di lingkungan penggantian terdaftar dengan penyeimbang beban Elastic Load Balancing dan lalu lintas mulai diarahkan ke sana.
   e. Instance di lingkungan asli dideregistrasi dan ditangani sesuai dengan spesifikasi Anda di grup penerapan, baik dihentikan atau terus berjalan.

Blue/Green penyebaran melalui CloudFormation

Anda dapat mengelola CodeDeploy blue/green penerapan dengan memodelkan sumber daya Anda dengan templat CloudFormation .

Saat Anda memodelkan blue/green sumber daya menggunakan CloudFormation templat, Anda membuat pembaruan tumpukan CloudFormation yang memperbarui set tugas Anda. Lalu lintas

produksi bergeser dari tugas asli layanan Anda yang disetel ke set tugas pengganti sekaligus, dengan penerapan linier dan waktu pemanggangan, atau dengan penerapan kenari. Pembaruan tumpukan memulai penerapan di. CodeDeploy Anda dapat melihat status dan riwayat penerapan di CodeDeploy, tetapi Anda tidak membuat atau mengelola CodeDeploy sumber daya di luar CloudFormation templat.

> **Note**
>
> Untuk blue/green penerapan melalui CloudFormation, Anda tidak membuat CodeDeploy aplikasi atau grup penyebaran.

Metode ini hanya mendukung blue/green penerapan Amazon ECS. Untuk informasi selengkapnya tentang blue/green penerapan CloudFormation, lihat. Buat blue/green penyebaran Amazon ECS melalui CloudFormation

Kami ingin mendengar pendapat Anda

Kami menyambut umpan balik Anda. Untuk menghubungi kami, kunjungi CodeDeploy forum.

Topik

- Primary Components
- Deployments
- Application Specification Files

CodeDeploy komponen utama

Sebelum Anda mulai bekerja dengan layanan ini, Anda harus membiasakan diri dengan komponen utama dari proses CodeDeploy penyebaran.

Topik

- Aplikasi
- Platform komputasi
- Konfigurasi deployment
- Grup deployment
- Tipe deployment

- Profil instans IAM
- Revisi
- Peran layanan
- Revisi target
- Komponen lainnya

Aplikasi

Aplikasi adalah nama yang secara unik mengidentifikasi aplikasi yang ingin Anda terapkan. CodeDeploy menggunakan nama ini, yang berfungsi sebagai wadah, untuk memastikan kombinasi yang benar dari revisi, konfigurasi penerapan, dan grup penyebaran direferensikan selama penerapan.

Platform komputasi

Platform komputasi adalah platform tempat CodeDeploy menyebarkan aplikasi. Ada tiga platform komputasi:

- EC2/On-Premises: Menjelaskan instance server fisik yang dapat berupa instans cloud Amazon EC2, server lokal, atau keduanya. Aplikasi yang dibuat menggunakan platform EC2/On-Premises komputasi dapat terdiri dari file yang dapat dieksekusi, file konfigurasi, gambar, dan banyak lagi.

  Penerapan yang menggunakan platform EC2/On-Premises komputasi mengelola cara lalu lintas diarahkan ke instance dengan menggunakan tipe in-place atau deployment. blue/green Untuk informasi selengkapnya, lihat Ikhtisar jenis CodeDeploy penerapan.
- AWS Lambda: Digunakan untuk menyebarkan aplikasi yang terdiri dari versi terbaru dari fungsi Lambda. AWS Lambda mengelola fungsi Lambda dalam lingkungan komputasi tanpa server yang terdiri dari struktur komputasi ketersediaan tinggi. Semua administrasi sumber daya komputasi dilakukan oleh AWS Lambda. Untuk informasi selengkapnya, lihat Komputasi dan Aplikasi Tanpa Server. Untuk informasi selengkapnya tentang AWS Lambda dan fungsi Lambda, lihat. AWS Lambda

  Anda dapat mengelola cara lalu lintas dialihkan ke versi fungsi Lambda yang diperbarui selama penerapan dengan memilih konfigurasi canary, linear, atau all-at-once.
- Amazon ECS: Digunakan untuk menyebarkan aplikasi kontainer Amazon ECS sebagai set tugas. CodeDeploy melakukan blue/green penyebaran dengan menginstal versi aplikasi yang diperbarui sebagai set tugas pengganti baru. CodeDeploy mengubah rute lalu lintas produksi dari tugas

aplikasi asli yang disetel ke set tugas pengganti. Set tugas asli dihentikan setelah penerapan berhasil. Untuk informasi selengkapnya tentang Amazon ECS, lihat Amazon Elastic Container Service.

Anda dapat mengelola cara lalu lintas dialihkan ke set tugas yang diperbarui selama penerapan dengan memilih konfigurasi canary, linier, atau all-at-once.

> **Note**
>
> blue/green Penerapan Amazon ECS didukung melalui keduanya dan. CodeDeploy CloudFormation Detail untuk penerapan ini dijelaskan di bagian berikutnya.

Konfigurasi deployment

Konfigurasi penerapan adalah seperangkat aturan penerapan dan kondisi keberhasilan dan kegagalan penerapan yang digunakan CodeDeploy selama penerapan. Jika penerapan Anda menggunakan platform EC2/On-Premises komputasi, Anda dapat menentukan jumlah minimum instans sehat untuk penerapan. Jika penerapan Anda menggunakan AWS Lambda atau platform komputasi Amazon ECS, Anda dapat menentukan cara lalu lintas dirutekan ke fungsi Lambda atau set tugas ECS yang diperbarui.

Untuk informasi selengkapnya tentang menentukan jumlah minimum host sehat untuk penerapan yang menggunakan platform EC2/On-Premises komputasi, lihat. Tentang jumlah minimum contoh sehat

Konfigurasi penerapan berikut menentukan cara lalu lintas dirutekan selama penerapan yang menggunakan Lambda atau platform komputasi ECS:

- Canary: Lalu lintas dialihkan dalam dua peningkatan. Anda dapat memilih dari opsi kenari yang telah ditentukan sebelumnya yang menentukan persentase lalu lintas yang digeser ke fungsi Lambda Anda yang diperbarui atau tugas ECS yang ditetapkan dalam kenaikan pertama dan interval, dalam hitungan menit, sebelum lalu lintas yang tersisa digeser dalam kenaikan kedua.
- Linier: Lalu lintas dialihkan dalam peningkatan yang sama dengan jumlah menit yang sama di antara setiap kenaikan. Anda dapat memilih dari opsi linier yang telah ditentukan sebelumnya yang menentukan persentase lalu lintas yang bergeser dalam setiap kenaikan dan jumlah menit di antara setiap kenaikan.

- All-at-once: Semua lalu lintas digeser dari fungsi Lambda asli atau tugas ECS diatur ke fungsi yang diperbarui atau set tugas sekaligus.

Grup deployment

Grup penyebaran adalah sekumpulan instance individual. Grup penerapan berisi instans yang ditandai secara individual, instans Amazon EC2 di grup Penskalaan Otomatis Amazon EC2, atau keduanya. Untuk informasi tentang tag instans Amazon EC2, lihat Bekerja dengan Tag Menggunakan Konsol. Untuk informasi tentang instance lokal, lihat. Working with On-Premises Instances Untuk informasi tentang Amazon EC2 Auto Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling Scaling, lihat.

Tipe deployment

Jenis penyebaran adalah metode yang digunakan untuk membuat revisi aplikasi terbaru tersedia pada instance dalam grup penyebaran. Ada dua jenis penyebaran:

- In-place deployment: Aplikasi pada setiap instance dalam grup penyebaran dihentikan, revisi aplikasi terbaru diinstal, dan versi baru aplikasi dimulai dan divalidasi. Anda dapat menggunakan penyeimbang beban sehingga setiap instance dideregistrasi selama penerapannya dan kemudian dikembalikan ke layanan setelah penerapan selesai. Hanya penerapan yang menggunakan platform EC2/On-Premises komputasi yang dapat menggunakan penerapan di tempat. Untuk informasi selengkapnya tentang penerapan di tempat, lihat. Ikhtisar penerapan di tempat
- Blue/green deployment: Perilaku penerapan Anda bergantung pada platform komputasi yang Anda gunakan:
  - Blue/green pada platform EC2/On-Premises komputasi: Instance dalam grup penerapan (lingkungan asli) digantikan oleh serangkaian instance yang berbeda (lingkungan pengganti) menggunakan langkah-langkah ini:
    - Instans disediakan untuk lingkungan pengganti.
    - Revisi aplikasi terbaru diinstal pada instance pengganti.
    - Waktu tunggu opsional terjadi untuk kegiatan seperti pengujian aplikasi dan verifikasi sistem.
    - Instans di lingkungan penggantian terdaftar dengan satu atau lebih penyeimbang beban Elastic Load Balancing, menyebabkan lalu lintas dialihkan ke sana. Contoh di lingkungan asli dideregistrasi dan dapat dihentikan atau terus berjalan untuk penggunaan lain.

> **Note**
>
> Jika Anda menggunakan platform EC2/On-Premises komputasi, ketahuilah bahwa blue/green penerapan hanya berfungsi dengan instans Amazon EC2.

- Blue/green pada platform komputasi Amazon ECS AWS Lambda atau Amazon: Lalu lintas digeser secara bertahap sesuai dengan konfigurasi penerapan kenari, linier, atau all-at-once.
- Blue/green penerapan melalui CloudFormation: Lalu lintas dialihkan dari sumber daya Anda saat ini ke sumber daya yang diperbarui sebagai bagian dari pembaruan CloudFormation tumpukan. Saat ini, hanya blue/green penerapan ECS yang didukung.

Untuk informasi selengkapnya tentang blue/green penerapan, lihat. Ikhtisar blue/green penerapan

> **Note**
>
> blue/green Penerapan Amazon ECS didukung menggunakan keduanya dan. CodeDeploy CloudFormation Detail untuk penerapan ini dijelaskan di bagian berikutnya.

Profil instans IAM

Profil instans IAM adalah peran IAM yang Anda lampirkan ke instans Amazon EC2 Anda. Profil ini mencakup izin yang diperlukan untuk mengakses bucket Amazon S3 GitHub atau repositori tempat aplikasi disimpan. Untuk informasi selengkapnya, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.

Revisi

Revisi adalah versi aplikasi Anda. Revisi penyebaran AWS Lambda adalah YAML- atau JSON-formatted file yang menentukan informasi tentang fungsi Lambda yang akan digunakan. Revisi EC2/On-Premises penyebaran adalah file arsip yang berisi konten sumber (kode sumber, halaman web, file yang dapat dieksekusi, dan skrip penyebaran) dan file spesifikasi aplikasi (file). AppSpec AWS Revisi Lambda dapat disimpan di ember Amazon S3. EC2/On-Premises revisi disimpan dalam ember atau repositori Amazon S3. GitHub Untuk Amazon S3, revisi secara unik diidentifikasi oleh kunci objek Amazon S3 dan ETag-nya, versi, atau keduanya. Untuk GitHub, revisi diidentifikasi secara unik oleh ID komitnya.

Peran layanan

Peran layanan adalah peran IAM yang memberikan izin ke AWS layanan sehingga dapat mengakses sumber daya. AWS Kebijakan yang Anda lampirkan ke peran layanan menentukan AWS sumber daya yang dapat diakses oleh layanan dan tindakan yang dapat dilakukan dengan sumber daya tersebut. Untuk CodeDeploy, peran layanan digunakan untuk hal berikut:

- Untuk membaca tag yang diterapkan pada instans atau nama grup Amazon EC2 Auto Scaling yang terkait dengan instans. Hal ini memungkinkan CodeDeploy untuk mengidentifikasi instance yang dapat digunakan aplikasi.
- Untuk melakukan operasi pada instans, Amazon EC2 Auto Scaling mengelompokkan, dan penyeimbang beban Elastic Load Balancing.
- Untuk mempublikasikan informasi ke topik Amazon SNS sehingga notifikasi dapat dikirim saat penerapan atau kejadian instans tertentu terjadi.
- Untuk mengambil informasi tentang CloudWatch alarm untuk mengatur pemantauan alarm untuk penyebaran.

Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Revisi target

Revisi target adalah versi terbaru dari revisi aplikasi yang telah Anda unggah ke repositori Anda dan ingin menyebarkan ke instance dalam grup penyebaran. Dengan kata lain, revisi aplikasi saat ini ditargetkan untuk deployment. Ini juga merupakan revisi yang ditarik untuk penerapan otomatis.

Komponen lainnya

Untuk informasi tentang komponen lain dalam CodeDeploy alur kerja, lihat topik berikut:

- Pilih jenis CodeDeploy repositori
- Deployments
- Application Specification Files
- Instance Health
- Bekerja dengan CodeDeploy agen
- Working with On-Premises Instances

CodeDeploy penyebaran

Topik ini memberikan informasi tentang komponen dan alur kerja penerapan di. CodeDeploy Proses penerapan bervariasi, tergantung pada platform komputasi atau metode penerapan (Lambda, Amazon ECS EC2/On-Premises, atau melalui AWS CloudFormation) yang Anda gunakan untuk penerapan.

Topik

- Penerapan pada AWS Platform Komputasi Lambda
- Deployment di Platform Komputasi Amazon ECS
- Penerapan pada Platform Komputasi EC2/On-Premises

Penerapan pada AWS Platform Komputasi Lambda

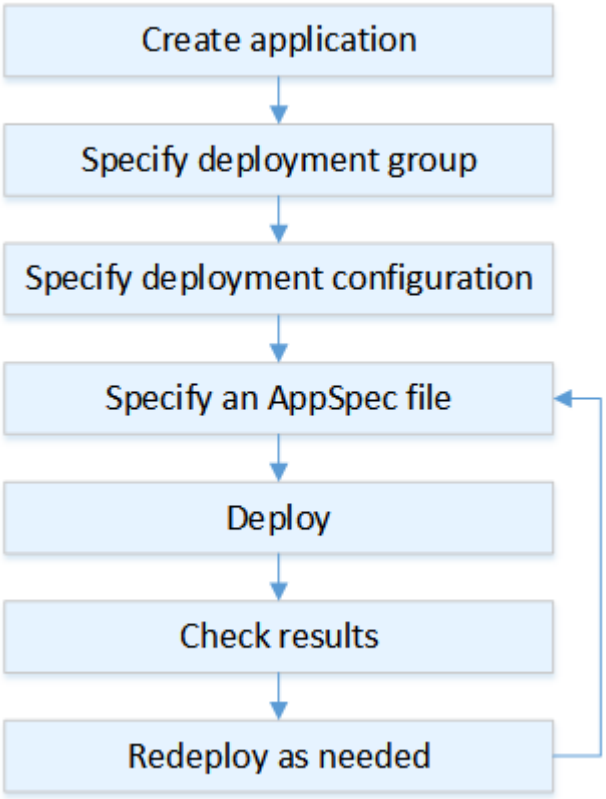
Topik ini memberikan informasi tentang komponen dan alur kerja CodeDeploy penerapan yang menggunakan platform komputasi Lambda AWS .

Topik

- Alur kerja penerapan pada AWS Lambda platform komputasi
- Mengunggah revisi aplikasi Anda
- Membuat grup aplikasi dan penerapan
- Menerapkan revisi aplikasi
- Memperbarui aplikasi Anda
- Penerapan yang dihentikan dan gagal
- Penerapan ulang dan rollback penerapan

Alur kerja penerapan pada AWS Lambda platform komputasi

Diagram berikut menunjukkan langkah-langkah utama dalam penyebaran AWS Lambda fungsi baru dan diperbarui.



Langkah-langkah ini meliputi:

1. Buat aplikasi dan berikan nama yang secara unik mengidentifikasi revisi aplikasi yang ingin Anda terapkan. Untuk menerapkan fungsi Lambda, pilih platform komputasi AWS Lambda saat Anda membuat aplikasi. CodeDeploy menggunakan nama ini selama penerapan untuk memastikannya merujuk komponen penerapan yang benar, seperti grup penyebaran, konfigurasi penerapan, dan revisi aplikasi. Untuk informasi selengkapnya, lihat Buat aplikasi dengan CodeDeploy.
2. Siapkan grup penyebaran dengan menentukan nama grup penyebaran Anda.
3. Pilih konfigurasi penerapan untuk menentukan bagaimana lalu lintas digeser dari versi AWS Lambda fungsi asli Anda ke versi fungsi Lambda baru Anda. Untuk informasi selengkapnya, lihat View Deployment Configuration Details.
4. Mengunggah file spesifikasi aplikasi (AppSpec file) ke Amazon S3. AppSpec File menentukan versi fungsi Lambda dan fungsi Lambda yang digunakan untuk memvalidasi penerapan Anda. Jika Anda tidak ingin membuat AppSpec file, Anda dapat menentukan versi fungsi Lambda dan fungsi validasi penerapan Lambda secara langsung di konsol menggunakan YAMG atau JSON. Untuk informasi selengkapnya, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.

5. Menerapkan revisi aplikasi Anda ke grup penyebaran. AWS CodeDeploy menerapkan revisi fungsi Lambda yang Anda tentukan. Lalu lintas dialihkan ke revisi fungsi Lambda Anda menggunakan file AppSpec deployment yang Anda pilih saat Anda membuat aplikasi Anda. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy.
6. Periksa hasil penerapan. Untuk informasi selengkapnya, lihat Memantau penyebaran di CodeDeploy.

Mengunggah revisi aplikasi Anda

Tempatkan AppSpec file di Amazon S3 atau masukkan langsung ke konsol atau. AWS CLI Untuk informasi selengkapnya, lihat Application Specification Files.

Membuat grup aplikasi dan penerapan

Grup CodeDeploy penyebaran pada platform komputasi AWS Lambda mengidentifikasi kumpulan satu atau beberapa file. AppSpec Setiap AppSpec file dapat menggunakan satu versi fungsi Lambda. Grup penyebaran juga mendefinisikan satu set opsi konfigurasi untuk penerapan masa depan, seperti alarm dan konfigurasi rollback.

Menerapkan revisi aplikasi

Sekarang Anda siap untuk menerapkan revisi fungsi yang ditentukan dalam AppSpec file ke grup penyebaran. Anda dapat menggunakan CodeDeploy konsol atau perintah create-deployment. Ada parameter yang dapat Anda tentukan untuk mengontrol penerapan Anda, termasuk revisi, grup penyebaran, dan konfigurasi penerapan.

Memperbarui aplikasi Anda

Anda dapat membuat pembaruan pada aplikasi Anda dan kemudian menggunakan CodeDeploy konsol atau memanggil perintah create-deployment untuk mendorong revisi.

Penerapan yang dihentikan dan gagal

Anda dapat menggunakan CodeDeploy konsol atau perintah stop-deployment untuk menghentikan penerapan. Saat Anda mencoba menghentikan penerapan, salah satu dari tiga hal terjadi:

- Penerapan berhenti, dan operasi mengembalikan status berhasil. Dalam kasus ini, tidak ada lagi peristiwa siklus hidup penerapan yang dijalankan pada grup penerapan untuk penerapan yang dihentikan.

- Penyebaran tidak segera berhenti, dan operasi mengembalikan status tertunda. Dalam kasus ini, beberapa peristiwa siklus hidup penerapan mungkin masih berjalan di grup penerapan. Setelah operasi yang tertunda selesai, panggilan berikutnya untuk menghentikan penyebaran mengembalikan status berhasil.
- Penerapan tidak dapat berhenti, dan operasi mengembalikan kesalahan. Untuk informasi selengkapnya, lihat ErrorInformationdan Kesalahan umum di Referensi AWS CodeDeploy API.

Seperti penerapan yang dihentikan, penerapan yang gagal dapat mengakibatkan beberapa peristiwa siklus hidup penerapan telah dijalankan. Untuk mengetahui mengapa penerapan gagal, Anda dapat menggunakan CodeDeploy konsol atau menganalisis data file log dari penerapan yang gagal. Untuk informasi selengkapnya, lihat Revisi aplikasi dan pembersihan file log dan Lihat data log untuk CodeDeploy EC2/On-Premises penerapan.

Penerapan ulang dan rollback penerapan

CodeDeploy mengimplementasikan rollback dengan menerapkan kembali, sebagai penerapan baru, revisi yang sebelumnya diterapkan.

Anda dapat mengonfigurasi grup penerapan untuk secara otomatis memutar kembali penerapan saat kondisi tertentu terpenuhi, termasuk saat penerapan gagal atau ambang batas pemantauan alarm terpenuhi. Anda juga dapat mengganti pengaturan rollback yang ditentukan untuk grup penerapan dalam penerapan individual.

Anda juga dapat memilih untuk mengembalikan penerapan yang gagal dengan menerapkan ulang revisi yang sebelumnya digunakan secara manual.

Dalam semua kasus, penerapan baru atau rolled-back diberi ID penerapan sendiri. Daftar penerapan yang dapat Anda lihat di CodeDeploy konsol menunjukkan mana yang merupakan hasil dari penerapan otomatis.

Untuk informasi selengkapnya, lihat Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.

Deployment di Platform Komputasi Amazon ECS

Topik ini memberikan informasi tentang komponen dan alur kerja CodeDeploy penerapan yang menggunakan platform komputasi Amazon ECS.

Topik

- Sebelum memulai deployment Amazon ECS
- Alur kerja penerapan (tingkat tinggi) pada platform komputasi Amazon ECS
- Apa yang terjadi selama penyebaran Amazon ECS
- Mengunggah revisi aplikasi Anda
- Membuat grup aplikasi dan penerapan
- Menerapkan revisi aplikasi
- Memperbarui aplikasi Anda
- Penerapan yang dihentikan dan gagal
- Penerapan ulang dan rollback penerapan
- blue/green Penerapan Amazon ECS melalui AWS CloudFormation

Sebelum memulai deployment Amazon ECS

Sebelum Anda memulai penyebaran aplikasi Amazon ECS, Anda harus menyiapkan yang berikut ini. Beberapa persyaratan ditentukan saat Anda membuat grup penyebaran, dan beberapa ditentukan dalam AppSpec file.

Persyaratan	Dimana ditentukan
Kluster Amazon ECS	Grup deployment
Layanan Amazon ECS	Grup deployment
Application Load Balancer atau Network Load Balancer	Grup deployment
Listener produksi	Grup deployment
Uji pendengar (opsional)	Grup deployment
Dua grup target	Grup deployment
Definisi tugas Amazon ECS	AppSpec berkas
Nama kontainer	AppSpec berkas
Pelabuhan kontainer	AppSpec berkas

Kluster Amazon ECS

Cluster Amazon ECS adalah pengelompokan tugas atau layanan yang logis. Anda menentukan klaster Amazon ECS yang berisi layanan Amazon ECS saat membuat grup CodeDeploy penerapan aplikasi. Untuk informasi selengkapnya, lihat klaster Amazon ECS di Panduan Pengguna Layanan Amazon Elastic Container.

Layanan Amazon ECS

Layanan Amazon ECS memelihara dan menjalankan instans definisi tugas yang ditentukan dalam klaster Amazon ECS. Layanan Amazon ECS Anda harus diaktifkan untuk CodeDeploy. Secara default, layanan Amazon ECS diaktifkan untuk penerapan Amazon ECS. Saat membuat grup penerapan, Anda memilih untuk menerapkan layanan Amazon ECS yang ada di kluster Amazon ECS Anda. Untuk informasi selengkapnya, lihat layanan Amazon ECS di Panduan Pengguna Layanan Amazon Elastic Container.

Application Load Balancer atau Network Load Balancer

Anda harus menggunakan Elastic Load Balancing dengan layanan Amazon ECS yang ingin Anda perbarui dengan penyebaran Amazon ECS. Anda dapat menggunakan Application Load Balancer atau Network Load Balancer. Kami merekomendasikan Application Load Balancer sehingga Anda dapat memanfaatkan fitur-fitur seperti pemetaan port dinamis dan routing berbasis jalur dan aturan prioritas. Anda menentukan penyeimbang beban saat membuat grup CodeDeploy penerapan aplikasi. Untuk informasi selengkapnya, lihat Siapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS dan Membuat penyeimbang beban di Panduan Pengguna Amazon Elastic Container Service.

Satu atau dua pendengar

Listener digunakan oleh penyeimbang beban Anda untuk mengarahkan lalu lintas ke grup target Anda. Diperlukan satu pendengar produksi. Anda dapat menentukan pendengar pengujian kedua opsional yang mengarahkan lalu lintas ke set tugas pengganti saat Anda menjalankan pengujian validasi. Anda menentukan satu atau kedua pendengar saat membuat grup penerapan. Jika Anda menggunakan konsol Amazon ECS untuk membuat layanan Amazon ECS Anda, pendengar Anda dibuat untuk Anda. Untuk informasi selengkapnya, lihat Pendengar untuk penyeimbang beban aplikasi Anda di Panduan Pengguna Elastic Load Balancing dan Membuat layanan di Panduan PenggunaAmazon Elastic Container Service.

Dua kelompok sasaran Amazon ECS

Kelompok sasaran digunakan untuk mengarahkan lalu lintas ke target yang terdaftar. Penyebaran Amazon ECS memerlukan dua grup target: satu untuk set tugas asli aplikasi Amazon ECS

Anda dan satu untuk set tugas penggantinya. Selama penerapan, CodeDeploy membuat set tugas pengganti dan mengalihkan lalu lintas dari tugas asli yang disetel ke yang baru. Anda menentukan grup target saat membuat grup penerapan CodeDeploy aplikasi.

Selama penerapan, CodeDeploy tentukan grup target mana yang terkait dengan set tugas di layanan Amazon ECS Anda yang memiliki status `PRIMARY` (ini adalah kumpulan tugas asli) dan mengaitkan satu grup target dengannya, lalu mengaitkan grup target lainnya dengan set tugas pengganti. Jika Anda melakukan penerapan lain, grup target yang terkait dengan set tugas asli penerapan saat ini dikaitkan dengan set tugas pengganti penerapan berikutnya. Untuk informasi selengkapnya, lihat Grup target untuk penyeimbang beban aplikasi Anda di Panduan Pengguna Elastic Load Balancing.

Definisi tugas Amazon ECS

Definisi tugas diperlukan untuk menjalankan wadah Docker yang berisi aplikasi Amazon ECS Anda. Anda menentukan ARN definisi tugas Anda dalam file CodeDeploy aplikasi Anda. AppSpec Untuk informasi selengkapnya, lihat definisi tugas Amazon ECS di Panduan Pengguna Layanan Kontainer Elastis Amazon danAppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

Wadah untuk aplikasi Amazon ECS Anda

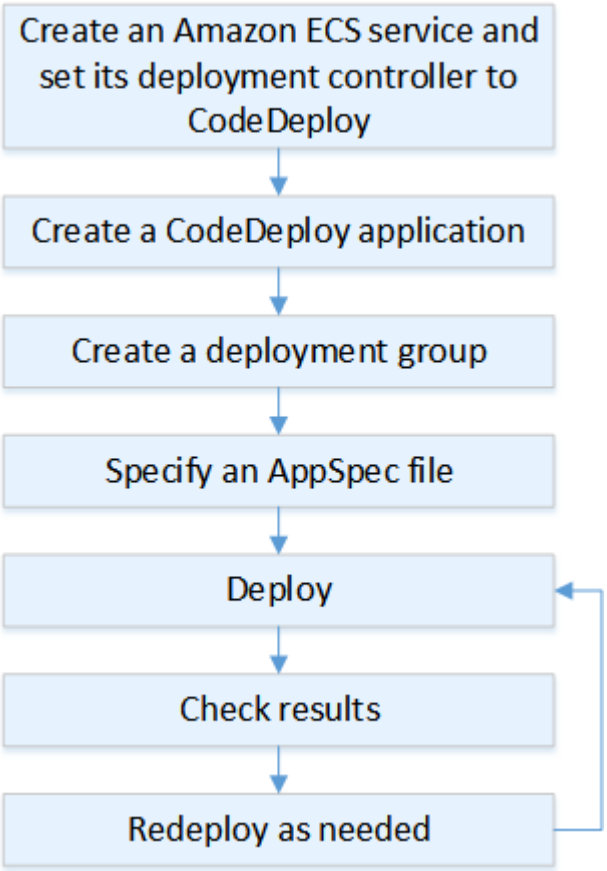
Container Docker adalah unit perangkat lunak yang mengemas kode dan dependensinya sehingga aplikasi Anda dapat berjalan. Sebuah wadah mengisolasi aplikasi Anda sehingga berjalan di lingkungan komputasi yang berbeda. Penyeimbang beban Anda mengarahkan lalu lintas ke kontainer di set tugas aplikasi Amazon ECS Anda. Anda menentukan nama penampung Anda dalam AppSpec file CodeDeploy aplikasi Anda. Penampung yang ditentukan dalam AppSpec file Anda harus merupakan salah satu kontainer yang ditentukan dalam definisi tugas Amazon ECS Anda. Untuk informasi selengkapnya, lihat Apa itu Amazon Elastic Container Service? di Panduan Pengguna Layanan Kontainer Elastis Amazon danAppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

Port untuk set tugas pengganti Anda

Selama penerapan Amazon ECS, penyeimbang beban mengarahkan lalu lintas ke port ini pada penampung yang ditentukan dalam file aplikasi Anda CodeDeploy. AppSpec Anda menentukan port dalam AppSpec file CodeDeploy aplikasi Anda. Untuk informasi selengkapnya, lihat AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

Alur kerja penerapan (tingkat tinggi) pada platform komputasi Amazon ECS

Diagram berikut menunjukkan langkah-langkah utama dalam penyebaran layanan Amazon ECS yang diperbarui.



Langkah-langkah ini meliputi:

1. Buat AWS CodeDeploy aplikasi dengan menentukan nama yang secara unik mewakili apa yang ingin Anda gunakan. Untuk menerapkan aplikasi Amazon ECS, di AWS CodeDeploy aplikasi Anda, pilih platform komputasi Amazon ECS. CodeDeploy menggunakan aplikasi selama penerapan untuk mereferensikan komponen penyebaran yang benar, seperti grup penyebaran, grup target, pendengar, dan perilaku pengalihan rute lalu lintas, dan revisi aplikasi. Untuk informasi selengkapnya, lihat Buat aplikasi dengan CodeDeploy.
2. Siapkan grup penerapan dengan menentukan:
   - Nama grup penyebaran.
   - Cluster Amazon ECS dan nama layanan Anda. Pengontrol penyebaran layanan Amazon ECS harus disetel ke. CodeDeploy

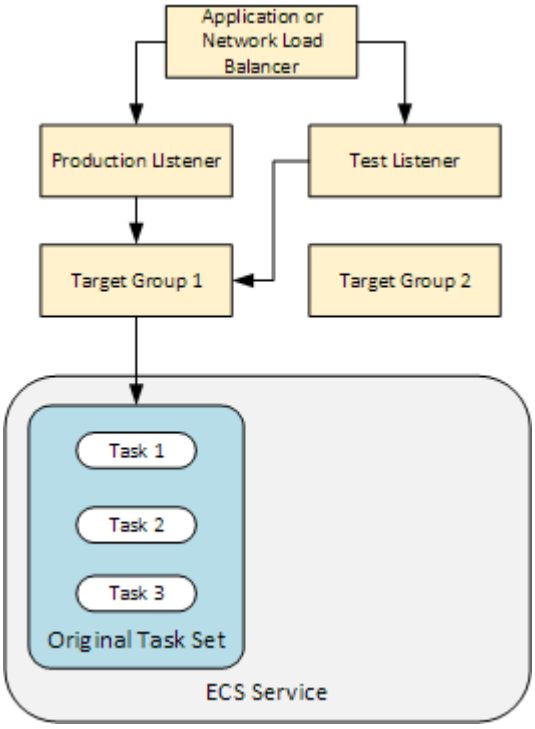
- Pendengar produksi, pendengar uji opsional, dan grup target yang digunakan selama penerapan.
- Pengaturan penerapan, seperti kapan mengalihkan lalu lintas produksi ke tugas Amazon ECS pengganti yang disetel di layanan Amazon ECS Anda dan kapan harus menghentikan tugas Amazon ECS asli yang disetel di layanan Amazon ECS Anda.
- Pengaturan opsional, seperti pemicu, alarm, dan perilaku rollback.

3. Tentukan file spesifikasi aplikasi (AppSpec file). Anda dapat mengunggahnya ke Amazon S3, memasukkannya ke konsol dalam format YAMAL atau JSON, atau menentukannya dengan atau SDK. AWS CLI AppSpec File tersebut menentukan definisi tugas Amazon ECS untuk penerapan, nama kontainer, dan pemetaan port yang digunakan untuk merutekan lalu lintas, dan fungsi Lambda berjalan setelah kait siklus hidup penerapan. Nama kontainer harus berupa wadah dalam definisi tugas Amazon ECS Anda. Untuk informasi selengkapnya, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
4. Terapkan revisi aplikasi Anda. AWS CodeDeploy mengalihkan lalu lintas dari versi asli tugas yang ditetapkan di layanan Amazon ECS Anda ke set tugas pengganti yang baru. Grup target yang ditentukan dalam grup penyebaran digunakan untuk melayani lalu lintas ke set tugas asli dan pengganti. Setelah penerapan selesai, set tugas asli dihentikan. Anda dapat menentukan pendengar pengujian opsional untuk menyajikan lalu lintas pengujian ke versi pengganti Anda sebelum lalu lintas dialihkan ke versi tersebut. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy.
5. Periksa hasil penerapan. Untuk informasi selengkapnya, lihat Memantau penyebaran di CodeDeploy.

Apa yang terjadi selama penyebaran Amazon ECS

Sebelum penerapan Amazon ECS dengan pendengar pengujian dimulai, Anda harus mengonfigurasi komponennya. Untuk informasi selengkapnya, lihat Sebelum memulai deployment Amazon ECS.

Diagram berikut menunjukkan hubungan antara komponen-komponen ini ketika penyebaran Amazon ECS siap untuk dimulai.



Saat penerapan dimulai, peristiwa siklus hidup penerapan mulai dijalankan satu per satu. Beberapa peristiwa siklus hidup adalah kait yang hanya menjalankan fungsi Lambda yang ditentukan dalam file. AppSpec Peristiwa siklus hidup penerapan dalam tabel berikut tercantum dalam urutan yang dieksekusi. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan Amazon ECS.

Acara siklus hidup	Tindakan peristiwa siklus hidup
`BeforeInstall` (pengait untuk fungsi Lambda)	Jalankan fungsi Lambda.
Menginstal	Siapkan set tugas pengganti.
`AfterInstall` (pengait untuk fungsi Lambda)	Jalankan fungsi Lambda.
AllowTestTraffic	Rutekan lalu lintas dari pendengar pengujian ke grup target 2.
`AfterAllowTestTraffic` (pengait untuk fungsi Lambda)	Jalankan fungsi Lambda.

Acara siklus hidup	Tindakan peristiwa siklus hidup
`BeforeAllowTraffic` (pengait untuk fungsi Lambda)	Jalankan fungsi Lambda.
AllowTraffic	Rutekan lalu lintas dari pendengar produksi ke grup target 2.
`AfterAllowTraffic`	Jalankan fungsi Lambda.

> **Note**
>
> Fungsi Lambda di hook adalah opsional.

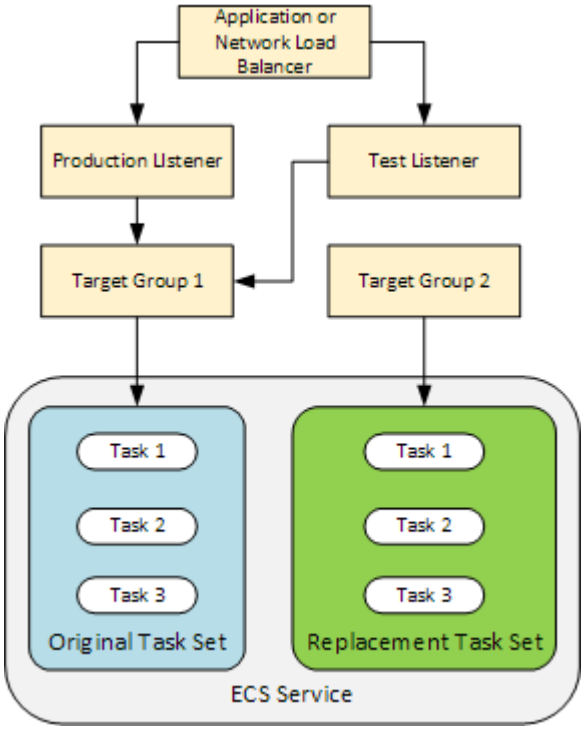
1. Jalankan fungsi Lambda apa pun yang ditentukan dalam `BeforeInstall` hook dalam file. AppSpec
2. Selama acara `Install` siklus hidup:

   a. Set tugas pengganti dibuat di layanan Amazon ECS Anda.

   b. Aplikasi kontainer yang diperbarui diinstal ke dalam set tugas pengganti.

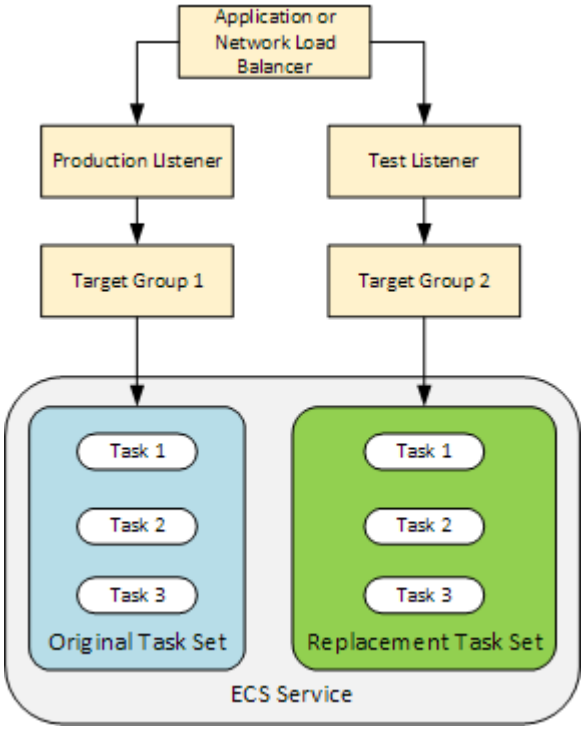
   c. Kelompok target kedua dikaitkan dengan set tugas pengganti.

   Diagram ini menunjukkan komponen penyebaran dengan set tugas pengganti baru. Aplikasi kontainer ada di dalam set tugas ini. Set tugas terdiri dari tiga tugas. (Aplikasi dapat memiliki sejumlah tugas.) Kelompok target kedua sekarang dikaitkan dengan set tugas pengganti.



3. Jalankan fungsi Lambda apa pun yang ditentukan dalam `AfterInstall` hook dalam file. AppSpec

4. `AllowTestTraffic`Acara ini dipanggil. Selama peristiwa siklus hidup ini, pendengar pengujian merutekan lalu lintas ke aplikasi kontainer yang diperbarui.



5. Jalankan fungsi Lambda apa pun yang ditentukan dalam `AfterAllowTestTraffic` hook dalam file. AppSpec Fungsi Lambda dapat memvalidasi penerapan menggunakan lalu lintas pengujian. Misalnya, fungsi Lambda dapat menyajikan lalu lintas ke listener pengujian dan melacak metrik dari set tugas pengganti. Jika rollback dikonfigurasi, Anda dapat mengonfigurasi CloudWatch alarm yang memicu rollback saat tes validasi di fungsi Lambda Anda gagal.

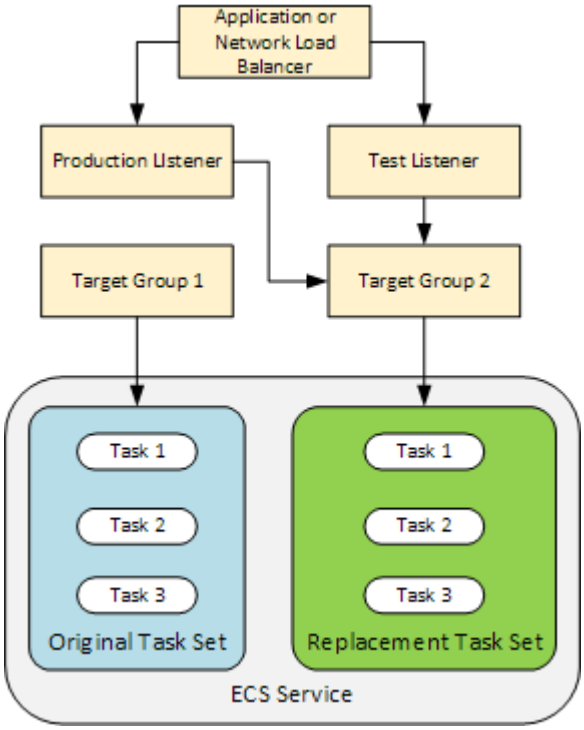
   Setelah tes validasi selesai, salah satu hal berikut terjadi:

   - Jika validasi gagal dan rollback dikonfigurasi, status penerapan ditandai `Failed` dan komponen kembali ke statusnya saat penerapan dimulai.
   - Jika validasi gagal dan rollback tidak dikonfigurasi, status penerapan ditandai `Failed` dan komponen tetap dalam keadaan saat ini.
   - Jika validasi berhasil, penerapan berlanjut ke hook. `BeforeAllowTraffic`

   Lihat informasi selengkapnya di Memantau penyebaran dengan CloudWatch alarm di CodeDeploy, Rollback otomatis, dan Mengonfigurasi opsi lanjutan untuk grup deployment.

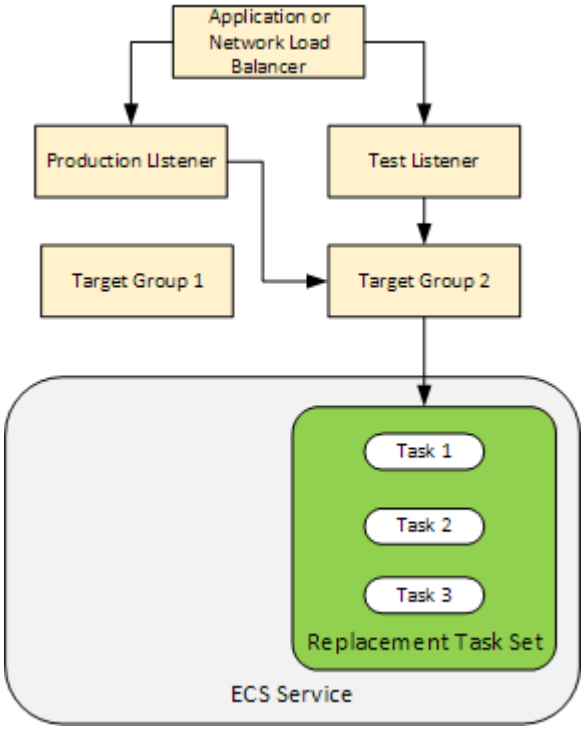
6. Jalankan fungsi Lambda apa pun yang ditentukan dalam `BeforeAllowTraffic` hook dalam file. AppSpec

7. `AllowTraffic`Acara ini dipanggil. Lalu lintas produksi dialihkan dari tugas asli yang disetel ke set tugas pengganti. Diagram berikut menunjukkan set tugas penggantian menerima lalu lintas produksi.



8. Jalankan fungsi Lambda apa pun yang ditentukan dalam `AfterAllowTraffic` hook dalam file. AppSpec

9. Setelah semua peristiwa berhasil, status penerapan disetel ke `Succeeded` dan set tugas asli dihapus.



Mengunggah revisi aplikasi Anda

Tempatkan AppSpec file di Amazon S3 atau masukkan langsung ke konsol atau. AWS CLI Untuk informasi selengkapnya, lihat Application Specification Files.

Membuat grup aplikasi dan penerapan

Grup CodeDeploy penerapan pada platform komputasi Amazon ECS mengidentifikasi pendengar untuk menayangkan lalu lintas ke aplikasi Amazon ECS Anda yang diperbarui dan dua grup target yang digunakan selama penerapan Anda. Grup penyebaran juga mendefinisikan serangkaian opsi konfigurasi, seperti alarm dan konfigurasi rollback.

Menerapkan revisi aplikasi

Sekarang Anda siap untuk menerapkan layanan Amazon ECS yang diperbarui yang ditentukan dalam grup penerapan Anda. Anda dapat menggunakan CodeDeploy konsol atau perintah create-deployment. Ada parameter yang dapat Anda tentukan untuk mengontrol penyebaran Anda, termasuk grup revisi dan penerapan.

Memperbarui aplikasi Anda

Anda dapat membuat pembaruan pada aplikasi Anda dan kemudian menggunakan CodeDeploy konsol atau memanggil perintah create-deployment untuk mendorong revisi.

Penerapan yang dihentikan dan gagal

Anda dapat menggunakan CodeDeploy konsol atau perintah stop-deployment untuk menghentikan penerapan. Saat Anda mencoba menghentikan penerapan, salah satu dari tiga hal terjadi:

- Penerapan berhenti, dan operasi mengembalikan status berhasil. Dalam kasus ini, tidak ada lagi peristiwa siklus hidup penerapan yang dijalankan pada grup penerapan untuk penerapan yang dihentikan.
- Penyebaran tidak segera berhenti, dan operasi mengembalikan status tertunda. Dalam kasus ini, beberapa peristiwa siklus hidup penerapan mungkin masih berjalan di grup penerapan. Setelah operasi yang tertunda selesai, panggilan berikutnya untuk menghentikan penyebaran mengembalikan status berhasil.
- Penerapan tidak dapat berhenti, dan operasi mengembalikan kesalahan. Untuk informasi selengkapnya, lihat Informasi kesalahan dan Kesalahan umum di Referensi AWS CodeDeploy API.

Penerapan ulang dan rollback penerapan

CodeDeploy mengimplementasikan rollback dengan mengalihkan lalu lintas dari tugas pengganti yang disetel ke set tugas asli.

Anda dapat mengonfigurasi grup penerapan untuk secara otomatis memutar kembali penerapan saat kondisi tertentu terpenuhi, termasuk saat penerapan gagal atau ambang batas pemantauan alarm terpenuhi. Anda juga dapat mengganti pengaturan rollback yang ditentukan untuk grup penerapan dalam penerapan individual.

Anda juga dapat memilih untuk mengembalikan penerapan yang gagal dengan menerapkan ulang revisi yang sebelumnya digunakan secara manual.

Dalam semua kasus, penerapan baru atau rolled-back diberi ID penerapan sendiri. CodeDeploy Konsol menampilkan daftar penerapan yang merupakan hasil dari penerapan otomatis.

Jika Anda menerapkan ulang, grup target yang terkait dengan set tugas asli penerapan saat ini dikaitkan dengan set tugas pengganti penempatan ulang.

Untuk informasi selengkapnya, lihat Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.

blue/green Penerapan Amazon ECS melalui AWS CloudFormation

Anda dapat menggunakan AWS CloudFormation untuk mengelola blue/green penyebaran Amazon ECS melalui. CodeDeploy Untuk informasi selengkapnya, lihat Buat blue/green penyebaran Amazon ECS melalui CloudFormation.

> **Note**
>
> Mengelola blue/green penerapan Amazon ECS tidak CloudFormation tersedia di Wilayah Asia Pasifik (Osaka).

Penerapan pada Platform Komputasi EC2/On-Premises

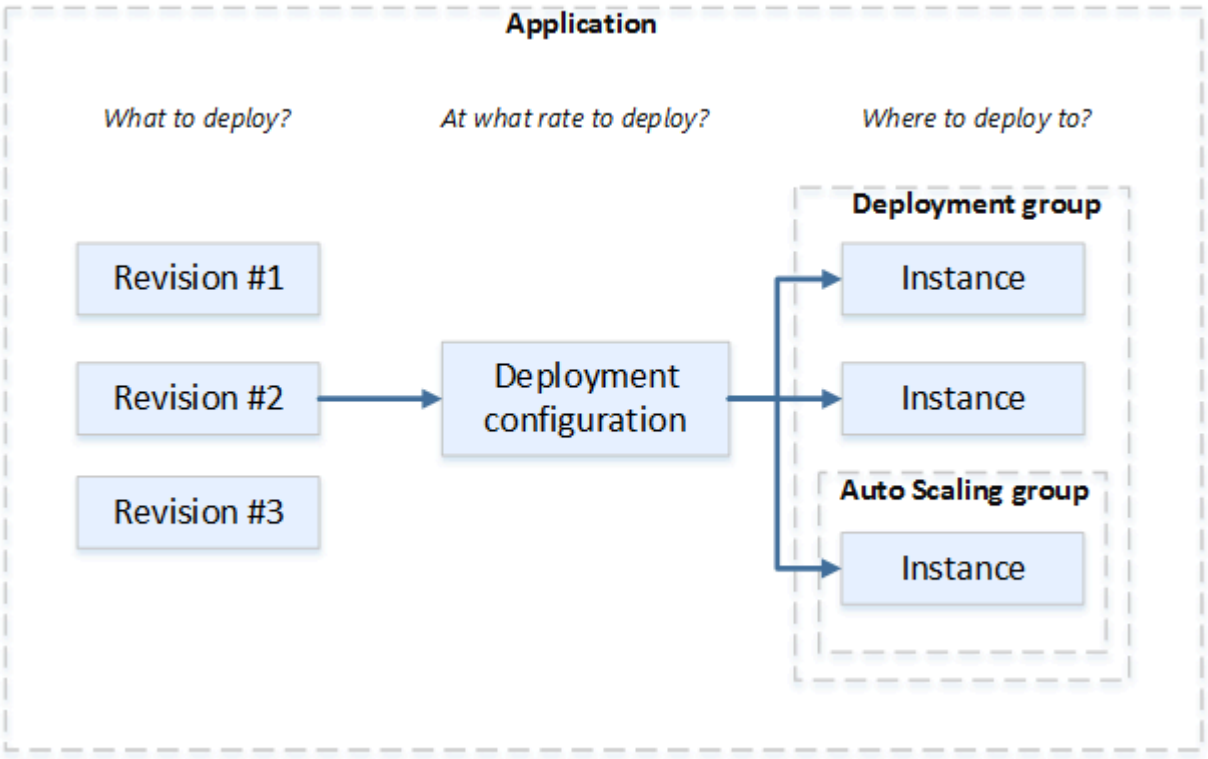
Topik ini memberikan informasi tentang komponen dan alur kerja CodeDeploy penerapan yang menggunakan platform komputasi. EC2/On-Premises Untuk informasi tentang blue/green penerapan, lihat. Ikhtisar blue/green penerapan

Topik

- Komponen penyebaran pada platform EC2/on-premises komputasi
- Alur kerja penerapan pada platform komputasi EC2/on-premises
- Menyiapkan instance
- Mengunggah revisi aplikasi Anda
- Membuat grup aplikasi dan penerapan
- Menerapkan revisi aplikasi
- Memperbarui aplikasi Anda
- Penerapan yang dihentikan dan gagal
- Penerapan ulang dan rollback penerapan

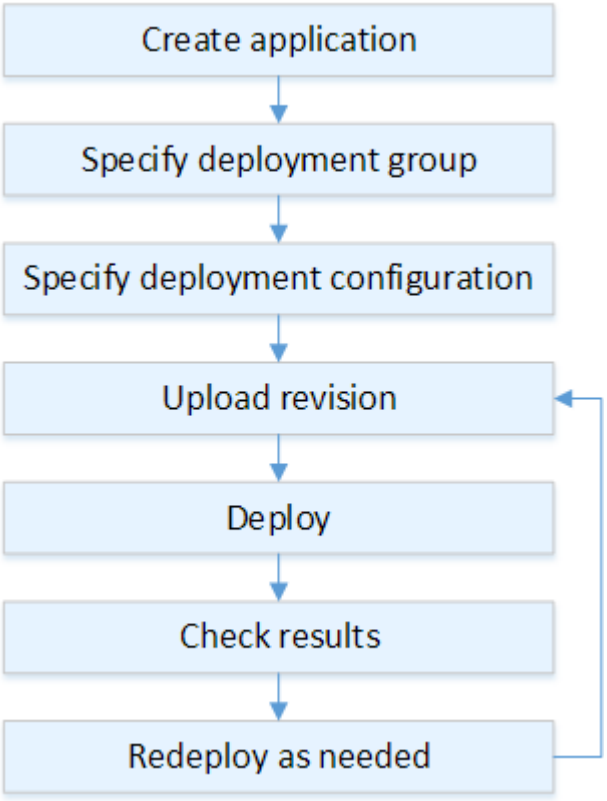
Komponen penyebaran pada platform EC2/on-premises komputasi

Diagram berikut menunjukkan komponen dalam CodeDeploy penyebaran pada platform EC2/On-Premises komputasi.



Alur kerja penerapan pada platform komputasi EC2/on-premises

Diagram berikut menunjukkan langkah-langkah utama dalam penyebaran revisi aplikasi:



Langkah-langkah ini meliputi:

1. Buat aplikasi dan berikan nama yang secara unik mengidentifikasi revisi aplikasi yang ingin Anda terapkan dan platform komputasi untuk aplikasi Anda. CodeDeploy menggunakan nama ini selama penerapan untuk memastikannya merujuk komponen penerapan yang benar, seperti grup penyebaran, konfigurasi penerapan, dan revisi aplikasi. Untuk informasi selengkapnya, lihat Buat aplikasi dengan CodeDeploy.
2. Siapkan grup penerapan dengan menentukan jenis penerapan dan instance yang ingin Anda gunakan untuk menerapkan revisi aplikasi. Penerapan di tempat memperbarui instance dengan revisi aplikasi terbaru. blue/green Penerapan mendaftarkan sekumpulan instance pengganti untuk grup penerapan dengan penyeimbang beban dan membatalkan pendaftaran instans asli.

   Anda dapat menentukan tag yang diterapkan ke instance, nama grup Amazon EC2 Auto Scaling, atau keduanya.

   Jika Anda menentukan satu grup tag dalam grup penyebaran, CodeDeploy deploy ke instance yang memiliki setidaknya satu tag yang ditentukan diterapkan. Jika Anda menentukan dua atau beberapa grup tag, CodeDeploy hanya akan diterapkan ke instance yang memenuhi kriteria untuk masing-masing grup tag. Untuk informasi selengkapnya, lihat Tagging Instances for Deployments.

Dalam semua kasus, instans harus dikonfigurasi untuk digunakan dalam penerapan (yaitu, mereka harus ditandai atau milik grup Amazon EC2 Auto Scaling) dan memiliki agen diinstal dan CodeDeploy dijalankan.

Anda dapat menggunakan CloudFormation template untuk menyiapkan instans Amazon EC2 dengan cepat berdasarkan Amazon Linux atau Windows Server. Kami juga menyediakan CodeDeploy agen mandiri sehingga Anda dapat menginstalnya di Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL), atau instans Windows Server. Untuk informasi selengkapnya, lihat Buat grup penerapan dengan CodeDeploy.

Anda juga dapat menentukan parameter berikut:

- Pemberitahuan Amazon SNS. Buat pemicu yang mengirim pemberitahuan ke pelanggan topik Amazon SNS saat peristiwa tertentu, seperti peristiwa sukses atau gagal, terjadi dalam penerapan dan instance. Untuk informasi selengkapnya, lihat Monitoring Deployments with Amazon SNS Event Notifications.
- Alarm-based manajemen penyebaran. Menerapkan pemantauan CloudWatch alarm Amazon untuk menghentikan penerapan ketika metrik Anda melebihi atau jatuh di bawah ambang batas yang ditetapkan. CloudWatch
- Rollback penerapan otomatis. Konfigurasikan penerapan untuk memutar kembali secara otomatis ke revisi baik yang sebelumnya diketahui saat penerapan gagal atau ambang batas alarm terpenuhi.

3. Tentukan konfigurasi penerapan untuk menunjukkan berapa banyak instance revisi aplikasi Anda harus diterapkan secara bersamaan dan untuk menjelaskan kondisi keberhasilan dan kegagalan untuk penerapan. Untuk informasi selengkapnya, lihat View Deployment Configuration Details.
4. Unggah revisi aplikasi ke Amazon GitHub S3 atau. Selain file yang ingin Anda gunakan dan skrip apa pun yang ingin Anda jalankan selama penerapan, Anda harus menyertakan file spesifikasi aplikasi (file)AppSpec . File ini berisi instruksi penyebaran, seperti tempat untuk menyalin file ke setiap instance dan kapan menjalankan skrip penerapan. Untuk informasi selengkapnya, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
5. Menerapkan revisi aplikasi Anda ke grup penyebaran. CodeDeploy Agen pada setiap instans dalam grup penerapan menyalin revisi aplikasi Anda dari Amazon S3 GitHub atau ke instance. CodeDeploy Agen kemudian membuka bundel revisi, dan menggunakan AppSpec file, menyalin file ke lokasi yang ditentukan dan mengeksekusi skrip penerapan apa pun. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy.

6. Periksa hasil penerapan. Untuk informasi selengkapnya, lihat Memantau penyebaran di CodeDeploy.
7. Menerapkan ulang revisi. Anda mungkin ingin melakukan ini jika Anda perlu memperbaiki bug di konten sumber, atau menjalankan skrip penerapan dalam urutan yang berbeda, atau mengatasi penerapan yang gagal. Untuk melakukan ini, gabungkan ulang konten sumber yang direvisi, skrip penerapan apa pun, dan AppSpec file menjadi revisi baru, lalu unggah revisi ke bucket atau repositori Amazon S3. GitHub Kemudian jalankan penerapan baru ke grup penyebaran yang sama dengan revisi baru. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy.

Menyiapkan instance

Anda harus menyiapkan instance sebelum dapat menerapkan revisi aplikasi untuk pertama kalinya. Jika revisi aplikasi memerlukan tiga server produksi dan dua server cadangan, Anda meluncurkan atau menggunakan lima instance.

Untuk menyediakan instance secara manual:

1. Instal CodeDeploy agen pada instance. CodeDeploy Agen dapat diinstal pada Amazon Linux, Ubuntu Server, RHEL, dan Windows Server instance.
2. Aktifkan penandaan, jika Anda menggunakan tag untuk mengidentifikasi instance dalam grup penyebaran. CodeDeploy bergantung pada tag untuk mengidentifikasi dan mengelompokkan instance ke dalam grup CodeDeploy penerapan. Meskipun tutorial Memulai menggunakan keduanya, Anda cukup menggunakan kunci atau nilai untuk menentukan tag untuk grup penyebaran.
3. Luncurkan instans Amazon EC2 dengan profil instans IAM terlampir. Profil instans IAM harus dilampirkan ke instans Amazon EC2 saat diluncurkan agar agen CodeDeploy dapat memverifikasi identitas instans.
4. Membuat peran layanan. Berikan akses layanan sehingga CodeDeploy dapat memperluas tag di AWS akun Anda.

Untuk penerapan awal, CloudFormation template melakukan semua ini untuk Anda. Ini membuat dan mengkonfigurasi instans Amazon EC2 tunggal baru berdasarkan Amazon Linux atau Windows Server dengan CodeDeploy agen yang sudah diinstal. Untuk informasi selengkapnya, lihat Bekerja dengan instance untuk CodeDeploy.

> **Note**
>
> Untuk blue/green penerapan, Anda dapat memilih antara menggunakan instance yang sudah Anda miliki untuk lingkungan pengganti atau mengizinkan CodeDeploy penyediaan instance baru untuk Anda sebagai bagian dari proses penerapan.

Mengunggah revisi aplikasi Anda

Tempatkan AppSpec file di bawah folder root dalam struktur folder konten sumber aplikasi Anda. Untuk informasi selengkapnya, lihat Application Specification Files.

Bundel struktur folder konten sumber aplikasi ke dalam format file arsip seperti zip, tar, atau tar terkompresi. Unggah file arsip (revisi) ke bucket GitHub atau repositori Amazon S3.

> **Note**
>
> Format file arsip tar dan tar terkompresi (.tar dan.tar.gz) tidak didukung untuk instance Windows Server.

Membuat grup aplikasi dan penerapan

Grup CodeDeploy penerapan mengidentifikasi kumpulan instance berdasarkan tag, nama grup Amazon EC2 Auto Scaling, atau keduanya. Beberapa revisi aplikasi dapat diterapkan ke instance yang sama. Revisi aplikasi dapat diterapkan ke beberapa instance.

Misalnya, Anda dapat menambahkan tag “Prod” ke tiga server produksi dan “Backup” ke dua server cadangan. Kedua tag ini dapat digunakan untuk membuat dua grup penyebaran yang berbeda dalam CodeDeploy aplikasi, memungkinkan Anda memilih kumpulan server mana (atau keduanya) yang harus berpartisipasi dalam penerapan.

Anda dapat menggunakan beberapa grup tag dalam grup penerapan untuk membatasi penerapan ke kumpulan instance yang lebih kecil. Untuk informasi, lihat Tagging Instances for Deployments.

Menerapkan revisi aplikasi

Sekarang Anda siap untuk menerapkan revisi aplikasi Anda dari Amazon S3 GitHub atau ke grup penyebaran. Anda dapat menggunakan CodeDeploy konsol atau perintah create-deployment. Ada

parameter yang dapat Anda tentukan untuk mengontrol penerapan Anda, termasuk revisi, grup penyebaran, dan konfigurasi penerapan.

Memperbarui aplikasi Anda

Anda dapat membuat pembaruan pada aplikasi Anda dan kemudian menggunakan CodeDeploy konsol atau memanggil perintah create-deployment untuk mendorong revisi.

Penerapan yang dihentikan dan gagal

Anda dapat menggunakan CodeDeploy konsol atau perintah stop-deployment untuk menghentikan penerapan. Saat Anda mencoba menghentikan penerapan, salah satu dari tiga hal terjadi:

- Penerapan berhenti, dan operasi mengembalikan status berhasil. Dalam kasus ini, tidak ada lagi peristiwa siklus hidup penerapan yang dijalankan pada grup penerapan untuk penerapan yang dihentikan. Beberapa file mungkin telah disalin, dan beberapa skrip mungkin sudah dijalankan, satu atau lebih instance dalam grup penyebaran.
- Penyebaran tidak segera berhenti, dan operasi mengembalikan status tertunda. Dalam kasus ini, beberapa peristiwa siklus hidup penerapan mungkin masih berjalan di grup penerapan. Beberapa file mungkin telah disalin, dan beberapa skrip mungkin sudah dijalankan, satu atau lebih instance dalam grup penyebaran. Setelah operasi yang tertunda selesai, panggilan berikutnya untuk menghentikan penyebaran mengembalikan status berhasil.
- Penerapan tidak dapat berhenti, dan operasi mengembalikan kesalahan. Untuk informasi selengkapnya, lihat ErrorInformationdan Kesalahan umum di Referensi AWS CodeDeploy API.

Seperti penerapan yang dihentikan, penerapan yang gagal dapat mengakibatkan beberapa peristiwa siklus hidup penerapan telah dijalankan pada satu atau beberapa instance dalam grup penerapan. Untuk mengetahui mengapa penerapan gagal, Anda dapat menggunakan CodeDeploy konsol, memanggil perintah get-deployment-instance, atau menganalisis data file log dari penerapan yang gagal. Untuk informasi selengkapnya, lihat Revisi aplikasi dan pembersihan file log dan Lihat data log untuk CodeDeploy EC2/On-Premises penerapan.

Penerapan ulang dan rollback penerapan

CodeDeploy mengimplementasikan rollback dengan menerapkan kembali, sebagai penerapan baru, revisi yang sebelumnya diterapkan.

Anda dapat mengonfigurasi grup penerapan untuk secara otomatis memutar kembali penerapan saat kondisi tertentu terpenuhi, termasuk saat penerapan gagal atau ambang batas pemantauan alarm

terpenuhi. Anda juga dapat mengganti pengaturan rollback yang ditentukan untuk grup penerapan dalam penerapan individual.

Anda juga dapat memilih untuk mengembalikan penerapan yang gagal dengan menerapkan ulang revisi yang sebelumnya digunakan secara manual.

Dalam semua kasus, penerapan baru atau rolled-back diberi ID penerapan sendiri. Daftar penerapan yang dapat Anda lihat di CodeDeploy konsol menunjukkan mana yang merupakan hasil dari penerapan otomatis.

Lihat informasi yang lebih lengkap di Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.

CodeDeploy spesifikasi aplikasi (AppSpec) file

File spesifikasi aplikasi (AppSpec file), yang unik untuk CodeDeploy, adalah file berformat YAML atau berformat JSON. AppSpec File ini digunakan untuk mengelola setiap penyebaran sebagai serangkaian kait peristiwa siklus hidup, yang didefinisikan dalam file.

Untuk informasi tentang cara membuat AppSpec file yang terbentuk dengan baik, lihatCodeDeploy AppSpec referensi file.

Topik

- AppSpec file di Platform Komputasi Amazon ECS
- AppSpec berkas pada sebuah AWS Lambda platform komputasi
- AppSpec file pada platform EC2/on-premises komputasi
- Bagaimana CodeDeploy agen menggunakan AppSpec file

AppSpec file di Platform Komputasi Amazon ECS

Jika aplikasi Anda menggunakan platform komputasi Amazon ECS, AppSpec file dapat diformat dengan YAMAL atau JSON. Itu juga dapat diketik langsung ke editor di konsol. AppSpec File ini digunakan untuk menentukan:

- Nama layanan Amazon ECS dan nama kontainer serta port yang digunakan untuk mengarahkan lalu lintas ke set tugas baru.
- Fungsi yang akan digunakan sebagai tes validasi.

Anda dapat menjalankan fungsi Lambda validasi setelah peristiwa siklus hidup penerapan. Lihat informasi selengkapnya di AppSpec Bagian 'kait' untuk penerapan Amazon ECS, AppSpec struktur file untuk penerapan Amazon ECS, dan AppSpec Contoh file untuk penyebaran Amazon ECS.

AppSpec berkas pada sebuah AWS Lambda platform komputasi

Jika aplikasi Anda menggunakan platform komputasi AWS Lambda, AppSpec file dapat diformat dengan YAMAL atau JSON. Itu juga dapat diketik langsung ke editor di konsol. AppSpec File ini digunakan untuk menentukan:

- Versi AWS Lambda fungsi untuk menyebarkan.
- Fungsi yang akan digunakan sebagai tes validasi.

Anda dapat menjalankan fungsi Lambda validasi setelah peristiwa siklus hidup penerapan. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk sebuah AWS Penyebaran Lambda.

AppSpec file pada platform EC2/on-premises komputasi

Jika aplikasi Anda menggunakan platform EC2/On-Premises komputasi, AppSpec file selalu YAML-formatted. AppSpec File ini digunakan untuk:

- Petakan file sumber dalam revisi aplikasi Anda ke tujuan mereka pada instance.
- Tentukan izin khusus untuk file yang digunakan.
- Tentukan skrip yang akan dijalankan pada setiap instance pada berbagai tahap proses penyebaran.

Anda dapat menjalankan skrip pada instance setelah banyak peristiwa siklus hidup penerapan individual. CodeDeploy hanya menjalankan skrip yang ditentukan dalam file, tetapi skrip tersebut dapat memanggil skrip lain pada instance. Anda dapat menjalankan semua jenis skrip selama didukung oleh sistem operasi yang berjalan pada instance. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan EC2/On-Premises.

Bagaimana CodeDeploy agen menggunakan AppSpec file

Selama penyebaran, CodeDeploy agen mencari nama acara saat ini di bagian kait file. AppSpec Jika acara tidak ditemukan, CodeDeploy agen melanjutkan ke langkah berikutnya. Jika acara ditemukan, CodeDeploy agen mengambil daftar skrip untuk dieksekusi. Skrip dijalankan secara berurutan, dalam

urutan di mana mereka muncul dalam file. Status setiap skrip dicatat dalam file log CodeDeploy agen pada instance.

Jika skrip berjalan dengan sukses, ia mengembalikan kode keluar 0 (nol).

> **Note**
>
> CodeDeploy Agen tidak digunakan dalam penyebaran AWS Lambda atau Amazon ECS.

Selama acara Install, CodeDeploy agen menggunakan pemetaan yang ditentukan di bagian file file untuk menentukan folder atau AppSpec file mana yang akan disalin dari revisi ke instance.

Jika CodeDeploy agen yang diinstal pada sistem operasi tidak cocok dengan apa yang tercantum dalam AppSpec file, penerapan gagal.

Untuk informasi tentang file log CodeDeploy agen, lihatBekerja dengan CodeDeploy agen.

Memulai dengan CodeDeploy

Topik

- Langkah 1: Menyiapkan
- Langkah 2: Buat peran layanan untuk CodeDeploy
- Langkah 3: Batasi izin CodeDeploy pengguna
- Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda

Langkah 1: Menyiapkan

Sebelum Anda menggunakan AWS CodeDeploy untuk pertama kalinya, Anda harus menyelesaikan langkah-langkah pengaturan. Langkah-langkahnya melibatkan pembuatan AWS akun (jika Anda belum memilikinya), dan pengguna administratif.

Dalam panduan ini, pengguna administratif disebut pengguna CodeDeploy administratif.

Mendaftar untuk Akun AWS

Untuk memulai AWS, Anda membutuhkan Akun AWS. Ada dua cara untuk mendaftar AWS. Jika Anda ingin kontrol penuh atas konfigurasi akun dan di mana konten Anda disimpan, atau jika Anda memiliki beban kerja yang diatur seperti HIPAA atau FedRAMP, gunakan Daftar untuk (lanjutan). AWS Jika Anda adalah pengguna baru dan ingin menggunakan login yang sudah Anda miliki untuk masuk AWS dan membuat AWS lingkungan dengan default yang telah dikonfigurasi sebelumnya di pilih Wilayah AWS, gunakan Daftar untuk. AWS

> **Important**
>
> Kami sangat menyarankan Anda mengonfigurasi pengguna CodeDeploy adminstratif sebagai identitas tenaga kerja (pengguna yang dikelola di Pusat Identitas IAM) dengan. AWS CLI Banyak prosedur dalam panduan ini menganggap Anda menggunakan AWS CLI untuk melakukan konfigurasi.

> **Important**
>
> Jika Anda mengonfigurasi AWS CLI, Anda mungkin diminta untuk menentukan AWS Wilayah. Pilih salah satu Wilayah yang didukung yang tercantum di Wilayah dan titik akhir di. Referensi Umum AWS

Langkah 2: Buat peran layanan untuk CodeDeploy

Di AWS, peran layanan digunakan untuk memberikan izin ke AWS layanan sehingga dapat mengakses AWS sumber daya. Kebijakan yang Anda lampirkan ke peran layanan menentukan sumber daya mana layanan dapat mengakses dan yang dapat dilakukan dengan sumber daya tersebut.

Peran layanan yang Anda buat CodeDeploy harus diberikan izin yang diperlukan untuk platform komputasi Anda. Jika Anda menerapkan ke lebih dari satu platform komputasi, buat satu peran layanan untuk masing-masing. Untuk menambahkan izin, lampirkan satu atau beberapa kebijakan yang AWS disediakan berikut:

Untuk EC2/On-Premises penerapan, lampirkan kebijakan. **`AWSCodeDeployRole`** Ini memberikan izin untuk peran layanan Anda untuk:

- Baca tag pada instans Anda atau identifikasi instans Amazon EC2 Anda dengan nama grup Amazon EC2 Auto Scaling.
- Membaca, membuat, memperbarui, dan menghapus grup Amazon EC2 Auto Scaling, kait siklus hidup, dan kebijakan penskalaan.
- Publikasikan informasi ke topik Amazon SNS.
- Ambil informasi tentang CloudWatch alarm.
- Baca dan perbarui Elastic Load Balancing.

> **Note**
>
> Jika Anda membuat grup Auto Scaling dengan template peluncuran, Anda harus menambahkan izin berikut:
>
> - `ec2:RunInstances`
> - `ec2:CreateTags`
> - `iam:PassRole`

Untuk informasi selengkapnya, lihatLangkah 2: Buat peran layanan, Membuat template peluncuran untuk grup Auto Scaling, dan Luncurkan dukungan template di Panduan Pengguna Amazon EC2 Auto Scaling.

Untuk penerapan Amazon ECS, jika Anda ingin akses penuh ke layanan dukungan, lampirkan kebijakan. **`AWSCodeDeployRoleForECS`** Ini memberikan izin untuk peran layanan Anda untuk:

- Baca, perbarui, dan hapus set tugas Amazon ECS.
- Perbarui grup target Elastic Load Balancing, pendengar, dan aturan.
- Memanggil AWS Lambda fungsi.
- Akses file revisi di bucket Amazon S3.
- Ambil informasi tentang CloudWatch alarm.
- Publikasikan informasi ke topik Amazon SNS.

Untuk penerapan Amazon ECS, jika Anda ingin akses terbatas ke layanan dukungan, lampirkan kebijakan. **`AWSCodeDeployRoleForECSLimited`** Ini memberikan izin untuk peran layanan Anda untuk:

- Baca, perbarui, dan hapus set tugas Amazon ECS.
- Ambil informasi tentang CloudWatch alarm.
- Publikasikan informasi ke topik Amazon SNS.

Untuk penerapan AWS Lambda, jika Anda ingin mengizinkan penerbitan ke Amazon SNS, lampirkan kebijakan. **`AWSCodeDeployRoleForLambda`** Ini memberikan izin untuk peran layanan Anda untuk:

- Baca, perbarui, dan panggil AWS Lambda fungsi dan alias.
- Akses file revisi di bucket Amazon S3.
- Ambil informasi tentang CloudWatch alarm.
- Publikasikan informasi ke topik Amazon SNS.

Untuk penerapan AWS Lambda, jika Anda ingin membatasi akses ke Amazon SNS, lampirkan kebijakan. **`AWSCodeDeployRoleForLambdaLimited`** Ini memberikan izin untuk peran layanan Anda untuk:

- Baca, perbarui, dan panggil AWS Lambda fungsi dan alias.
- Akses file revisi di bucket Amazon S3.
- Ambil informasi tentang CloudWatch alarm.

Sebagai bagian dari pengaturan peran layanan, Anda juga memperbarui hubungan kepercayaannya untuk menentukan titik akhir yang ingin Anda berikan aksesnya.

Anda dapat membuat peran layanan dengan konsol IAM, API IAM AWS CLI, atau IAM.

Topik

- Buat peran layanan (konsol)
- Buat peran layanan (CLI)
- Dapatkan peran layanan ARN (konsol)
- Dapatkan peran layanan ARN (CLI)

Buat peran layanan (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Pilih AWS layanan, dan di bawah Kasus penggunaan, dari daftar drop-down, pilih CodeDeploy.
4. Pilih kasus penggunaan Anda:
   - Untuk EC2/On-Premises penerapan, pilih. CodeDeploy
   - Untuk penerapan AWS Lambda, pilih Lambda. CodeDeploy
   - Untuk penerapan Amazon ECS, pilih CodeDeploy - ECS.
5. Pilih Berikutnya.
6. Pada halaman Tambahkan izin, kebijakan izin yang benar untuk kasus penggunaan ditampilkan. Pilih Berikutnya.
7. Pada halaman Nama, tinjau, dan buat, di Nama peran, masukkan nama untuk peran layanan (misalnya,**`CodeDeployServiceRole`**), lalu pilih Buat peran.

   Anda juga dapat memasukkan deskripsi untuk peran layanan ini di Deskripsi peran.

8. Jika Anda ingin peran layanan ini memiliki izin untuk mengakses semua titik akhir yang didukung saat ini, Anda selesai dengan prosedur ini.

   Untuk membatasi peran layanan ini dari akses ke beberapa titik akhir, lanjutkan dengan langkah-langkah yang tersisa dalam prosedur ini.

9. Dalam daftar peran, cari dan pilih peran yang baru saja Anda buat (`CodeDeployServiceRole`).

10. Pilih tab Trust relationship.

11. Pilih Edit kebijakan kepercayaan.

    Anda akan melihat kebijakan berikut, yang menyediakan izin peran layanan untuk mengakses semua titik akhir yang didukung:

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

    Untuk memberikan akses peran layanan ke hanya beberapa titik akhir yang didukung, ganti isi kotak teks kebijakan kepercayaan dengan kebijakan berikut. Hapus baris untuk titik akhir yang ingin Anda cegah aksesnya, lalu pilih Perbarui kebijakan.

```
{
    "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"
        }
    ]
}
```

Untuk informasi selengkapnya tentang membuat peran layanan, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM.

Buat peran layanan (CLI)

1. Pada mesin pengembangan Anda, buat file teks bernama, misalnya,`CodeDeployDemo-Trust.json`. File ini digunakan untuk memungkinkan CodeDeploy untuk bekerja atas nama Anda.

   Lakukan salah satu tindakan berikut:

   - Untuk memberikan akses ke semua AWS Wilayah yang didukung, simpan konten berikut dalam file:

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

   - Untuk memberikan akses ke hanya beberapa Wilayah yang didukung, ketik konten berikut ke dalam file, dan hapus baris untuk Wilayah yang ingin Anda kecualikan aksesnya:

   ```
   {
       "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**
>
> Jangan gunakan koma setelah titik akhir terakhir dalam daftar.

2. Dari direktori yang sama, panggil create-role perintah untuk membuat peran layanan bernama **`CodeDeployServiceRole`** berdasarkan informasi dalam file teks yang baru saja Anda buat:

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

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

Dalam output perintah, catat nilai `Arn` entri di bawah `Role` objek. Anda membutuhkannya nanti saat membuat grup penerapan. Jika Anda lupa nilainya, ikuti instruksi di[Dapatkan peran layanan ARN (CLI)](#).

3. Kebijakan terkelola yang Anda gunakan bergantung pada platform komputasi.

   - Jika penerapan Anda ke platform EC2/On-Premises komputasi:

     Panggil attach-role-policy perintah untuk memberikan peran layanan bernama **`CodeDeployServiceRole`** izin berdasarkan kebijakan terkelola IAM yang dinamai. **`AWSCodeDeployRole`** Contoh:

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

   - Jika penerapan Anda ke platform komputasi AWS Lambda:

     Panggil attach-role-policy perintah untuk memberikan peran layanan bernama **`CodeDeployServiceRole`** izin berdasarkan kebijakan terkelola IAM bernama **`AWSCodeDeployRoleForLambda`** atau. **`AWSCodeDeployRoleForLambdaLimited`** Contoh:

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

   - Jika penerapan Anda ke platform komputasi Amazon ECS:

     Panggil attach-role-policy perintah untuk memberikan peran layanan bernama **`CodeDeployServiceRole`** izin berdasarkan kebijakan terkelola IAM bernama **`AWSCodeDeployRoleForECS`** atau. **`AWSCodeDeployRoleForECSLimited`** Contoh:

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

Untuk informasi selengkapnya tentang membuat peran layanan, lihat Membuat peran untuk AWS layanan di Panduan Pengguna IAM.

Dapatkan peran layanan ARN (konsol)

Untuk menggunakan konsol IAM untuk mendapatkan ARN dari peran layanan:

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Di kotak Filter, ketik**`CodeDeployServiceRole`**, lalu tekan Enter.
4. Pilih CodeDeployServiceRole.
5. Catat nilai bidang ARN Peran.

Dapatkan peran layanan ARN (CLI)

Untuk menggunakan AWS CLI untuk mendapatkan ARN dari peran layanan, panggil get-role perintah terhadap peran layanan bernama: **`CodeDeployServiceRole`**

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

Nilai yang dikembalikan adalah ARN dari peran layanan.

Langkah 3: Batasi izin CodeDeploy pengguna

Untuk alasan keamanan, kami menyarankan Anda membatasi izin pengguna administratif yang Anda buat hanya Langkah 1: Menyiapkan untuk yang diperlukan untuk membuat dan mengelola penerapan di. CodeDeploy

Gunakan serangkaian prosedur berikut untuk membatasi izin pengguna CodeDeploy administratif.

Sebelum Anda mulai

- Pastikan Anda telah membuat pengguna CodeDeploy administratif di IAM Identity Center mengikuti petunjuk diLangkah 1: Menyiapkan.

Untuk membuat set izin

Anda akan menetapkan izin ini disetel ke pengguna CodeDeploy administratif nanti.

1. Masuk ke Konsol Manajemen AWS dan buka AWS IAM Identity Center konsol di https://console.aws.amazon.com/singlesignon/.
2. Di panel navigasi, pilih Set izin, lalu pilih Buat set izin.
3. Pilih Set izin khusus.
4. Pilih Berikutnya.
5. Pilih kebijakan Inline.
6. Hapus kode sampel.
7. Tambahkan kode kebijakan berikut:

   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"
        }
    ]
}
```

Dalam kebijakan ini, ganti *arn:aws:iam::account-ID:role/CodeDeployServiceRole* dengan nilai ARN dari peran CodeDeploy layanan yang Anda buat. Langkah 2: Buat peran layanan untuk CodeDeploy Anda dapat menemukan nilai ARN di halaman detail peran layanan di konsol IAM.

Kebijakan sebelumnya memungkinkan Anda menerapkan aplikasi ke platform komputasi AWS Lambda, platform komputasi, dan platform komputasi EC2/On-Premises Amazon ECS.

Anda dapat menggunakan CloudFormation templat yang disediakan dalam dokumentasi ini untuk meluncurkan instans Amazon EC2 yang kompatibel dengannya. CodeDeploy Untuk menggunakan CloudFormation templat untuk membuat aplikasi, grup penyebaran, atau konfigurasi penerapan, Anda harus menyediakan akses ke CloudFormation—dan AWS layanan serta tindakan yang CloudFormation bergantung pada—dengan menambahkan `cloudformation:*` izin ke kebijakan izin pengguna CodeDeploy administratif, seperti ini:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:*"
            ],
            "Resource": "*"
        }
    ]
}
```

8. Pilih Berikutnya.
9. Dalam nama set Izin, masukkan:

```
CodeDeployUserPermissionSet
```

10. Pilih Berikutnya.
11. Pada halaman Tinjau dan buat, tinjau informasi dan pilih Buat.

Untuk menetapkan izin yang disetel ke pengguna CodeDeploy administratif

1. Di panel navigasi, pilih Akun AWS, lalu pilih kotak centang di Akun AWS samping tempat Anda masuk saat ini.
2. Pilih tombol Tetapkan pengguna atau grup.
3. Pilih tab Pengguna.
4. Pilih kotak centang di sebelah pengguna CodeDeploy administratif.
5. Pilih Berikutnya.
6. Pilih kotak centang di sebelah`CodeDeployUserPermissionSet`.
7. Pilih Berikutnya.
8. Tinjau informasi dan pilih Kirim.

   Anda sekarang telah menetapkan pengguna CodeDeploy administratif dan `CodeDeployUserPermissionSet` untuk Anda Akun AWS, mengikat mereka bersama-sama.

Untuk keluar dan masuk kembali sebagai pengguna CodeDeploy administratif

1. Sebelum Anda keluar, pastikan Anda memiliki URL portal AWS akses dan nama pengguna dan kata sandi satu kali untuk pengguna CodeDeploy admin.

> **Note**
>
> Jika Anda tidak memiliki informasi ini, buka halaman detail pengguna CodeDeploy adminstratif di IAM Identity Center, pilih Reset password, Hasilkan password satu kali [...] , dan Atur ulang kata sandi lagi untuk menampilkan informasi di layar.

2. Keluar dari AWS.
3. Rekatkan URL portal AWS akses ke bilah alamat browser Anda.
4. Masuk sebagai pengguna CodeDeploy adminstratif.

   Sebuah Akun AWSkotak muncul di layar.
5. Pilih Akun AWS, lalu pilih nama yang Anda tetapkan pengguna CodeDeploy adminstratif dan set izin. Akun AWS
6. Di sebelah`CodeDeployUserPermissionSet`, pilih Konsol manajemen.

   Konsol Manajemen AWS Muncul. Anda sekarang masuk sebagai pengguna CodeDeploy adminstratif dengan izin terbatas. Anda sekarang dapat melakukan CodeDeploy-related operasi, dan hanya CodeDeploy-related operasi, sebagai pengguna ini.

Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda

> **Note**
>
> Jika Anda menggunakan platform komputasi Amazon ECS atau AWS Lambda, lewati langkah ini.

Instans Amazon EC2 Anda memerlukan izin untuk mengakses bucket atau GitHub repositori Amazon S3 tempat aplikasi disimpan. Untuk meluncurkan instans Amazon EC2 yang kompatibel dengannya CodeDeploy, Anda harus membuat peran IAM tambahan, profil instans. Petunjuk ini menunjukkan cara membuat profil instans IAM untuk dilampirkan ke instans Amazon EC2 Anda. Peran ini memberikan izin CodeDeploy agen untuk mengakses bucket atau GitHub repositori Amazon S3 tempat aplikasi Anda disimpan.

Anda dapat membuat profil instans IAM dengan AWS CLI, konsol IAM, atau API IAM.

> **Note**
>
> Anda dapat melampirkan profil instans IAM ke instans Amazon EC2 saat meluncurkannya atau ke instans yang diluncurkan sebelumnya. Untuk informasi selengkapnya, lihat Profil instans.

Topik

- Buat profil instans IAM untuk instans Amazon EC2 (CLI)
- Membuat profil instans IAM untuk instans Amazon EC2 (konsol)

Buat profil instans IAM untuk instans Amazon EC2 (CLI)

Dalam langkah-langkah ini, kami berasumsi Anda telah mengikuti instruksi dalam tiga langkah pertamaMemulai dengan CodeDeploy.

1. Pada mesin pengembangan Anda, buat file teks bernama`CodeDeployDemo-EC2-Trust.json`. Tempel konten berikut, yang memungkinkan Amazon EC2 berfungsi atas nama Anda:

   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. Di direktori yang sama, buat file teks bernama`CodeDeployDemo-EC2-Permissions.json`. Tempel konten berikut:

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

> **Note**
>
> Kami menyarankan Anda membatasi kebijakan ini hanya untuk bucket Amazon S3 yang harus diakses instans Amazon EC2 Anda. Pastikan untuk memberikan akses ke bucket

Amazon S3 yang berisi agen. CodeDeploy Jika tidak, kesalahan dapat terjadi ketika CodeDeploy agen diinstal atau diperbarui pada instance. Untuk memberikan akses profil instans IAM hanya ke beberapa bucket kit CodeDeploy sumber daya di Amazon S3, gunakan kebijakan berikut, tetapi hapus baris untuk bucket yang ingin Anda cegah aksesnya:

```
{
  "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**
>
> Jika Anda ingin menggunakan otorisasi IAM atau titik akhir CodeDeploy Amazon Virtual Private Cloud (VPC), Anda perlu menambahkan lebih banyak izin. Lihat Menggunakan CodeDeploy dengan Amazon Virtual Private Cloud untuk informasi selengkapnya.

3. Dari direktori yang sama, panggil create-role perintah untuk membuat peran IAM bernama**`CodeDeployDemo-EC2-Instance-Profile`**, berdasarkan informasi di file pertama:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

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

4. Dari direktori yang sama, panggil put-role-policy perintah untuk memberikan peran bernama **`CodeDeployDemo-EC2-Instance-Profile`** izin berdasarkan informasi dalam file kedua:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

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

5. Hubungi attach-role-policy untuk memberikan peran izin Amazon EC2 Systems Manager sehingga SSM dapat menginstal agen. CodeDeploy Kebijakan ini tidak diperlukan jika Anda

berencana untuk menginstal agen dari bucket Amazon S3 publik dengan baris perintah. Pelajari lebih lanjut tentang menginstal CodeDeploy agen.

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

6. Panggil create-instance-profile perintah diikuti dengan add-role-to-instance-profile perintah untuk membuat profil instans IAM bernama**`CodeDeployDemo-EC2-Instance-Profile`**. Profil instans memungkinkan Amazon EC2 meneruskan peran IAM yang diberi nama **`CodeDeployDemo-EC2-Instance-Profile`** ke instans Amazon EC2 saat instans pertama kali diluncurkan:

   ```
   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
   ```

   Jika Anda perlu mendapatkan nama profil instans IAM, lihat list-instance-profiles-for-role di bagian IAM Referensi.AWS CLI

Anda sekarang telah membuat profil instans IAM untuk dilampirkan ke instans Amazon EC2 Anda. Untuk informasi selengkapnya, lihat peran IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2.

Membuat profil instans IAM untuk instans Amazon EC2 (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Dalam kolom IAM, dalam panel navigas, pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pada halaman Tentukan izin, pilih JSON.
4. Hapus `JSON` kode contoh.
5. Tempel kode berikut:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
   ```

```
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Note

Kami menyarankan Anda membatasi kebijakan ini hanya untuk bucket Amazon S3 yang harus diakses instans Amazon EC2 Anda. Pastikan untuk memberikan akses ke bucket Amazon S3 yang berisi agen. CodeDeploy Jika tidak, kesalahan dapat terjadi ketika CodeDeploy agen diinstal atau diperbarui pada instance. Untuk memberikan akses profil instans IAM hanya ke beberapa bucket kit CodeDeploy sumber daya di Amazon S3, gunakan kebijakan berikut, tetapi hapus baris untuk bucket yang ingin Anda cegah aksesnya:

```
{
  "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**
>
> Jika Anda ingin menggunakan otorisasi IAM atau titik akhir CodeDeploy Amazon Virtual Private Cloud (VPC), Anda perlu menambahkan lebih banyak izin. Lihat Menggunakan CodeDeploy dengan Amazon Virtual Private Cloud untuk informasi selengkapnya.

6. Pilih Berikutnya.
7. Pada halaman Tinjau dan buat, di kotak Nama kebijakan, ketik**`CodeDeployDemo-EC2-Permissions`**.
8. (Opsional) Untuk Deskripsi, ketik deskripsi untuk kebijakan.
9. Pilih Buat kebijakan.
10. Di panel navigasi, pilih Peran, lalu pilih Buat peran.
11. Di bawah Kasus penggunaan, pilih kasus penggunaan EC2.
12. Pilih Berikutnya.
13. Dalam daftar kebijakan, pilih kotak centang di samping kebijakan yang baru saja Anda buat (CodeDeployDemo-EC2-Permissions). Jika perlu, gunakan kotak pencarian untuk menemukan kebijakan.

14. Untuk menggunakan Systems Manager untuk menginstal atau mengkonfigurasi CodeDeploy agen, pilih kotak centang di sebelahnya AmazonSSMManagedInstanceCore. Kebijakan AWS terkelola ini memungkinkan instance untuk menggunakan fungsionalitas inti layanan Systems Manager. Jika perlu, gunakan kotak pencarian untuk menemukan kebijakan. Kebijakan ini tidak diperlukan jika Anda berencana untuk menginstal agen dari bucket Amazon S3 publik dengan baris perintah. Pelajari lebih lanjut tentang menginstal CodeDeploy agen.
15. Pilih Berikutnya.
16. Pada halaman Nama, tinjau, dan buat, di Nama peran, masukkan nama untuk peran layanan (misalnya,**`CodeDeployDemo-EC2-Instance-Profile`**), lalu pilih Buat peran.

    Anda juga dapat memasukkan deskripsi untuk peran layanan ini di Deskripsi peran.

Anda sekarang telah membuat profil instans IAM untuk dilampirkan ke instans Amazon EC2 Anda. Untuk informasi selengkapnya, lihat peran IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2.

Integrasi produk dan layanan dengan CodeDeploy

Secara default, CodeDeploy terintegrasi dengan sejumlah AWS layanan dan produk dan layanan mitra. Informasi berikut dapat membantu Anda mengonfigurasi CodeDeploy untuk berintegrasi dengan produk dan layanan yang Anda gunakan.

- Integrasi dengan yang lain AWS layanan
- Integrasi dengan produk dan layanan mitra
- Contoh integrasi dari komunitas

Integrasi dengan yang lain AWS layanan

CodeDeploy terintegrasi dengan AWS layanan berikut:

Amazon CloudWatch	Amazon CloudWatch adalah layanan pemantauan untuk sumber daya AWS cloud dan aplikasi yang Anda jalankan AWS. Anda dapat menggunakan Amazon CloudWatch untuk mengumpulkan dan melacak metrik, mengumpulkan dan memantau file log, dan mengatur alarm. CodeDeploy mendukung CloudWatch alat-alat berikut: • CloudWatch alarm untuk memantau penerapan Anda dan menghentikannya ketika metrik pemantauan yang ditentuka n melebihi atau jatuh di bawah ambang batas yang Anda tentukan dalam aturan alarm. CloudWatch Untuk menggunak an pemantauan alarm, pertama-tama Anda mengatur alarm CloudWatch, lalu menambahkannya CodeDeploy ke aplikasi atau grup penyebaran tempat penerapan harus berhenti saat alarm diaktifkan. Pelajari lebih lanjut:

- Membuat CloudWatch alarm Log

- Amazon CloudWatch Events untuk mendeteksi dan bereaksi terhadap perubahan dalam status instans atau penerapan dalam operasi Anda. CodeDeploy Kemudian, berdasarkan aturan yang Anda buat, CloudWatch Acara memanggil satu atau beberapa tindakan target saat penerapan atau instance memasuki status yang Anda tentukan dalam aturan.

  Pelajari lebih lanjut:

  - Memantau penerapan dengan Amazon Events CloudWatch

- Amazon CloudWatch Log untuk memantau tiga jenis log yang dibuat oleh CodeDeploy agen tanpa harus masuk ke instance satu per satu.

  Pelajari lebih lanjut:

  - Lihat CodeDeploy log di konsol CloudWatch Log

Amazon EC2 Auto Scaling	CodeDeploy mendukung Amazon EC2 Auto Scaling. AWS Layanan ini dapat secara otomatis meluncurkan instans Amazon EC2 berdasarkan kriteria yang Anda tentukan, misalnya: • Batas terlampaui untuk pemanfaatan CPU tertentu. • Disk membaca atau menulis. • Lalu lintas jaringan masuk atau keluar selama interval waktu tertentu. Anda dapat meningkatkan skala grup instans Amazon EC2 kapan pun Anda membutuhk annya dan kemudian menggunakannya CodeDeploy untuk menerapkan revisi aplikasi ke instans tersebut secara otomatis. Amazon EC2 Auto Scaling menghentikan instans Amazon EC2 tersebut saat tidak lagi diperluka n. Pelajari lebih lanjut: • Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling • Di bawah kap: CodeDeploy dan integrasi Auto Scaling

Layanan Kontainer Elastis Amazon	Anda dapat menggunakan CodeDeploy untuk menerapkan aplikasi kontainer Amazon ECS sebagai set tugas. CodeDeploy melakukan blue/green penyebaran dengan menginstal versi aplikasi yang diperbarui sebagai set tugas pengganti baru. CodeDeploy mengubah rute lalu lintas produksi dari tugas aplikasi asli yang disetel ke set tugas pengganti. Set tugas asli dihentikan setelah penerapan berhasil. Untuk informasi selengkapnya tentang Amazon ECS, lihat Amazon Elastic Container Service. Anda dapat mengelola cara lalu lintas dialihkan ke set tugas yang diperbarui selama penerapan dengan memilih konfigurasi canary, linier, atau all-at-once. Untuk informasi selengkap nya tentang penerapan Amazon ECS, lihat Penerapan di platform komputasi Amazon ECS.

AWS CloudTrail	CodeDeploy terintegrasi dengan AWS CloudTrail. Layanan ini menangkap panggilan API yang dilakukan oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. CodeDeploy CloudTrailmenangkap panggilan API dari CodeDeploy konsol, dari CodeDeploy perintah melalui AWS CLI, atau dari CodeDeploy API secara langsung. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan: • Permintaan mana yang dibuat untuk CodeDeploy. • Alamat IP sumber dari mana permintaan dibuat. • Siapa yang membuat permintaan. • Ketika itu dibuat. Pelajari lebih lanjut: • Monitoring Deployments

AWS Cloud9	AWS Cloud9adalah lingkungan pengemban gan terintegrasi (IDE) online berbasis cloud yang dapat Anda gunakan untuk menulis, menjalankan, men-debug, dan menyebark an kode hanya menggunakan browser dari mesin yang terhubung ke internet. AWS Cloud9 termasuk editor kode, debugger, terminal, dan alat-alat penting, seperti dan AWS CLI Git. • Anda dapat menggunakan AWS Cloud9 IDE untuk menjalankan, men-debug, dan membangun kode yang ada di GitHub repositori. Anda dapat melihat, mengubah, dan menyimpan kode menggunakan jendela Lingkungan IDE dan tab editor. Ketika Anda siap, Anda dapat menggunakan Git di sesi AWS Cloud9 terminal untuk mendorong perubahan kode ke GitHub repositori Anda, dan kemudian gunakan AWS CodeDeploy untuk menyebarkan pembaruan Anda. Untuk informasi selengkapnya tentang menggunak an AWS Cloud9 with GitHub, lihat GitHub contoh untuk AWS Cloud9. • Anda dapat menggunakan AWS Cloud9 IDE untuk memperbarui AWS Lambda fungsi. Anda kemudian dapat menggunakan AWS CodeDeploy untuk membuat penyebaran yang menggeser lalu lintas ke versi baru fungsi Anda AWS Lambda . Untuk informasi selengkapnya, lihat Bekerja dengan AWS Lambda fungsi di Lingkungan Pengemban gan AWS Cloud9 Terpadu (IDE). Untuk informasi selengkapnya AWS Cloud9, lihat Apa itu AWS Cloud9 dan Memulai AWS Cloud9.

AWS CodePipeline	AWS CodePipelineadalah layanan pengirima n berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perangkat lunak Anda dalam proses pengiriman berkelanjutan. Anda dapat menggunakan AWS CodePipeline untuk menentukan proses rilis Anda sendiri sehingga layanan membangun, menguji, dan menyebark an kode Anda setiap kali ada perubahan kode. Misalnya, Anda mungkin memiliki tiga grup penerapan untuk aplikasi: Beta, Gamma, dan Prod. Anda dapat mengatur pipeline sehingga setiap kali ada perubahan dalam kode sumber Anda, pembaruan akan diterapkan ke setiap grup penyebaran, satu per satu. Anda dapat mengonfigurasi AWS CodePipeline untuk digunakan CodeDeploy untuk menyebark an: • Kode ke instans Amazon EC2, instans lokal, atau keduanya. • Versi fungsi AWS Lambda tanpa server. Anda dapat membuat grup CodeDeploy aplikasi, deployment, dan deployment untuk digunakan dalam tindakan deploy dalam satu tahap baik sebelum membuat pipeline atau di wizard Create Pipeline. Pelajari lebih lanjut: • AWS panduan untuk DevOps memulai — Pelajari cara menggunakannya CodePipeline CodeDeploy untuk terus mengirimkan dan

	menerapkan kode sumber di CodeCommit repositori ke instans Amazon EC2. • Panduan pipa sederhana (Amazon S3 Bucket) • Panduan pipa sederhana (Repositori) CodeCommit • Four-stagetutorial pipa
AWS Model Aplikasi Tanpa Server	AWS Serverless Application Model (AWS SAM) adalah model untuk mendefinisikan aplikasi tanpa server. Ini meluas CloudFormation untuk menyediakan cara sederhana untuk mendefini sikan AWS Lambda fungsi, API Amazon API Gateway, dan tabel Amazon DynamoDB yang diperlukan oleh aplikasi tanpa server. Jika Anda sudah menggunakan AWS SAM, Anda dapat menambahkan preferensi penerapan untuk mulai menggunakan CodeDeploy untuk mengelola cara lalu lintas digeser selama penerapan aplikasi AWS Lambda. Untuk informasi selengkapnya, lihat Model Aplikasi AWS Tanpa Server.

Elastic Load Balancing	CodeDeploy mendukung Elastic Load Balancing, layanan yang mendistribusikan lalu lintas aplikasi masuk di beberapa instans Amazon EC2. Untuk CodeDeploy penerapan, penyeimba ng beban juga mencegah lalu lintas dialihkan ke instance ketika mereka tidak siap, saat ini sedang digunakan, atau tidak lagi diperlukan sebagai bagian dari lingkungan. Pelajari lebih lanjut: • Integrating CodeDeploy with Elastic Load Balancing

Topik

- Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling
- Integrasi CodeDeploy dengan Elastic Load Balancing

Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling

CodeDeploy mendukung Amazon EC2 Auto Scaling AWS , layanan yang meluncurkan instans Amazon EC2 secara otomatis sesuai dengan kondisi yang Anda tentukan. Kondisi ini dapat mencakup batas yang terlampaui dalam interval waktu tertentu untuk pemanfaatan CPU, membaca atau menulis disk, atau lalu lintas jaringan masuk atau keluar. Amazon EC2 Auto Scaling menghentikan instans saat tidak lagi diperlukan. Untuk informasi selengkapnya, lihat Apa itu Amazon EC2 Auto Scaling? di Panduan Pengguna Penskalaan Otomatis Amazon EC2.

Saat instans Amazon EC2 baru diluncurkan sebagai bagian dari CodeDeploy grup Penskalaan Otomatis Amazon EC2, Anda dapat menerapkan revisi Anda ke instans baru secara otomatis. Anda juga dapat mengoordinasikan penerapan CodeDeploy dengan instans Amazon EC2 Auto Scaling yang terdaftar dengan penyeimbang beban Elastic Load Balancing. Untuk informasi selengkapnya, lihat Integrating CodeDeploy with Elastic Load Balancing dan Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2.

> **Note**
>
> Anda mungkin mengalami masalah jika mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2. Jika satu penerapan gagal, misalnya, instance akan mulai dimatikan, tetapi penerapan lain yang sedang berjalan dapat memakan waktu satu jam untuk habis waktu. Untuk informasi selengkapnya, lihat Hindari mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2 dan Under the hood: CodeDeploy dan integrasi Amazon EC2 Auto Scaling.

Topik

- Menyebarkan CodeDeploy aplikasi ke grup Amazon EC2 Auto Scaling
- Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling
- Cara Kerja Amazon EC2 Auto Scaling CodeDeploy
- Menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Scaling

Menyebarkan CodeDeploy aplikasi ke grup Amazon EC2 Auto Scaling

Untuk menerapkan revisi CodeDeploy aplikasi ke grup Amazon EC2 Auto Scaling:

1. Buat atau temukan profil instans IAM yang memungkinkan grup Penskalaan Otomatis Amazon EC2 bekerja dengan Amazon S3. Untuk informasi selengkapnya, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.

   > **Note**
   >
   > Anda juga dapat menggunakan CodeDeploy untuk menerapkan revisi dari GitHub repositori ke grup Amazon EC2 Auto Scaling. Meskipun instans Amazon EC2 masih memerlukan profil instans IAM, profil tersebut tidak memerlukan izin tambahan untuk menerapkan dari repositori. GitHub

2. Buat atau gunakan grup Amazon EC2 Auto Scaling, yang menentukan profil instans IAM dalam konfigurasi atau templat peluncuran Anda. Untuk informasi selengkapnya, lihat peran IAM untuk aplikasi yang berjalan di instans Amazon EC2.
3. Membuat atau menemukan peran layanan yang memungkinkan CodeDeploy untuk membuat grup penerapan yang berisi grup Penskalaan Otomatis Amazon EC2.

4. Buat grup penerapan dengan CodeDeploy, tentukan nama grup Amazon EC2 Auto Scaling, peran layanan, dan beberapa opsi lainnya. Untuk informasi selengkapnya, lihat Buat grup penerapan untuk penerapan di tempat (konsol) atau Buat grup penerapan untuk penerapan di tempat (konsol).
5. Gunakan CodeDeploy untuk menerapkan revisi Anda ke grup penerapan yang berisi grup Penskalaan Otomatis Amazon EC2.

Untuk informasi selengkapnya, lihat Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling.

Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling

Penerapan terminasi adalah jenis CodeDeploy penerapan yang diaktifkan secara otomatis saat terjadi peristiwa penskalaan Auto Scaling. CodeDeploy melakukan penerapan penghentian tepat sebelum layanan Auto Scaling menghentikan instance. Selama penerapan penghentian, CodeDeploy tidak menerapkan apa pun. Sebagai gantinya, ini menghasilkan peristiwa siklus hidup, yang dapat Anda hubungkan ke skrip Anda sendiri untuk mengaktifkan fungsionalitas shutdown khusus. Misalnya, Anda dapat menghubungkan peristiwa `ApplicationStop` siklus hidup ke skrip yang mematikan aplikasi Anda dengan anggun sebelum instance dihentikan.

Untuk daftar peristiwa siklus hidup yang CodeDeploy dihasilkan selama penerapan penghentian, lihat. Ketersediaan kait acara siklus hidup

Jika penerapan penghentian gagal karena alasan apa pun, CodeDeploy akan memungkinkan penghentian instance untuk dilanjutkan. Ini berarti bahwa instance akan dimatikan meskipun CodeDeploy tidak menjalankan set lengkap (atau apa pun) peristiwa siklus hidup hingga selesai.

Jika Anda tidak mengaktifkan penerapan penghentian, layanan Auto Scaling akan tetap menghentikan instans Amazon EC2 saat peristiwa scale-in terjadi, tetapi tidak akan menghasilkan peristiwa siklus hidup. CodeDeploy

> **Note**
>
> Terlepas dari apakah Anda mengaktifkan penerapan penghentian atau tidak, jika layanan Auto Scaling menghentikan instans Amazon EC2 CodeDeploy saat penerapan sedang berlangsung, maka kondisi balapan dapat terjadi antara peristiwa siklus hidup yang dihasilkan oleh Auto Scaling dan layanan. CodeDeploy Misalnya, peristiwa `Terminating` siklus hidup (yang dihasilkan oleh layanan Auto Scaling) mungkin mengganti

`ApplicationStart` peristiwa (dihasilkan oleh penerapan). CodeDeploy Dalam skenario ini, Anda mungkin mengalami kegagalan dengan penghentian instans Amazon EC2 atau penerapan CodeDeploy .

Untuk mengaktifkan CodeDeploy untuk melakukan penerapan penghentian

- Pilih kotak centang Tambahkan pengakhiran ke grup Auto Scaling saat membuat atau memperbarui grup penerapan Anda. Untuk instruksi, lihatBuat grup penerapan untuk penerapan di tempat (konsol), atauMembuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol).

  Mengaktifkan kotak centang ini CodeDeploy menyebabkan pemasangan hook siklus hidup Auto Scaling ke grup Auto Scaling yang Anda tentukan saat membuat atau memperbarui grup penerapan. CodeDeploy Hook ini disebut hook terminasi dan memungkinkan penerapan terminasi.

Setelah kait terminasi diinstal, peristiwa scale-in (terminasi) terungkap sebagai berikut:

1. Layanan Auto Scaling (atau secara sederhana, Auto Scaling) menentukan bahwa peristiwa scale-in perlu terjadi, dan menghubungi layanan EC2 untuk menghentikan instans EC2.
2. Layanan EC2 mulai menghentikan instans EC2. Contoh bergerak ke `Terminating` negara bagian, dan kemudian ke `Terminating:Wait` negara bagian.
3. Selama`Terminating:Wait`, Auto Scaling menjalankan semua kait siklus hidup yang terpasang pada grup Auto Scaling, termasuk hook terminasi yang dipasang oleh. CodeDeploy
4. Hook terminasi mengirimkan pemberitahuan ke antrian Amazon SQS yang disurvei oleh. CodeDeploy
5. Setelah menerima pemberitahuan, CodeDeploy mem-parsing pesan, melakukan beberapa validasi, dan melakukan penyebaran penghentian.
6. Saat penerapan penghentian berjalan, CodeDeploy kirim detak jantung setiap lima menit ke Auto Scaling untuk memberi tahu bahwa instance masih dikerjakan.
7. Sejauh ini, instans EC2 masih dalam `Terminating:Wait` status (atau mungkin `Warmed:Pending:Wait` status, jika Anda telah mengaktifkan kumpulan hangat grup Auto Scaling).
8. Ketika penerapan selesai, CodeDeploy menunjukkan Auto Scaling `CONTINUE` ke proses penghentian EC2, terlepas dari apakah penerapan penghentian berhasil atau gagal.

Cara Kerja Amazon EC2 Auto Scaling CodeDeploy

Saat Anda membuat atau memperbarui grup CodeDeploy penerapan untuk menyertakan grup Auto Scaling CodeDeploy, mengakses grup Auto Scaling menggunakan peran layanan, lalu menginstal hook siklus hidup Auto Scaling CodeDeploy ke grup Auto Scaling Anda.

> **Note**
>
> Kait siklus hidup Auto Scaling berbeda dari peristiwa siklus hidup (juga disebut kait peristiwa siklus hidup) yang dihasilkan oleh dan dijelaskan dalam panduan ini. CodeDeploy AppSpec Bagian 'kait'

Kait siklus hidup Auto Scaling yang dipasang adalah: CodeDeploy

- Hook peluncuran - Kait ini memberi tahu CodeDeploy bahwa acara penskalaan Auto Scaling sedang berlangsung, dan CodeDeploy itu perlu memulai penerapan peluncuran.

  Selama penyebaran peluncuran, CodeDeploy:

  - Menerapkan revisi aplikasi Anda ke instance scaled-out.
  - Menghasilkan peristiwa siklus hidup untuk menunjukkan kemajuan penerapan. Anda dapat menghubungkan peristiwa siklus hidup ini ke skrip Anda sendiri untuk mengaktifkan fungsionalitas startup kustom. Untuk informasi lebih lanjut, lihat tabel diKetersediaan kait acara siklus hidup.

  Kait peluncuran dan penerapan peluncuran terkait selalu diaktifkan dan tidak dapat dimatikan.
- Hook terminasi — Hook opsional ini memberi tahu CodeDeploy bahwa peristiwa penskalaan Auto Scaling sedang berlangsung, dan CodeDeploy itu perlu memulai penerapan penghentian.

  Selama penerapan penghentian, buat CodeDeploy peristiwa siklus hidup untuk menunjukkan kemajuan shutdown instance. Untuk informasi selengkapnya, lihat Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling.

Topik

- Setelah CodeDeploy menginstal kait siklus hidup, bagaimana mereka digunakan?
- Bagaimana CodeDeploy nama grup Amazon EC2 Auto Scaling
- Urutan eksekusi peristiwa kait siklus hidup kustom

- Scale-out peristiwa selama penyebaran
- Scale-in peristiwa selama penyebaran
- Urutan acara di AWS CloudFormation skrip cfn-init

Setelah CodeDeploy menginstal kait siklus hidup, bagaimana mereka digunakan?

Setelah kait siklus hidup peluncuran dan penghentian diinstal, masing-masing digunakan oleh selama acara penskalaan dan CodeDeploy penskalaan grup Auto Scaling.

Acara scale-out (peluncuran) terungkap sebagai berikut:

1. Layanan Auto Scaling (atau secara sederhana, Auto Scaling) menentukan bahwa peristiwa scale-out perlu terjadi, dan menghubungi layanan EC2 untuk meluncurkan instans EC2 baru.
2. Layanan EC2 meluncurkan instans EC2 baru. Contoh bergerak ke `Pending` negara bagian, dan kemudian ke `Pending:Wait` negara bagian.
3. Selama`Pending:Wait`, Auto Scaling menjalankan semua kait siklus hidup yang terpasang pada grup Auto Scaling, termasuk hook peluncuran yang dipasang oleh. CodeDeploy
4. Kait peluncuran mengirimkan pemberitahuan ke antrian Amazon SQS yang disurvei oleh. CodeDeploy
5. Setelah menerima pemberitahuan, CodeDeploy mem-parsing pesan, melakukan beberapa validasi, dan memulai penyebaran peluncuran.
6. Saat penerapan peluncuran sedang berjalan, CodeDeploy kirim detak jantung setiap lima menit ke Auto Scaling untuk memberi tahu bahwa instance masih dikerjakan.
7. Sejauh ini, instans EC2 masih dalam `Pending:Wait` keadaan.
8. Ketika penerapan selesai, CodeDeploy menunjukkan Auto Scaling ke `CONTINUE` salah satu `ABANDON` atau proses peluncuran EC2, tergantung pada apakah penerapan berhasil atau gagal.
   - Jika CodeDeploy menunjukkan`CONTINUE`, Auto Scaling melanjutkan proses peluncuran, baik menunggu kait lain selesai, atau memasukkan instance ke dalam `Pending:Proceed` dan kemudian status. `InService`
   - Jika CodeDeploy menunjukkan**`ABANDON`**, Auto Scaling menghentikan instans EC2, dan memulai ulang prosedur peluncuran jika diperlukan untuk memenuhi jumlah instans yang diinginkan, seperti yang didefinisikan dalam pengaturan Kapasitas yang Diinginkan Auto Scaling.

Peristiwa scale-in (terminasi) terungkap sebagai berikut:

Lihat Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling.

Bagaimana CodeDeploy nama grup Amazon EC2 Auto Scaling

Selama blue/green penerapan pada platform EC2/On-Premises komputasi, Anda memiliki dua opsi untuk menambahkan instance ke lingkungan pengganti (hijau):

- Gunakan contoh yang sudah ada atau yang Anda buat secara manual.
- Gunakan pengaturan dari grup Penskalaan Otomatis Amazon EC2 yang Anda tentukan untuk menentukan dan membuat instance di grup Amazon EC2 Auto Scaling baru.

Jika Anda memilih opsi kedua, berikan grup CodeDeploy Penskalaan Otomatis Amazon EC2 baru untuk Anda. Ini menggunakan konvensi berikut untuk memberi nama grup:

```
CodeDeploy_deployment_group_name_deployment_id
```

Misalnya, jika penerapan dengan ID `10` menyebarkan grup penerapan bernama`alpha-deployments`, grup Penskalaan Otomatis Amazon EC2 yang disediakan akan diberi nama. `CodeDeploy_alpha-deployments_10` Untuk informasi selengkapnya, lihat Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol) dan GreenFleetProvisioningOption.

Urutan eksekusi peristiwa kait siklus hidup kustom

Anda dapat menambahkan kait siklus hidup Anda sendiri ke grup Amazon EC2 Auto Scaling yang digunakan. CodeDeploy Namun, urutan di mana peristiwa hook siklus hidup kustom tersebut dijalankan tidak dapat ditentukan sebelumnya sehubungan dengan peristiwa siklus hidup penerapan CodeDeploy default. Misalnya, jika Anda menambahkan hook siklus hidup khusus yang diberi nama `ReadyForSoftwareInstall` ke grup Penskalaan Otomatis Amazon EC2, Anda tidak dapat mengetahui sebelumnya apakah pengait tersebut akan dijalankan sebelum peristiwa siklus hidup penerapan default pertama, atau setelah peristiwa siklus hidup penerapan default terakhir. CodeDeploy

Untuk mempelajari cara menambahkan kait siklus hidup khusus ke grup Penskalaan Otomatis Amazon EC2, lihat Menambahkan kait siklus hidup di Panduan Pengguna Amazon EC2 Auto Scaling.

Scale-out peristiwa selama penyebaran

Jika peristiwa penskalaan Auto Scaling terjadi saat penerapan sedang berlangsung, instance baru akan diperbarui dengan revisi aplikasi yang sebelumnya digunakan, bukan revisi aplikasi terbaru. Jika penerapan berhasil, instance lama dan instance yang baru diskalakan akan menjadi hosting revisi aplikasi yang berbeda. Untuk memperbarui instance dengan revisi lama, CodeDeploy secara otomatis memulai penerapan tindak lanjut (segera setelah yang pertama) untuk memperbarui instans yang sudah ketinggalan zaman. Jika Anda ingin mengubah perilaku default ini sehingga instance EC2 yang sudah ketinggalan zaman dibiarkan di revisi yang lebih lama, lihat. Automatic updates to outdated instances

Jika Anda ingin menangguhkan proses penskalaan otomatis Amazon EC2 Auto Scaling saat penerapan sedang berlangsung, Anda dapat melakukannya melalui pengaturan dalam `common_functions.sh` skrip yang digunakan untuk penyeimbangan beban dengan. CodeDeploy Jika`HANDLE_PROCS=true`, peristiwa Auto Scaling berikut ditangguhkan secara otomatis selama proses penerapan:

- AzreBalance
- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

> **Important**
>
> Hanya konfigurasi CodeDeployDefault.OneAtATime penerapan yang mendukung fungsi ini.

Untuk informasi selengkapnya tentang penggunaan `HANDLE_PROCS=true` untuk menghindari masalah penerapan saat menggunakan Amazon EC2 Auto Scaling, lihat Pemberitahuan penting tentang AutoScaling penanganan proses di aws-codedeploy-samples pada. GitHub

Scale-in peristiwa selama penyebaran

Jika grup Auto Scaling mulai melakukan penskalaan saat CodeDeploy penerapan sedang berlangsung pada grup Auto Scaling tersebut, kondisi balapan dapat terjadi antara proses penghentian (termasuk peristiwa siklus hidup penerapan penghentian) dan peristiwa siklus hidup lainnya pada instance CodeDeploy penghentian. CodeDeploy Penerapan pada instance tertentu mungkin gagal jika instance dihentikan sebelum semua peristiwa CodeDeploy siklus hidup selesai.

Selain itu, CodeDeploy penerapan keseluruhan mungkin gagal atau tidak, tergantung pada cara Anda menyetel setelan host sehat Minimum dalam konfigurasi penerapan Anda.

Urutan acara di AWS CloudFormation skrip cfn-init

Jika Anda menggunakan `cfn-init` (or`cloud-init`) untuk menjalankan skrip pada Linux-based instance yang baru disediakan, penerapan Anda mungkin gagal kecuali jika Anda benar-benar mengontrol urutan peristiwa yang terjadi setelah instance dimulai.

Perintah itu harus:

1. Instance yang baru disediakan dimulai.
2. Semua skrip `cfn-init` bootstrap berjalan hingga selesai.
3. CodeDeploy Agen mulai.
4. Revisi aplikasi terbaru diterapkan ke instance.

Jika urutan peristiwa tidak dikontrol dengan hati-hati, CodeDeploy agen mungkin memulai penerapan sebelum semua skrip selesai berjalan.

Untuk mengontrol urutan peristiwa, gunakan salah satu praktik terbaik ini:

- Instal CodeDeploy agen melalui `cfn-init` skrip, letakkan setelah semua skrip lainnya.
- Sertakan CodeDeploy agen dalam AMI khusus dan gunakan `cfn-init` skrip untuk memulainya, menempatkannya setelah semua skrip lainnya.

Untuk informasi tentang penggunaan`cfn-init`, lihat cfn-init di AWS CloudFormation Panduan Pengguna.

Menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Scaling

Anda memiliki dua opsi untuk menentukan AMI dasar yang akan digunakan saat instans Amazon EC2 baru diluncurkan di grup Penskalaan Otomatis Amazon EC2:

- Anda dapat menentukan AMI kustom dasar yang sudah menginstal CodeDeploy agen. Karena agen sudah diinstal, opsi ini meluncurkan instans Amazon EC2 baru lebih cepat daripada opsi lainnya. Namun, opsi ini memberikan kemungkinan lebih besar bahwa penerapan awal instans Amazon EC2 akan gagal, terutama jika agen CodeDeploy kedaluwarsa. Jika Anda memilih opsi ini, kami sarankan Anda memperbarui CodeDeploy agen secara teratur di AMI kustom dasar Anda.

- Anda dapat menentukan AMI dasar yang tidak menginstal CodeDeploy agen dan menginstal agen karena setiap instans baru diluncurkan di grup Amazon EC2 Auto Scaling. Meskipun opsi ini meluncurkan instans Amazon EC2 baru lebih lambat daripada opsi lainnya, opsi ini memberikan kemungkinan lebih besar bahwa penerapan awal instans akan berhasil. Opsi ini menggunakan versi terbaru dari CodeDeploy agen.

Integrasi CodeDeploy dengan Elastic Load Balancing

Selama CodeDeploy penerapan, penyeimbang beban mencegah lalu lintas internet dialihkan ke instance ketika mereka tidak siap, saat ini sedang digunakan, atau tidak lagi diperlukan sebagai bagian dari lingkungan. Peran yang tepat yang dimainkan penyeimbang beban, bagaimanapun, tergantung pada apakah itu digunakan dalam blue/green penerapan atau penerapan di tempat.

> **Note**
>
> Penggunaan penyeimbang beban Elastic Load Balancing adalah wajib dalam blue/green penerapan dan opsional dalam penerapan di tempat.

Jenis Elastic Load Balancing

Elastic Load Balancing menyediakan tiga jenis load balancer yang dapat digunakan dalam CodeDeploy penerapan: Classic Load Balancer, Application Load Balancers, dan Network Load Balancer.

Classic Load Balancer

Rute dan load balances baik di transport layer (TCP/SSL) atau layer aplikasi (HTTP/HTTPS). Ini mendukung VPC.

> **Note**
>
> Classic Load Balancer tidak didukung dengan penerapan Amazon ECS.

Penyeimbang Beban Aplikasi

Rute dan load balances pada layer aplikasi (HTTP/HTTPS) dan mendukung routing berbasis jalur. Ini dapat merutekan permintaan ke port pada setiap instans EC2 atau instance kontainer di cloud pribadi virtual (VPC) Anda.

> **Note**
>
> Grup target Application Load Balancer harus memiliki tipe target `instance` untuk penerapan pada instans EC2, dan untuk penerapan Fargate. `IP` Untuk informasi selengkapnya, lihat Jenis target.

Penyeimbang Beban Jaringan

Rute dan load balances pada transport layer (TCP/UDP Layer-4) berdasarkan informasi alamat yang diekstrak dari header paket TCP, bukan dari konten paket. Network Load Balancer dapat menangani rentetan lalu lintas, mempertahankan IP sumber klien, dan menggunakan IP tetap untuk masa aktif load balancer.

Untuk mempelajari lebih lanjut tentang penyeimbang beban Elastic Load Balancing, lihat topik berikut:

- Apa itu Penyeimbangan Beban Elastis?
- Apa itu Classic Load Balancer?
- Apa itu Penyeimbang Beban Aplikasi?
- Apa itu Penyeimbang Beban Jaringan?

Blue/Green penyebaran

Mengalihkan lalu lintas instans di belakang penyeimbang beban Elastic Load Balancing sangat penting untuk penerapan. CodeDeploy blue/green

Selama blue/green penerapan, penyeimbang beban memungkinkan lalu lintas dialihkan ke instance baru dalam grup penyebaran tempat revisi aplikasi terbaru telah digunakan (lingkungan pengganti), sesuai dengan aturan yang Anda tentukan, dan kemudian memblokir lalu lintas dari contoh lama di mana revisi aplikasi sebelumnya berjalan (lingkungan asli).

Setelah instance di lingkungan pengganti terdaftar dengan satu atau lebih penyeimbang beban, instance dari lingkungan asli dideregistrasi dan, jika Anda mau, dihentikan.

Untuk blue/green penerapan, Anda dapat menentukan satu atau beberapa Classic Load Balancer, grup target Application Load Balancer, atau grup target Network Load Balancer di grup penyebaran Anda. Anda menggunakan CodeDeploy konsol atau AWS CLI menambahkan penyeimbang beban ke grup penerapan.

Untuk informasi selengkapnya tentang penyeimbang beban dalam blue/green penerapan, lihat topik berikut:

- Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2
- Buat aplikasi untuk blue/green penyebaran (konsol)
- Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol)

In-place penyebaran

Selama penerapan di tempat, penyeimbang beban mencegah lalu lintas internet dirutekan ke instance saat sedang digunakan, dan kemudian membuat instance tersedia untuk lalu lintas lagi setelah penerapan ke instance itu selesai.

Jika penyeimbang beban tidak digunakan selama penerapan di tempat, lalu lintas internet mungkin masih diarahkan ke instance selama proses penerapan. Akibatnya, pelanggan Anda mungkin mengalami aplikasi web yang rusak, tidak lengkap, atau usang. Saat Anda menggunakan penyeimbang beban Elastic Load Balancing dengan penerapan di tempat, instance dalam grup penerapan dideregistrasi dari penyeimbang beban, diperbarui dengan revisi aplikasi terbaru, dan kemudian didaftarkan ulang dengan penyeimbang beban sebagai bagian dari grup penyebaran yang sama setelah penerapan berhasil. CodeDeploy akan menunggu hingga 1 jam agar instance menjadi sehat di belakang penyeimbang beban. Jika instance tidak ditandai sebagai sehat oleh penyeimbang beban selama masa tunggu, CodeDeploy pindah ke instance berikutnya atau gagal penerapan, berdasarkan konfigurasi penerapan.

Untuk penerapan di tempat, Anda dapat menentukan satu atau beberapa Classic Load Balancer, grup target Application Load Balancer, atau grup target Network Load Balancer. Anda dapat menentukan penyeimbang beban sebagai bagian dari konfigurasi grup penyebaran, atau Anda dapat menggunakan skrip yang disediakan oleh CodeDeploy untuk mengimplementasikan penyeimbang beban.

Tentukan penyeimbang beban penerapan di tempat menggunakan grup penerapan

Untuk menambahkan penyeimbang beban ke grup penerapan, Anda menggunakan konsol atau. CodeDeploy AWS CLI Untuk informasi tentang menentukan penyeimbang beban dalam grup penerapan untuk penerapan di tempat, lihat topik berikut:

- Buat aplikasi untuk penyebaran di tempat (konsol)
- Buat grup penerapan untuk penerapan di tempat (konsol)
- Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2

Tentukan penyeimbang beban penerapan di tempat menggunakan skrip

Gunakan langkah-langkah dalam prosedur berikut untuk menggunakan skrip siklus hidup penerapan guna menyiapkan penyeimbangan beban untuk penerapan di tempat.

> **Note**
>
> Anda harus menggunakan konfigurasi CodeDeployDefault.OneAtATime penerapan hanya ketika Anda menggunakan skrip untuk menyiapkan penyeimbang beban untuk penerapan di tempat. Proses bersamaan tidak didukung, dan CodeDeployDefault.OneAtATime pengaturan memastikan eksekusi serial skrip. Untuk informasi selengkapnya tentang konfigurasi penerapan, lihat. Bekerja dengan konfigurasi penerapan di CodeDeploy

Dalam repositori CodeDeploy Sampel aktif GitHub, kami memberikan instruksi dan sampel yang dapat Anda sesuaikan untuk menggunakan penyeimbang beban CodeDeploy Elastic Load Balancing. Repositori ini mencakup tiga contoh skrip —`register_with_elb.sh`,`deregister_from_elb.sh`, dan `common_functions.sh` —yang menyediakan semua kode yang Anda butuhkan untuk memulai. Cukup edit placeholder di ketiga skrip ini, dan kemudian referensi skrip ini dari file Anda. `appspec.yml`

Untuk menyiapkan penerapan di tempat dengan instans Amazon CodeDeploy EC2 yang terdaftar dengan penyeimbang beban Elastic Load Balancing, lakukan hal berikut:

1. Unduh sampel untuk jenis penyeimbang beban yang ingin Anda gunakan untuk penerapan di tempat:

   - Classic Load Balancer

- Application Load Balancer atau Network Load Balancer (skrip yang sama dapat digunakan untuk kedua jenis)

2. Pastikan setiap instans Amazon EC2 target Anda telah diinstal. AWS CLI
3. Pastikan setiap instans Amazon EC2 target Anda memiliki profil instans IAM yang dilampirkan dengan, setidaknya, izin elasticloadbalancing: * dan autoscaling: *.
4. Sertakan dalam direktori kode sumber aplikasi Anda skrip peristiwa siklus hidup penerapan (`register_with_elb.sh`,, `deregister_from_elb.sh` dan). `common_functions.sh`
5. Dalam `appspec.yml` untuk revisi aplikasi, berikan instruksi CodeDeploy untuk menjalankan `register_with_elb.sh` skrip selama ApplicationStartacara dan `deregister_from_elb.sh` skrip selama ApplicationStopacara berlangsung.
6. Jika instans merupakan bagian dari grup Amazon EC2 Auto Scaling, Anda dapat melewati langkah ini.

   Dalam `common_functions.sh` naskah:

   - Jika Anda menggunakan Classic Load Balancer, tentukan nama penyeimbang beban Elastic Load Balancing, dan buat perubahan apa pun yang Anda perlukan pada pengaturan penerapan lain `ELB_LIST=""` dalam file.
   - Jika Anda menggunakan Application Load Balancer atau Network Load Balancer, tentukan nama nama grup target Elastic Load Balancing, dan buat perubahan apa pun yang Anda perlukan pada pengaturan penyebaran lain `TARGET_GROUP_LIST=""` dalam file.
7. Bundel kode sumber aplikasi Anda, skrip peristiwa siklus hidup penerapan`appspec.yml`, dan penerapan ke dalam revisi aplikasi, lalu unggah revisi tersebut. Terapkan revisi ke instans Amazon EC2. Selama penerapan, skrip peristiwa siklus hidup penerapan akan membatalkan pendaftaran instans Amazon EC2 dengan penyeimbang beban, menunggu koneksi terkuras, dan kemudian mendaftarkan ulang instans Amazon EC2 dengan penyeimbang beban setelah penerapan selesai.

Integrasi dengan produk dan layanan mitra

CodeDeploy memiliki integrasi bawaan untuk produk dan layanan mitra berikut:

Ansible	Jika Anda sudah memiliki satu set buku pedoman Ansible, tetapi hanya perlu di suatu tempat untuk menjalankannya, templat

	untuk Ansible dan CodeDeploy menunjukk an bagaimana beberapa kait penerapan sederhana dapat memastikan Ansible tersedia di instance penerapan lokal dan menjalank an buku pedoman. Jika Anda sudah memiliki proses untuk membangun dan memelihara inventaris Anda, ada juga modul Ansible yang dapat Anda gunakan untuk menginstal dan menjalankan CodeDeploy agen. Pelajari lebih lanjut: • Ansible dan CodeDeploy
Atlassian — Bambu dan Bitbucket	CodeDeploy Tugas untuk Bamboo mengompre s direktori yang berisi AppSpec file ke dalam file.zip, mengunggah file ke Amazon S3, dan kemudian memulai penerapan sesuai dengan konfigurasi yang disediakan dalam aplikasi. CodeDeploy Dukungan Bitbucket Atlassian CodeDeplo y memungkinkan Anda untuk mendorong kode ke instans Amazon EC2 langsung dari UI Bitbucket, sesuai permintaan, ke grup penyebaran Anda. Ini berarti bahwa setelah Anda memperbarui kode di repositori Bitbucket , Anda tidak perlu masuk ke platform integrasi berkelanjutan (CI) atau instans Amazon EC2 untuk menjalankan proses penerapan manual. Pelajari lebih lanjut: • Menggunakan CodeDeploy tugas untuk Bambu • Mengumumkan dukungan Atlassian Bitbucket untuk CodeDeploy

Koki	AWS menyediakan dua sampel template untuk mengintegrasikan Chef dan CodeDeploy. Yang pertama adalah buku masak Chef yang menginstal dan memulai agen. CodeDeploy Ini memungkinkan Anda untuk terus mengelola infrastruktur host Anda dengan Chef saat menggunakan CodeDeploy. Contoh template kedua menunjukkan bagaimana menggunakan CodeDeploy untuk mengatur menjalankan buku masak dan resep dengan chef-solo pada setiap node. Pelajari lebih lanjut: • Chef dan CodeDeploy
CircleCI	CircleCI menyediakan pengujian otomatis dan integrasi berkelanjutan dan toolset penerapan. Setelah Anda membuat peran IAM AWS untuk digunakan dengan CircleCI dan mengonfig urasi parameter penerapan di file circle.yl, Anda dapat menggunakan CircleCI with CodeDeplo y untuk membuat revisi aplikasi, mengungga hnya ke bucket Amazon S3, lalu memulai dan memantau penerapan Anda. Pelajari lebih lanjut: • Menggunakan CircleCI Orb untuk menyebark an aplikasi ke AWS CodeDeploy

CloudBees	Anda dapat menggunakan plugin CodeDeploy Jenkins, tersedia di CloudBeesDEV @cloud, sebagai tindakan pasca-build. Misalnya, di akhir pipeline pengiriman berkelanjutan, Anda dapat menggunakannya untuk menyebarkan revisi aplikasi ke armada server Anda. Pelajari lebih lanjut: • CodeDeploy Plugin Jenkins sekarang tersedia di DEV @cloud
Kodeship	Anda dapat menggunakan Codeship untuk menyebarkan revisi aplikasi melalui. CodeDeploy Anda dapat menggunakan Codeship UI untuk menambahkan CodeDeploy ke pipeline penerapan untuk cabang. Pelajari lebih lanjut: • Menyebarkan ke CodeDeploy • CodeDeploy integrasi pada Codeship
GitHub	Anda dapat menggunakan CodeDeploy untuk menyebarkan revisi aplikasi dari GitHubrepositori. Anda juga dapat memicu penyebaran dari GitHub repositori setiap kali kode sumber di repositori itu diubah. Pelajari lebih lanjut: • Integrasi dengan CodeDeploy GitHub • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub • Secara otomatis menyebarkan dari GitHub menggunakan CodeDeploy

HashiCorp Konsul	Anda dapat menggunakan alat HashiCorp Konsul open-source untuk membantu memastikan kesehatan dan stabilitas lingkungan aplikasi Anda saat Anda menerapkan aplikasi. CodeDeploy Anda dapat menggunakan Konsul untuk mendaftarkan aplikasi yang akan ditemukan selama penerapan, menempatkan aplikasi dan node dalam mode pemeliharaan untuk menghilangkannya dari penerapan, dan menghentikan penerapan jika instance target menjadi tidak sehat. Pelajari lebih lanjut: • CodeDeploy penyebaran dengan Konsul HashiCorp
Jenkins	Plugin CodeDeploy Jenkins menyediakan langkah pasca-build untuk proyek Jenkins Anda. Setelah build berhasil, ia meritsleting ruang kerja, mengunggah ke Amazon S3, dan memulai penerapan baru. Pelajari lebih lanjut: • CodeDeployPlugin Jenkins • Menyiapkan plugin Jenkins untuk CodeDeploy

Lab Boneka	AWS menyediakan contoh template untuk Wayang dan CodeDeploy. Yang pertama adalah modul Puppet yang menginstal dan memulai agen. CodeDeploy Ini memungkin kan Anda untuk terus mengelola infrastruktur host Anda dengan Puppet saat menggunak an CodeDeploy. Contoh template kedua menunjukkan bagaimana menggunakan CodeDeploy untuk mengatur menjalankan modul dan memanifestasikan dengan boneka tanpa master pada setiap node. Pelajari lebih lanjut: • Boneka dan CodeDeploy
SaltStack	Anda dapat mengintegrasikan SaltStack infrastruktur dengan CodeDeploy. Anda dapat menggunakan CodeDeploy modul untuk menginstal dan menjalankan CodeDeploy agen pada antek-antek Anda atau, dengan beberapa kait penyebaran sederhana, Anda dapat menggunakan CodeDeploy untuk mengatur jalannya Salt States Anda. Pelajari lebih lanjut: • SaltStack dan CodeDeploy

TeamCity	Anda dapat menggunakan plugin CodeDeploy Runner untuk menyebarkan aplikasi langsung dari. TeamCity Plugin menambahkan langkah TeamCity build yang menyiapkan dan mengunggah revisi aplikasi ke bucket Amazon S3, mendaftarkan revisi dalam CodeDeploy aplikasi, membuat penerapan dan, jika Anda mau, menunggu CodeDeploy penerapan selesai. Pelajari lebih lanjut: • CodeDeploy Runner (Unduh) • CodeDeploy Plugin pelari (Dokumentasi)
Travis CI	Anda dapat mengonfigurasi Travis CI untuk memicu penerapan CodeDeploy setelah build berhasil. Pelajari lebih lanjut: • Travis CI dan penerapan CodeDeploy

Topik

- Integrasi dengan CodeDeploy GitHub

Integrasi dengan CodeDeploy GitHub

CodeDeploy mendukung GitHub, hosting kode berbasis web dan layanan berbagi. CodeDeploy dapat menerapkan revisi aplikasi yang disimpan di GitHub repositori atau bucket Amazon S3 ke instance. CodeDeploy mendukung GitHub untuk EC2/On-Premises penerapan saja.

Topik

- Menyebarkan CodeDeploy revisi dari GitHub
- GitHub perilaku dengan CodeDeploy

Menyebarkan CodeDeploy revisi dari GitHub

Untuk menerapkan revisi aplikasi dari GitHub repositori ke instance:

1. Buat revisi yang kompatibel dengan CodeDeploy dan jenis instans Amazon EC2 yang akan Anda gunakan.

   Untuk membuat revisi yang kompatibel, ikuti petunjuk di Rencanakan revisi untuk CodeDeploy danTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.
2. Gunakan GitHub akun untuk menambahkan revisi Anda ke repositori GitHub .

   Untuk membuat GitHub akun, lihat Bergabung GitHub. Untuk membuat GitHub repositori, lihat Membuat repo.
3. Gunakan halaman Create deployment di CodeDeploy konsol atau AWS CLI create-deployment perintah untuk menyebarkan revisi Anda dari GitHub repositori Anda untuk menargetkan instance yang dikonfigurasi untuk digunakan dalam penerapan. CodeDeploy

   Jika Anda ingin memanggil create-deployment perintah, Anda harus terlebih dahulu menggunakan halaman Create deployment konsol untuk memberikan CodeDeploy izin untuk berinteraksi dengan GitHub atas nama GitHub akun pilihan Anda untuk aplikasi yang ditentukan. Anda hanya perlu melakukan ini sekali per aplikasi.

   Untuk mempelajari cara menggunakan halaman Create deployment untuk menyebarkan dari GitHub repositori, lihat. Buat penerapan dengan CodeDeploy

   Untuk mempelajari cara memanggil create-deployment perintah untuk menyebarkan dari GitHub repositori, lihat. Membuat penerapan Platform EC2/On-Premises Komputasi (CLI)

   Untuk mempelajari cara menyiapkan instance untuk digunakan dalam CodeDeploy penerapan, lihat. Bekerja dengan instance untuk CodeDeploy

Untuk informasi selengkapnya, lihat Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub.

GitHub perilaku dengan CodeDeploy

Topik

- GitHub otentikasi dengan aplikasi di CodeDeploy
- CodeDeploy interaksi dengan GitHub repositori pribadi dan publik

- CodeDeploy interaksi dengan repositori yang dikelola organisasi GitHub
- Secara otomatis menyebarkan dari CodePipeline dengan CodeDeploy

GitHub otentikasi dengan aplikasi di CodeDeploy

Setelah Anda memberikan CodeDeploy izin untuk berinteraksi GitHub, hubungan antara GitHub akun dan aplikasi itu disimpan CodeDeploy. Anda dapat menautkan aplikasi ke GitHub akun lain. Anda juga dapat mencabut izin CodeDeploy untuk berinteraksi. GitHub

Untuk menautkan GitHub akun ke aplikasi di CodeDeploy

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pilih aplikasi yang ingin Anda tautkan ke GitHub akun lain.
4. Jika aplikasi Anda tidak memiliki grup penyebaran, pilih Buat grup penerapan untuk membuatnya. Untuk informasi selengkapnya, lihat Buat grup penerapan dengan CodeDeploy. Grup penyebaran diperlukan untuk memilih Buat penerapan di langkah berikutnya.
5. Dari Deployment, pilih Create deployment.

   > **Note**
   >
   > Anda tidak perlu membuat penerapan baru. Saat ini ini satu-satunya cara untuk menautkan GitHub akun yang berbeda ke aplikasi.

6. Dalam pengaturan Deployment, untuk jenis Revisi, pilih Aplikasi saya disimpan di. GitHub
7. Lakukan salah satu tindakan berikut:

   - Untuk membuat koneksi untuk AWS CodeDeploy aplikasi ke GitHub akun, keluar dari GitHub tab browser web terpisah. Dalam nama GitHub token, ketikkan nama untuk mengidentifikasi koneksi ini, lalu pilih Connect to GitHub. Halaman web meminta Anda untuk mengizinkan CodeDeploy untuk berinteraksi dengan GitHub aplikasi Anda. Lanjutkan ke langkah 10.

- Untuk menggunakan koneksi yang telah Anda buat, dalam nama GitHub token, pilih namanya, lalu pilih Connect to GitHub. Lanjutkan ke langkah 8.
- Untuk membuat koneksi ke GitHub akun lain, keluar dari GitHub tab browser web terpisah. Dalam nama GitHub token, ketikkan nama untuk mengidentifikasi koneksi, lalu pilih Connect to GitHub. Halaman web meminta Anda untuk mengizinkan CodeDeploy untuk berinteraksi dengan GitHub aplikasi Anda. Lanjutkan ke langkah 10.

8. Jika Anda belum masuk GitHub, ikuti petunjuk di halaman Masuk untuk masuk dengan GitHub akun yang ingin Anda tautkan aplikasi.
9. Pilih Otorisasi aplikasi. GitHub memberikan CodeDeploy izin untuk berinteraksi dengan GitHub atas nama GitHub akun yang masuk untuk aplikasi yang dipilih.
10. Jika Anda tidak ingin membuat penerapan, pilih Batal.

Untuk mencabut izin untuk berinteraksi CodeDeploy dengan GitHub

1. Masuk GitHub menggunakan kredensil untuk GitHub akun tempat Anda ingin mencabut izinnya. AWS CodeDeploy
2. Buka halaman GitHub Aplikasi, cari CodeDeploydi daftar aplikasi resmi, dan kemudian ikuti GitHub prosedur untuk mencabut otorisasi untuk suatu aplikasi.

CodeDeploy interaksi dengan GitHub repositori pribadi dan publik

CodeDeploy mendukung penyebaran aplikasi dari GitHub repositori pribadi dan publik. Ketika Anda memberikan CodeDeploy izin untuk mengakses GitHub atas nama Anda, CodeDeploy akan memiliki akses baca-tulis ke semua GitHub repositori pribadi yang memiliki akses akun Anda GitHub . Namun, CodeDeploy hanya membaca dari GitHub repositori. Itu tidak akan menulis ke salah satu GitHub repositori pribadi Anda.

CodeDeploy interaksi dengan repositori yang dikelola organisasi GitHub

Secara default, GitHub repositori yang dikelola oleh organisasi (sebagai lawan dari repositori pribadi atau publik akun Anda sendiri) tidak memberikan akses ke aplikasi pihak ketiga, termasuk. CodeDeploy Penerapan Anda akan gagal jika pembatasan aplikasi pihak ketiga organisasi diaktifkan GitHub dan Anda mencoba menerapkan kode dari repositorinya GitHub . Ada dua cara untuk mengatasi masalah ini.

- Sebagai anggota organisasi, Anda dapat meminta pemilik organisasi untuk menyetujui akses ke CodeDeploy. Langkah-langkah untuk meminta akses ini tergantung pada apakah Anda telah mengotorisasi CodeDeploy akun pribadi Anda:
  - Jika Anda memiliki akses resmi ke CodeDeploy akun Anda, lihat Meminta persetujuan organisasi untuk aplikasi resmi Anda.
  - Jika Anda belum mengotorisasi akses ke CodeDeploy akun Anda, lihat Meminta persetujuan organisasi untuk aplikasi pihak ketiga.
- Pemilik organisasi dapat menonaktifkan semua pembatasan aplikasi pihak ketiga untuk organisasi. Untuk selengkapnya, lihat Menonaktifkan pembatasan aplikasi pihak ketiga untuk organisasi Anda.

Untuk informasi selengkapnya, lihat Tentang pembatasan aplikasi pihak ketiga.

Secara otomatis menyebarkan dari CodePipeline dengan CodeDeploy

Anda dapat memicu penyebaran dari CodePipeline setiap kali kode sumber berubah. Untuk informasi lebih lanjut, lihat. CodePipeline

Contoh integrasi dari komunitas

Bagian berikut menyediakan tautan ke unggahan blog, artikel, dan contoh yang disediakan komunitas.

> **Note**
>
> Tautan ini disediakan untuk kepentingan informasi saja, dan tidak boleh dianggap sebagai daftar komprehensif atau dukungan atas konten contoh. AWS tidak bertanggung jawab atas konten atau akurasi dari konten eksternal.

Unggahan blog

- Mengotomatiskan CodeDeploy penyediaan di CloudFormation

  Pelajari cara menyediakan penerapan aplikasi CodeDeploy dengan menggunakan CloudFormation.

  Telah Terbit Januari 2016

- AWS Toolkit for Eclipse Integrasi dengan CodeDeploy (Bagian 1)

  AWS Toolkit for Eclipse Integrasi dengan CodeDeploy (Bagian 2)

  AWS Toolkit for Eclipse Integrasi dengan CodeDeploy (Bagian 3)

  Pelajari bagaimana pengembang Java dapat menggunakan CodeDeploy plugin untuk Eclipse untuk menyebarkan aplikasi web AWS langsung dari lingkungan pengembangan Eclipse.

  Diterbitkan Februari 2015

- Secara otomatis menyebarkan dari GitHub menggunakan CodeDeploy

  Pelajari cara penerapan otomatis dari GitHub ke CodeDeploy dapat digunakan untuk membuat pipeline ujung-ke-ujung — mulai dari kontrol sumber hingga lingkungan pengujian atau produksi Anda.

  Telah Terbit Desember 2014

CodeDeploy tutorial

Bagian ini mencakup beberapa tutorial untuk membantu Anda mempelajari cara menggunakan CodeDeploy.

Prosedur dalam tutorial ini memberikan saran untuk lokasi penyimpanan file (misalnya, c:\temp) dan nama yang akan diberikan ke ember, subfolder, atau file (misalnya, amzn-s3-demo-bucket,, dan CodeDeployDemo-EC2-Trust.json, masing-masing) HelloWorldApp, tetapi Anda tidak diharuskan menggunakannya. Pastikan untuk mengganti lokasi dan nama file Anda saat Anda melakukan prosedur.

Topik

- Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix)
- Tutorial: Menyebarkan “halo, dunia!” aplikasi dengan CodeDeploy (Windows Server)
- Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)
- Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling
- Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub
- Tutorial: Menyebarkan aplikasi ke Amazon ECS
- Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi
- Tutorial: Menyebarkan fungsi CodeDeploy Lambda yang diperbarui dengan dan AWS Serverless Application Model

Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix)

Dalam tutorial ini, Anda menyebarkan WordPress, alat blogging open source dan sistem manajemen konten berdasarkan PHP dan MySQL, ke satu instans Amazon EC2 yang menjalankan Amazon Linux atau Red Hat Enterprise Linux (RHEL).

Bukan apa yang Anda cari?

- Untuk mempraktikkan penerapan ke instans Amazon EC2 yang menjalankan Windows Server sebagai gantinya, lihat. Tutorial: Menyebarkan “halo, dunia!” aplikasi dengan CodeDeploy (Windows Server)
- Untuk mempraktikkan penerapan ke instans lokal, bukan instans Amazon EC2, lihat. Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)

Langkah-langkah tutorial ini disajikan dari perspektif mesin pengembangan lokal yang menjalankan Linux, macOS, atau Unix. Meskipun Anda dapat menyelesaikan sebagian besar langkah-langkah ini pada mesin lokal yang menjalankan Windows, Anda harus menyesuaikan langkah-langkah yang mencakup perintah seperti chmod danwget, aplikasi seperti sed, dan jalur direktori seperti`/tmp`.

Sebelum Anda memulai tutorial ini, Anda harus menyelesaikan prasyarat di. Memulai dengan CodeDeploy Ini termasuk mengonfigurasi pengguna, menginstal atau meningkatkan AWS CLI, dan membuat profil instans IAM dan peran layanan.

Topik

- Langkah 1: Luncurkan dan konfigurasikan instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux
- Langkah 2: Konfigurasikan konten sumber Anda untuk disebarkan ke instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux Amazon EC2
- Langkah 3: Unggah WordPress aplikasi Anda ke Amazon S3
- Langkah 4: Menyebarkan aplikasi Anda WordPress
- Langkah 5: Perbarui dan terapkan ulang aplikasi Anda WordPress
- Langkah 6: Bersihkan WordPress aplikasi Anda dan sumber daya terkait

Langkah 1: Luncurkan dan konfigurasikan instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux

Untuk menerapkan WordPress aplikasi CodeDeploy, Anda memerlukan instans Amazon EC2 yang menjalankan Amazon Linux atau Red Hat Enterprise Linux (RHEL). Instans Amazon EC2 memerlukan aturan keamanan masuk baru yang memungkinkan koneksi HTTP. Aturan ini diperlukan untuk melihat WordPress halaman di browser setelah berhasil digunakan.

Ikuti petunjuk dalam Buat instans Amazon EC2 untuk CodeDeploy. Saat Anda mendapatkan bagian dalam instruksi tersebut tentang menetapkan tag instans Amazon EC2 ke instance, pastikan untuk menentukan kunci tag dan **Name** nilai tag. **CodeDeployDemo** (Jika Anda menentukan kunci tag atau nilai tag yang berbeda, maka instruksi di Langkah 4: Menyebarkan aplikasi Anda WordPress dapat menghasilkan hasil yang tidak terduga.)

Setelah Anda mengikuti petunjuk untuk meluncurkan instans Amazon EC2, kembali ke halaman ini, dan lanjutkan ke bagian berikutnya. Jangan melanjutkan ke Buat aplikasi dengan CodeDeploy langkah berikutnya.

Connect ke instans Amazon Linux atau RHEL Amazon EC2

Setelah instans Amazon EC2 baru Anda diluncurkan, ikuti petunjuk ini untuk berlatih menghubungkannya.

1. Gunakan ssh perintah (atau emulator SSH-capable terminal seperti PuTTY) untuk terhubung ke instans Amazon Linux atau RHEL Amazon EC2 Anda. Anda akan memerlukan alamat DNS publik dari instans dan kunci pribadi untuk key pair yang Anda gunakan saat memulai instans Amazon EC2. Untuk informasi selengkapnya, lihat Connect to Your Instance.

   Misalnya, jika alamat DNS publik**ec2-01-234-567-890.compute-1.amazonaws.com**, dan key pair instans Amazon EC2 untuk akses SSH **codedeploydemo.pem** diberi nama, Anda akan mengetik:

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

   Ganti */path/to/*`codedeploydemo.pem` dengan jalur ke `.pem` file Anda dan contoh alamat DNS dengan alamat ke instans Amazon Linux atau RHEL Amazon EC2 Anda.

   > **Note**
   >
   > Jika Anda menerima kesalahan tentang izin file kunci Anda yang terlalu terbuka, Anda harus membatasi izinnya untuk memberikan akses hanya kepada pengguna saat ini (Anda). Misalnya, dengan chmod perintah di Linux, macOS, atau Unix, ketik:

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

2. Setelah Anda masuk, Anda akan melihat spanduk AMI untuk instans Amazon EC2. Untuk Amazon Linux, seharusnya terlihat seperti ini:

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

3. Anda sekarang dapat keluar dari instans Amazon EC2 yang sedang berjalan.

> **Warning**
>
> Jangan menghentikan atau menghentikan instans Amazon EC2. Jika CodeDeploy tidak, tidak akan dapat menerapkannya.

Tambahkan aturan masuk yang memungkinkan lalu lintas HTTP ke instans Amazon Linux atau RHEL Amazon EC2 Anda

Langkah selanjutnya mengonfirmasi instans Amazon EC2 Anda memiliki port HTTP terbuka sehingga Anda dapat melihat halaman beranda WordPress aplikasi yang digunakan di browser.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Pilih Instans, lalu pilih instans Anda.
3. Pada tab Deskripsi, di bawah Grup keamanan, pilih lihat aturan masuk.

   Anda akan melihat daftar aturan di grup keamanan Anda seperti berikut ini:

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

4. Di bawah Grup keamanan, pilih grup keamanan untuk instans Amazon EC2 Anda. Itu bisa diberi nama**launch-wizard-*N***. ***N***Dalam nama adalah nomor yang ditetapkan ke grup keamanan Anda saat instance Anda dibuat.

   Pilih tab Masuk. Jika grup keamanan untuk instans Anda dikonfigurasi dengan benar, Anda akan melihat aturan dengan nilai-nilai berikut:

- Jenis: HTTP
- Protokol: TCP
- Rentang Port: 80
- Sumber: 0.0.0. 0/0

5. Jika Anda tidak melihat aturan dengan nilai-nilai ini, gunakan prosedur dalam Menambahkan Aturan ke Grup Keamanan untuk menambahkannya ke aturan keamanan baru.

Langkah 2: Konfigurasikan konten sumber Anda untuk disebarkan ke instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux Amazon EC2

Sekarang saatnya untuk mengonfigurasi konten sumber aplikasi Anda sehingga Anda memiliki sesuatu untuk diterapkan ke instance.

Topik

- Dapatkan kode sumbernya
- Buat skrip untuk menjalankan aplikasi Anda
- Tambahkan file spesifikasi aplikasi

Dapatkan kode sumbernya

Untuk tutorial ini, Anda menerapkan platform penerbitan WordPress konten dari mesin pengembangan Anda ke instans Amazon EC2 target. Untuk mendapatkan kode WordPress sumber, Anda dapat menggunakan panggilan baris perintah bawaan. Atau, jika Anda telah menginstal Git di mesin pengembangan Anda, Anda dapat menggunakannya sebagai gantinya.

Untuk langkah-langkah ini, kami berasumsi Anda mengunduh salinan kode WordPress sumber ke `/tmp` direktori di mesin pengembangan Anda. (Anda dapat memilih direktori apa pun yang Anda suka, tetapi ingat untuk mengganti lokasi Anda `/tmp` di mana pun itu ditentukan dalam langkah-langkah ini.)

Pilih salah satu dari dua opsi berikut untuk menyalin file WordPress sumber ke mesin pengembangan Anda. Opsi pertama menggunakan panggilan baris perintah bawaan. Opsi kedua menggunakan Git.

Topik

- Untuk mendapatkan salinan kode WordPress sumber (panggilan baris perintah bawaan)
- Untuk mendapatkan salinan kode WordPress sumber (Git)

Untuk mendapatkan salinan kode WordPress sumber (panggilan baris perintah bawaan)

1. Panggil wget perintah untuk mengunduh salinan kode WordPress sumber, sebagai file.zip, ke direktori saat ini:

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

2. Panggilunzip,mkdir,cp, dan rm perintah untuk:

   - Buka paket `master.zip` file ke `/tmp/WordPress_Temp` direktori (folder).
   - Salin isinya yang tidak di-zip ke folder `/tmp/WordPress` tujuan.
   - Hapus `/tmp/WordPress_Temp` folder dan `master` file sementara.

   Jalankan perintah satu per satu:

```
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
```

   Ini meninggalkan Anda dengan satu set file kode WordPress sumber yang bersih di `/tmp/WordPress` folder.

Untuk mendapatkan salinan kode WordPress sumber (Git)

1. Unduh dan instal Git di mesin pengembangan Anda.
2. Di `/tmp/WordPress` folder, panggil git init perintah.

3. Panggil git clone perintah untuk mengkloning WordPress repositori publik, buat salinannya sendiri di folder tujuan: `/tmp/WordPress`

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

   Ini meninggalkan Anda dengan satu set file kode WordPress sumber yang bersih di `/tmp/WordPress` folder.

Buat skrip untuk menjalankan aplikasi Anda

Selanjutnya, buat folder dan skrip di direktori. CodeDeploy menggunakan skrip ini untuk menyiapkan dan menerapkan revisi aplikasi Anda pada instans Amazon EC2 target. Anda dapat menggunakan editor teks apa pun untuk membuat skrip.

1. Buat direktori skrip dalam salinan kode WordPress sumber Anda:

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

2. Buat `install_dependencies.sh` file di`/tmp/WordPress/scripts`. Tambahkan baris berikut ke file. `install_dependencies.sh`Script ini menginstal Apache, MySQL, dan PHP. Ini juga menambahkan dukungan MySQL ke PHP.

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

3. Buat `start_server.sh` file di`/tmp/WordPress/scripts`. Tambahkan baris berikut ke file. `start_server.sh`Script ini dimulai Apache dan MySQL.

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

4. Buat `stop_server.sh` file di`/tmp/WordPress/scripts`. Tambahkan baris berikut ke file. `stop_server.sh`Script ini menghentikan Apache dan 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. Buat `create_test_db.sh` file di`/tmp/WordPress/scripts`. Tambahkan baris berikut ke file. `create_test_db.sh`Script ini menggunakan MySQL untuk membuat **`test`** database WordPress untuk digunakan.

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

6. Akhirnya, buat `change_permissions.sh` skrip di`/tmp/WordPress/scripts`. Ini digunakan untuk mengubah izin folder di Apache.

> **Important**
>
> Skrip ini memperbarui izin pada `/tmp/WordPress` folder sehingga siapa pun dapat menulis ke sana. Ini diperlukan agar WordPress dapat menulis ke database-nya selamaLangkah 5: Perbarui dan terapkan ulang aplikasi Anda WordPress. Setelah WordPress aplikasi diatur, jalankan perintah berikut untuk memperbarui izin ke pengaturan yang lebih aman:
>
> ```
> chmod -R 755 /var/www/html/WordPress
> ```

```
#!/bin/bash
```

```
chmod -R 777 /var/www/html/WordPress
```

7. Berikan semua skrip izin yang dapat dieksekusi. Pada baris perintah, ketik:

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

Tambahkan file spesifikasi aplikasi

Selanjutnya, tambahkan file spesifikasi aplikasi (AppSpec file), file berformat YAMB yang digunakan oleh ke: CodeDeploy

- Petakan file sumber dalam revisi aplikasi Anda ke tujuannya pada instans Amazon EC2 target.
- Tentukan izin khusus untuk file yang digunakan.
- Tentukan skrip yang akan dijalankan pada instans Amazon EC2 target selama penerapan.

AppSpec File harus diberi nama`appspec.yml`. Itu harus ditempatkan di direktori root kode sumber aplikasi. Dalam tutorial ini, direktori root adalah `/tmp/WordPress`

Dengan editor teks Anda, buat file bernama`appspec.yml`. Tambahkan baris berikut ke file:

```
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 menggunakan AppSpec file ini untuk menyalin semua file dalam `/tmp/WordPress` folder pada mesin pengembangan ke `/var/www/html/WordPress` folder pada instans Amazon EC2 target. Selama penerapan, CodeDeploy jalankan skrip yang ditentukan seperti pada `/var/www/html/WordPress/scripts` folder `root` pada instans Amazon EC2 target pada peristiwa tertentu selama siklus hidup penerapan, seperti dan. **`BeforeInstall`** **`AfterInstall`** Jika salah satu skrip ini membutuhkan waktu lebih dari 300 detik (5 menit) untuk dijalankan, CodeDeploy hentikan penerapan dan tandai penerapan sebagai gagal.

Untuk informasi selengkapnya tentang pengaturan ini, lihatCodeDeploy AppSpec referensi file.

> **Important**
>
> Lokasi dan jumlah spasi antara masing-masing item dalam file ini penting. Jika spasi salah, CodeDeploy memunculkan kesalahan yang mungkin sulit untuk di-debug. Lihat informasi yang lebih lengkap di AppSpec Spasi file.

Langkah 3: Unggah WordPress aplikasi Anda ke Amazon S3

Sekarang Anda akan mempersiapkan dan mengunggah konten sumber Anda ke lokasi yang CodeDeploy dapat menyebarkannya. Petunjuk berikut menunjukkan cara menyediakan bucket Amazon S3, menyiapkan file revisi aplikasi untuk bucket, menggabungkan file revisi, lalu mendorong revisi ke bucket.

> **Note**
>
> Meskipun tidak tercakup dalam tutorial ini, Anda dapat menggunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub repositori ke instance. Untuk informasi selengkapnya, lihat Integrasi dengan CodeDeploy GitHub.

Topik

- Menyediakan bucket Amazon S3
- Siapkan file aplikasi untuk ember
- Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip

Menyediakan bucket Amazon S3

Buat wadah penyimpanan atau bucket di Amazon S3—atau gunakan bucket yang sudah ada. Pastikan Anda dapat mengunggah revisi ke bucket dan instans Amazon EC2 yang digunakan dalam penerapan dapat mengunduh revisi dari bucket.

Anda dapat menggunakan AWS CLI, konsol Amazon S3, atau API Amazon S3 untuk membuat bucket Amazon S3. Setelah Anda membuat bucket, pastikan untuk memberikan izin akses ke bucket dan AWS akun Anda.

> **Note**
>
> Nama bucket harus unik di Amazon S3 untuk semua AWS akun. Jika Anda tidak dapat menggunakannya**`amzn-s3-demo-bucket`**, coba nama bucket yang berbeda, seperti **`amzn-s3-demo-bucket`** diikuti dengan tanda hubung dan inisiasi Anda atau pengenal unik lainnya. Kemudian pastikan untuk mengganti nama bucket Anda dengan di **`amzn-s3-demo-bucket`** mana pun Anda melihatnya di seluruh tutorial ini.
> Bucket Amazon S3 harus dibuat di AWS Wilayah yang sama tempat instans Amazon EC2 target Anda diluncurkan. Misalnya, jika Anda membuat bucket di Wilayah AS Timur (Virginia N.), instans Amazon EC2 target Anda harus diluncurkan di Wilayah AS Timur (Virginia Utara).

Topik

- Untuk membuat bucket Amazon S3 (CLI)
- Untuk membuat bucket Amazon S3 (konsol)
- Berikan izin ke bucket Amazon S3 dan AWS akun

Untuk membuat bucket Amazon S3 (CLI)

Panggil mb perintah untuk membuat bucket Amazon S3 bernama: **`amzn-s3-demo-bucket`**

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

Untuk membuat bucket Amazon S3 (konsol)

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di konsol Amazon S3, pilih Buat ember.
3. Di kotak nama Bucket, ketikkan nama untuk bucket.
4. Dalam daftar Wilayah, pilih Wilayah target, lalu pilih Buat.

Berikan izin ke bucket Amazon S3 dan AWS akun

Anda harus memiliki izin untuk mengunggah ke bucket Amazon S3. Anda dapat menentukan izin ini melalui kebijakan bucket Amazon S3. Misalnya, dalam kebijakan bucket Amazon S3 berikut, menggunakan karakter wildcard (*) memungkinkan AWS akun `111122223333` untuk mengunggah file ke direktori apa pun di bucket Amazon S3 bernama: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Untuk melihat ID AWS akun, lihat Menemukan ID AWS akun Anda.

Sekarang adalah saat yang tepat untuk memverifikasi bucket Amazon S3 akan memungkinkan permintaan unduhan dari setiap instans Amazon EC2 yang berpartisipasi. Anda dapat menentukan ini melalui kebijakan bucket Amazon S3. Misalnya, dalam kebijakan bucket Amazon S3 berikut, menggunakan karakter wildcard (*) memungkinkan instans Amazon EC2 dengan profil instans IAM terlampir yang berisi ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` untuk mengunduh file dari direktori apa pun di bucket Amazon S3 bernama: `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"
                ]
            }
        }
    ]
}
```

Untuk informasi tentang cara membuat dan melampirkan kebijakan bucket Amazon S3, lihat contoh kebijakan Bucket.

Untuk informasi tentang cara membuat dan melampirkan kebijakan IAM, lihat Bekerja dengan kebijakan.

Siapkan file aplikasi untuk ember

Pastikan file WordPress aplikasi, AppSpec file, dan skrip diatur pada mesin pengembangan Anda mirip dengan yang berikut ini:

```
/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...)
```

Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip

Bundel file WordPress aplikasi dan AppSpec file ke dalam file arsip (dikenal sebagai revisi aplikasi).

> **Note**
>
> Anda mungkin dikenakan biaya untuk menyimpan objek dalam ember dan untuk mentransfer revisi aplikasi ke dalam dan keluar dari ember. Untuk informasi selengkapnya, lihat Harga Amazon S3.

1. Pada mesin pengembangan, beralih ke folder tempat file disimpan:

   ```
   cd /tmp/WordPress
   ```

   > **Note**
   >
   > Jika Anda tidak beralih ke folder ini, maka bundling file akan dimulai di folder Anda saat ini. Misalnya, jika folder Anda saat ini `/tmp` bukan`/tmp/WordPress`, maka bundling akan dimulai dengan file dan subfolder di `tmp` folder, yang mungkin mencakup lebih dari subfolder`WordPress`.

2. Panggil create-application perintah untuk mendaftarkan aplikasi baru bernama**`WordPress_App`**:

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

3. Panggil perintah CodeDeploy push untuk menggabungkan file bersama-sama, mengunggah revisi ke Amazon S3, dan mendaftarkan informasi tentang revisi CodeDeploy yang diunggah, semuanya dalam satu tindakan.

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

```
  --ignore-hidden-files
```

Perintah ini menggabungkan file dari direktori saat ini (tidak termasuk file tersembunyi) ke dalam satu file arsip bernama**`WordPressApp.zip`**, mengunggah revisi ke **`amzn-s3-demo-bucket`** bucket, dan mendaftarkan informasi tentang revisi yang CodeDeploy diunggah.

Langkah 4: Menyebarkan aplikasi Anda WordPress

Sekarang Anda menerapkan contoh revisi WordPress aplikasi yang Anda unggah ke Amazon S3. Anda dapat menggunakan AWS CLI atau CodeDeploy konsol untuk menerapkan revisi dan memantau kemajuan penerapan. Setelah revisi aplikasi berhasil digunakan, Anda memeriksa hasilnya.

Topik

- Terapkan revisi aplikasi Anda dengan CodeDeploy
- Pantau dan pecahkan masalah penerapan Anda
- Verifikasi penerapan Anda

Terapkan revisi aplikasi Anda dengan CodeDeploy

Gunakan AWS CLI atau konsol untuk menerapkan revisi aplikasi Anda.

Topik

- Untuk menerapkan revisi aplikasi Anda (CLI)
- Untuk menerapkan revisi aplikasi Anda (konsol)

Untuk menerapkan revisi aplikasi Anda (CLI)

1. Penyebaran membutuhkan grup penyebaran. Namun, sebelum Anda membuat grup penyebaran, Anda memerlukan peran layanan ARN. Peran layanan adalah peran IAM yang memberikan izin layanan untuk bertindak atas nama Anda. Dalam hal ini, peran layanan memberikan CodeDeploy izin untuk mengakses instans Amazon EC2 Anda untuk memperluas (membaca) tag instans Amazon EC2 mereka.

   Anda seharusnya sudah mengikuti instruksi Buat peran layanan (CLI) untuk membuat peran layanan. Untuk mendapatkan ARN dari peran layanan, lihat. Dapatkan peran layanan ARN (CLI)

2. Sekarang setelah Anda memiliki peran layanan ARN, panggil create-deployment-group perintah untuk membuat grup penyebaran bernama**`WordPress_DepGroup`**, terkait dengan aplikasi bernama**`WordPress_App`**, menggunakan tag Amazon EC2 bernama dan konfigurasi penerapan bernama**`CodeDeployDemo`**: **`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**
>
> Perintah create-deployment-group menyediakan dukungan untuk membuat pemicu yang menghasilkan pengiriman notifikasi Amazon SNS ke pelanggan topik tentang peristiwa tertentu dalam penerapan dan instance. Perintah ini juga mendukung opsi untuk memutar kembali penerapan secara otomatis dan mengatur alarm untuk menghentikan penerapan saat memantau ambang batas di alarm Amazon terpenuhi. CloudWatch Perintah untuk tindakan ini tidak termasuk dalam tutorial ini.

3. Sebelum Anda membuat penerapan, instance dalam grup penyebaran Anda harus memiliki agen yang diinstal. CodeDeploy Anda dapat menginstal agen dari baris perintah AWS Systems Manager dengan perintah berikut:

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

Perintah ini menciptakan asosiasi di Systems Manager State Manager yang akan menginstal CodeDeploy agen dan kemudian mencoba memperbaruinya pada pukul 2:00 setiap Minggu pagi. Untuk informasi lebih lanjut tentang CodeDeploy agen, lihat Bekerja dengan CodeDeploy agen. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu AWS Systems Manager.

4. Sekarang panggil create-deployment perintah untuk membuat deployment yang terkait dengan aplikasi bernama**`WordPress_App`**, konfigurasi deployment bernama**`CodeDeployDefault.OneAtATime`**, dan grup deployment bernama**`WordPress_DepGroup`**, menggunakan revisi aplikasi yang dinamai **`WordPressApp.zip`** dalam bucket bernama: **`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
```

Untuk menerapkan revisi aplikasi Anda (konsol)

1. Sebelum Anda menggunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda, Anda memerlukan peran layanan ARN. Peran layanan adalah peran IAM yang memberikan izin layanan untuk bertindak atas nama Anda. Dalam hal ini, peran layanan memberikan CodeDeploy izin untuk mengakses instans Amazon EC2 Anda untuk memperluas (membaca) tag instans Amazon EC2 mereka.

   Anda seharusnya sudah mengikuti instruksi Buat peran layanan (konsol) untuk membuat peran layanan. Untuk mendapatkan ARN dari peran layanan, lihat. Dapatkan peran layanan ARN (konsol)
2. Sekarang setelah Anda memiliki ARN, gunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda:

   Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.
3. Di panel navigasi, perluas Deploy, lalu pilih Applications.
4. Dalam daftar aplikasi, pilih WordPress_App.
5. Pada tab Deployment groups, pilih Create deployment group.

6. Dalam nama grup Deployment, masukkan**`WordPress_DepGroup`**.
7. Di bawah Jenis Deployment, pilih In-placedeployment.
8. Dalam konfigurasi Lingkungan, pilih instans Amazon EC2.
9. Dalam konfigurasi Agen dengan AWS Systems Manager, pertahankan defaultnya.
10. Di Key, masukkan**`Name`**.
11. Dalam Value (Nilai), masukkan **`CodeDeployDemo`**.

> **Note**
>
> Setelah Anda mengetik**`CodeDeployDemo`**, 1 akan muncul di bawah Instans pencocokan untuk mengonfirmasi CodeDeploy ditemukan satu instans Amazon EC2 yang cocok.

12. Dalam konfigurasi Deployment, pilih CodeDeployDefault.OneAtATime.
13. Di ARN peran Layanan, pilih ARN peran layanan, lalu pilih Buat grup penyebaran.
14. Pilih Buat penerapan.
15. Di grup Deployment pilih**`WordPress_DepGroup`**.
16. Di samping jenis Repositori, pilih Aplikasi saya disimpan di Amazon S3. Di lokasi Revisi, masukkan lokasi contoh revisi WordPress aplikasi yang sebelumnya Anda unggah ke Amazon S3. Untuk mendapatkan lokasi:
    a. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
    b. Dalam daftar bucket, pilih amzn-s3-demo-bucket (atau nama bucket tempat Anda mengunggah revisi aplikasi).
    c. Dalam daftar objek, pilih WordPressApp.zip.
    d. Pada tab Ikhtisar, salin nilai bidang Tautan ke clipboard Anda.

       Mungkin terlihat seperti ini:

       `https://s3.amazonaws.com/amzn-s3-demo-bucket/WordPressApp.zip`
    e. Kembali ke CodeDeploy konsol, dan di lokasi Revisi, tempel nilai bidang Tautan.
17. Jika pesan muncul di daftar Jenis file yang menyatakan jenis file tidak dapat dideteksi, pilih.zip.
18. (Opsional) Ketik komentar di kotak Deskripsi Deployment.
19. Perluas penggantian grup Deployment, dan dari konfigurasi Deployment, pilih. CodeDeployDefault.OneAtATime

20. Pilih Mulai penerapan. Informasi tentang penerapan yang baru dibuat muncul di halaman Deployment.

Pantau dan pecahkan masalah penerapan Anda

Gunakan AWS CLI atau konsol untuk memantau dan memecahkan masalah penyebaran Anda.

Topik

- Untuk memantau dan memecahkan masalah penerapan Anda (CLI)
- Untuk memantau dan memecahkan masalah penerapan Anda (konsol)

Untuk memantau dan memecahkan masalah penerapan Anda (CLI)

1. Dapatkan ID penerapan dengan memanggil list-deployments perintah terhadap aplikasi bernama **`WordPress_App`** dan grup penyebaran bernama: **`WordPress_DepGroup`**

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

2. Panggil get-deployment perintah dengan ID penyebaran:

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

3. Perintah mengembalikan status keseluruhan penerapan. Jika berhasil, nilainya`Succeeded`.

   Jika status keseluruhannya`Failed`, Anda dapat memanggil perintah seperti list-deployment-instance dan get-deployment-instance untuk memecahkan masalah. Untuk opsi pemecahan masalah lainnya, lihat. Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance

Untuk memantau dan memecahkan masalah penerapan Anda (konsol)

Pada halaman Deployment di CodeDeploy konsol, Anda dapat memantau status penerapan di kolom Status.

Untuk mendapatkan informasi selengkapnya tentang penerapan Anda, terutama jika nilai kolom Status memiliki nilai selain Berhasil:

1. Dalam tabel Deployment, pilih nama penyebaran. Setelah penerapan gagal, pesan yang menjelaskan alasan kegagalan ditampilkan.
2. Dalam aktivitas Instance, informasi lebih lanjut tentang penerapan ditampilkan. Setelah penerapan gagal, Anda mungkin dapat menentukan instans Amazon EC2 mana dan pada langkah mana penerapan gagal.
3. Jika Anda ingin melakukan lebih banyak pemecahan masalah, Anda dapat menggunakan teknik seperti yang dijelaskan di. View Instance Details Anda juga dapat menganalisis file log penyebaran pada instans Amazon EC2. Untuk informasi selengkapnya, lihat Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance.

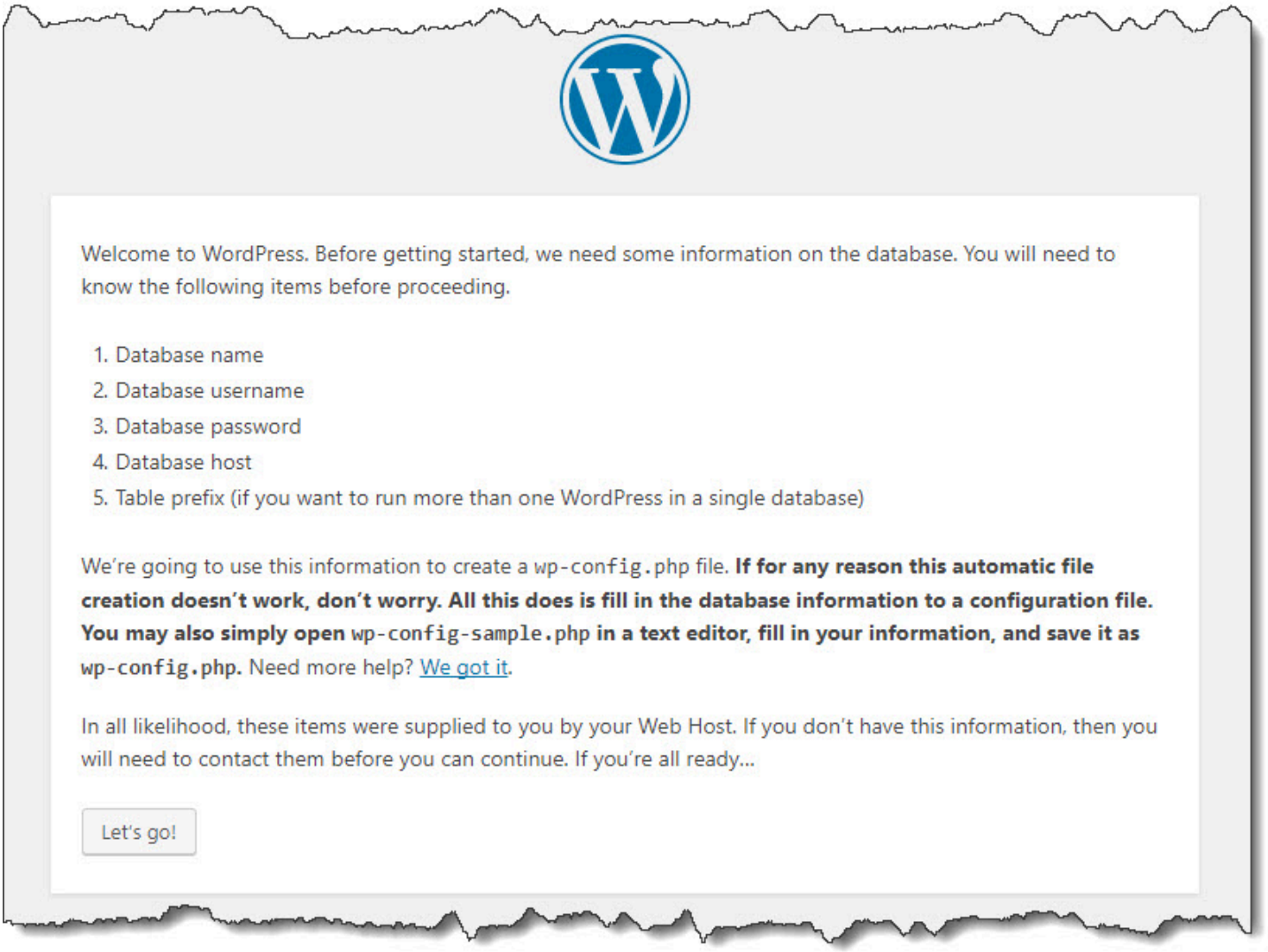
Verifikasi penerapan Anda

Setelah penerapan Anda berhasil, verifikasi WordPress penginstalan Anda berfungsi. Gunakan alamat DNS publik instans Amazon EC2, diikuti `/WordPress` oleh, untuk melihat situs Anda di browser web. (Untuk mendapatkan nilai DNS publik, di konsol Amazon EC2, pilih instans Amazon EC2, dan pada tab Deskripsi, cari nilai DNS Publik.)

Misalnya, jika alamat DNS publik instans **`ec2-01-234-567-890.compute-1.amazonaws.com`** Amazon EC2 Anda, Anda akan menggunakan URL berikut:

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

Ketika Anda melihat situs di browser Anda, Anda akan melihat halaman WordPress selamat datang yang terlihat mirip dengan berikut ini:



Jika instans Amazon EC2 Anda tidak memiliki aturan masuk HTTP yang ditambahkan ke grup keamanannya, maka halaman WordPress selamat datang tidak akan muncul. Jika Anda melihat pesan yang mengatakan server jarak jauh tidak merespons, pastikan grup keamanan untuk instans Amazon EC2 Anda memiliki aturan masuk. Lihat informasi yang lebih lengkap di Tambahkan aturan masuk yang memungkinkan lalu lintas HTTP ke instans Amazon Linux atau RHEL Amazon EC2 Anda.

Langkah 5: Perbarui dan terapkan ulang aplikasi Anda WordPress

Sekarang setelah Anda berhasil menerapkan revisi aplikasi Anda, perbarui WordPress kode pada mesin pengembangan, dan kemudian gunakan CodeDeploy untuk menerapkan ulang situs. Setelah itu, Anda akan melihat perubahan kode pada instans Amazon EC2.

Topik

- Siapkan WordPress situs
- Ubah situs
- Menerapkan ulang situs

Siapkan WordPress situs

Untuk melihat efek perubahan kode, selesaikan pengaturan WordPress situs sehingga Anda memiliki instalasi yang berfungsi penuh.

1. Ketik URL situs Anda ke browser web Anda. URL adalah alamat DNS publik dari instans Amazon EC2 ditambah `/WordPress` ekstensi. Untuk WordPress situs contoh ini (dan contoh alamat DNS publik instans Amazon EC2), URL-nya adalah. **`http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress`**
2. Jika Anda belum menyiapkan situs, halaman selamat datang WordPress default akan muncul. Pilih Ayo pergi! .
3. Untuk menggunakan database MySQL default, pada halaman konfigurasi database, ketikkan nilai berikut:

   - Nama Database: **`test`**
   - Nama pengguna: **`root`**
   - Kata sandi: Biarkan kosong.
   - Host Database: **`localhost`**
   - Awalan Tabel: **`wp_`**

   Pilih Kirim untuk mengatur database.
4. Lanjutkan pengaturan situs. Pada halaman Selamat Datang, isi nilai apa pun yang Anda inginkan, dan pilih Instal WordPress. Setelah instalasi selesai, Anda dapat masuk ke dasbor Anda.

> **Important**
>
> Selama penyebaran WordPress aplikasi, **`change_permissions.sh`** skrip memperbarui izin `/tmp/WordPress` folder sehingga siapa pun dapat menulis ke sana. Sekarang adalah saat

yang tepat untuk menjalankan perintah berikut untuk membatasi izin sehingga hanya Anda, pemilik, yang dapat menulis kepadanya:

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

Ubah situs

Untuk memodifikasi WordPress situs, buka folder aplikasi di mesin pengembangan Anda:

```
cd /tmp/WordPress
```

Untuk memodifikasi beberapa warna situs, dalam `wp-content/themes/twentyfifteen/style.css` file, gunakan editor teks atau sed `#fff` untuk mengubahnya`#768331`.

Di Linux atau sistem lain dengan GNUsed, gunakan:

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

Di macOS, Unix, atau sistem lain dengan sed BSD, gunakan:

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

Menerapkan ulang situs

Sekarang setelah Anda memodifikasi kode situs, gunakan Amazon S3 dan CodeDeploy untuk menyebarkan kembali situs.

Bundel dan unggah perubahan ke Amazon S3, seperti yang dijelaskan dalam. Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip (Saat Anda mengikuti instruksi tersebut, ingatlah bahwa Anda tidak perlu membuat aplikasi.) Berikan revisi baru kunci yang sama seperti before (**`WordPressApp.zip`**). Unggah ke bucket Amazon S3 yang sama yang Anda buat sebelumnya (misalnya,**`amzn-s3-demo-bucket`**).

Gunakan AWS CLI, CodeDeploy konsol, atau CodeDeploy API untuk menerapkan ulang situs.

Topik

- Untuk menyebarkan kembali situs (CLI)
- Untuk menyebarkan kembali situs (konsol)

Untuk menyebarkan kembali situs (CLI)

Panggil create-deployment perintah untuk membuat penerapan berdasarkan revisi yang baru diunggah. Gunakan aplikasi bernama**`WordPress_App`**, konfigurasi penerapan bernama**`CodeDeployDefault.OneAtATime`**, grup penyebaran bernama**`WordPress_DepGroup`**, dan revisi yang disebutkan **`WordPressApp.zip`** dalam bucket bernama: **`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
```

Anda dapat memeriksa status penerapan, seperti yang dijelaskan dalamPantau dan pecahkan masalah penerapan Anda.

CodeDeploy Setelah menempatkan kembali situs, kunjungi kembali situs di browser web Anda untuk memverifikasi warna telah diubah. (Anda mungkin perlu menyegarkan browser Anda.) Jika warnanya telah diubah, selamat! Anda telah berhasil memodifikasi dan memindahkan situs Anda!

Untuk menyebarkan kembali situs (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dalam daftar aplikasi, pilih WordPress_App.
4. Pada tab Grup Deployment, pilih**`WordPress_DepGroup`**.
5. Pilih Buat penerapan.
6. Pada halaman Create deployment:

a. Di grup Deployment, pilih**`WordPress_DepGroup`**.

b. Di area tipe Repositori, pilih Aplikasi saya disimpan di Amazon S3, lalu salin tautan Amazon S3 revisi Anda ke kotak lokasi Revisi. Untuk menemukan nilai tautan:

   i. Di tab browser terpisah:

      Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

      Jelajahi dan buka amzn-s3-demo-bucket, lalu pilih revisi Anda,. **`WordPressApp.zip`**

   ii. Jika panel Properties tidak terlihat di konsol Amazon S3, pilih tombol Properties.

   iii. Di panel Properti, salin nilai bidang Tautan ke kotak Lokasi revisi di CodeDeploy konsol.

c. Jika muncul pesan yang mengatakan jenis file tidak dapat dideteksi, pilih.zip.

d. Biarkan kotak deskripsi Deployment kosong.

e. Perluas penggantian grup Deployment dan dari konfigurasi Deployment, pilih. CodeDeployDefault.OneAtATime

f. Pilih Mulai penerapan. Informasi tentang penerapan yang baru dibuat muncul di halaman Deployment.

g. Anda dapat memeriksa status penerapan, seperti yang dijelaskan dalamPantau dan pecahkan masalah penerapan Anda.

   CodeDeploy Setelah menempatkan kembali situs, kunjungi kembali situs di browser web Anda untuk memverifikasi warna telah diubah. (Anda mungkin perlu menyegarkan browser Anda.) Jika warnanya telah diubah, selamat! Anda telah berhasil memodifikasi dan memindahkan situs Anda!

Langkah 6: Bersihkan WordPress aplikasi Anda dan sumber daya terkait

Anda sekarang telah berhasil membuat pembaruan ke WordPress kode dan menerapkan kembali situs. Untuk menghindari biaya berkelanjutan untuk sumber daya yang Anda buat untuk tutorial ini, Anda harus menghapus:

- Setiap CloudFormation tumpukan (atau menghentikan instans Amazon EC2 apa pun, jika Anda membuatnya di luar). CloudFormation
- Ember Amazon S3 apa pun.

- `WordPress_App`Aplikasi di CodeDeploy.
- Asosiasi Manajer AWS Systems Manager Negara untuk CodeDeploy agen.

Anda dapat menggunakan AWS CLI, Amazon S3 CloudFormation, Amazon EC2, CodeDeploy dan konsol, atau API untuk melakukan AWS pembersihan.

Topik

- Untuk membersihkan sumber daya (CLI)
- Untuk membersihkan sumber daya (konsol)
- Apa selanjutnya?

Untuk membersihkan sumber daya (CLI)

1. Jika Anda menggunakan CloudFormation template kami untuk tutorial ini, panggil delete-stack perintah terhadap tumpukan bernama**`CodeDeployDemoStack`**. Ini akan menghentikan semua instans Amazon EC2 yang menyertainya dan menghapus semua peran IAM yang menyertainya yang dibuat oleh tumpukan:

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

2. Untuk menghapus bucket Amazon S3, panggil rm perintah dengan --recursive sakelar terhadap bucket bernama. **`amzn-s3-demo-bucket`** Ini akan menghapus ember dan semua benda di ember:

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

3. Untuk menghapus `WordPress_App` aplikasi, panggil delete-application perintah. Ini juga akan menghapus semua catatan grup penyebaran terkait dan catatan penyebaran untuk aplikasi:

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

4. Untuk menghapus asosiasi Manajer Negara Systems Manager, panggil delete-association perintah.

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

   Anda bisa mendapatkan *`association-id`* dengan memanggil describe-association perintah.

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

Jika Anda tidak menggunakan CloudFormation tumpukan untuk tutorial ini, panggil terminate-instances perintah untuk menghentikan instans Amazon EC2 yang Anda buat secara manual. Berikan ID instans Amazon EC2 untuk mengakhiri:

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

Untuk membersihkan sumber daya (konsol)

Jika Anda menggunakan CloudFormation template kami untuk tutorial ini, hapus CloudFormation tumpukan terkait.

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di kotak Filter, ketik nama CloudFormation tumpukan yang Anda buat sebelumnya (misalnya,**`CodeDeployDemoStack`**).
3. Pilih kotak di samping nama tumpukan. Di menu Actions, pilih Delete Stack.

   CloudFormation menghapus tumpukan, menghentikan semua instans Amazon EC2 yang menyertainya, dan menghapus semua peran IAM yang menyertainya.

Untuk menghentikan instans Amazon EC2 yang Anda buat di luar tumpukan: CloudFormation

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Dalam daftar INSTANCES, pilih Instans.
3. Di kotak pencarian, ketik nama instans Amazon EC2 yang ingin Anda akhiri (misalnya,**`CodeDeployDemo`**), lalu tekan Enter.
4. Pilih nama instans Amazon EC2.
5. Di menu Actions, arahkan ke Instance State, lalu pilih Terminate. Saat diminta, pilih Ya, Hentikan.

Ulangi langkah-langkah ini untuk setiap contoh.

Untuk menghapus bucket Amazon S3:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dalam daftar bucket, telusuri dan pilih nama bucket Amazon S3 yang Anda buat sebelumnya (misalnya**`amzn-s3-demo-bucket`**,).
3. Sebelum Anda dapat menghapus ember, Anda harus terlebih dahulu menghapus isinya. Pilih semua file dalam ember, seperti**`WordPressApp.zip`**. Di menu Tindakan, pilih Hapus. Saat diminta untuk mengonfirmasi penghapusan, pilih OK.
4. Setelah ember kosong, Anda dapat menghapus ember. Dalam daftar ember, pilih baris ember (tetapi bukan nama ember). Pilih Hapus bucket, dan ketika diminta untuk mengonfirmasi, pilih OK.

Untuk menghapus `WordPress_App` aplikasi dari CodeDeploy:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dalam daftar aplikasi, pilih WordPress_App.
4. Pada halaman Detail aplikasi, pilih Hapus aplikasi.
5. Saat diminta, masukkan nama aplikasi untuk mengonfirmasi bahwa Anda ingin menghapusnya, lalu pilih Hapus.

Untuk menghapus asosiasi Manajer Negara Systems Manager:

1. Buka AWS Systems Manager konsol di https://console.aws.amazon.com/systems-manager.
2. Di panel navigasi, pilih State Manager.
3. Pilih asosiasi yang Anda buat dan pilih Hapus.

Apa selanjutnya?

Jika Anda sudah tiba di sini, selamat! Anda telah berhasil menyelesaikan CodeDeploy penerapan, dan kemudian memperbarui kode situs Anda dan menerapkannya kembali.

Tutorial: Menyebarkan “halo, dunia!” aplikasi dengan CodeDeploy (Windows Server)

Dalam tutorial ini, Anda menyebarkan satu halaman web ke satu instans Windows Server Amazon EC2 yang menjalankan Internet Information Services (IIS) sebagai server webnya. Halaman web ini menampilkan “Halo, Dunia!” sederhana pesan.

Bukan apa yang Anda cari?

- Untuk mempraktikkan penerapan ke instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux (RHEL) Amazon EC2, lihat. Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix)
- Untuk mempraktikkan penerapan ke instans lokal, lihat. Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)

Langkah-langkah tutorial ini disajikan dari perspektif Windows. Meskipun Anda dapat menyelesaikan sebagian besar langkah-langkah ini pada mesin lokal yang menjalankan Linux, macOS, atau Unix, Anda harus menyesuaikan langkah-langkah yang mencakup jalur Windows-based direktori seperti. `c:\temp` Selain itu, jika Anda ingin terhubung ke instans Amazon EC2, Anda memerlukan aplikasi klien yang dapat terhubung melalui Remote Desktop Protocol (RDP) ke instans Amazon EC2 yang menjalankan Windows Server. (Windows menyertakan aplikasi klien koneksi RDP secara default.)

Sebelum Anda memulai tutorial ini, Anda harus menyelesaikan prasyaratMemulai dengan CodeDeploy, termasuk mengonfigurasi pengguna Anda, menginstal atau meningkatkan AWS CLI, dan membuat profil instans IAM dan peran layanan.

Topik

- Langkah 1: Luncurkan instans Windows Server Amazon EC2
- Langkah 2: Konfigurasikan konten sumber Anda untuk menyebarkan ke instans Windows Server Amazon EC2
- Langkah 3: Unggah “halo, dunia!” aplikasi ke Amazon S3

- Langkah 4: Menyebarkan aplikasi Hello World Anda
- Langkah 5: Perbarui dan gunakan kembali “halo, dunia!” application
- Langkah 6: Bersihkan “halo, dunia!” aplikasi dan sumber daya terkait

Langkah 1: Luncurkan instans Windows Server Amazon EC2

Untuk menggunakan aplikasi Hello World CodeDeploy, Anda memerlukan instans Amazon EC2 yang menjalankan Windows Server.

Ikuti petunjuk dalam Buat instans Amazon EC2 untuk CodeDeploy. Saat Anda siap menetapkan tag instans Amazon EC2 ke instance, pastikan untuk menentukan kunci tag dan **`Name`** nilai tag. **`CodeDeployDemo`** (Jika Anda menentukan kunci tag atau nilai tag yang berbeda, maka instruksi di Langkah 4: Menyebarkan aplikasi Hello World Anda mungkin menghasilkan hasil yang tidak terduga.)

Setelah meluncurkan instans Amazon EC2, kembali ke halaman ini, dan lanjutkan ke bagian berikutnya. Jangan melanjutkan ke Buat aplikasi dengan CodeDeploy langkah berikutnya.

Connect ke instans Amazon EC2

Setelah instans Amazon EC2 diluncurkan, ikuti petunjuk ini untuk berlatih menghubungkannya.

> **Note**
>
> Dalam petunjuk ini, kami menganggap Anda menjalankan Windows dan aplikasi klien Koneksi Desktop Windows. Untuk selengkapnya, lihat Menyambungkan ke instans Windows Anda menggunakan RDP. Anda mungkin perlu menyesuaikan instruksi ini untuk sistem operasi lain atau aplikasi klien koneksi RDP lainnya.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi, di Instans, pilih Instans.
3. Jelajahi dan pilih instance Windows Server Anda dalam daftar.
4. Pilih Hubungkan.
5. Pilih Dapatkan Kata Sandi, lalu pilih Pilih File.
6. Jelajahi dan pilih file key pair instans Amazon EC2 yang terkait dengan instans Windows Server Amazon EC2, lalu pilih Buka.

7. Pilih Dekripsi Kata Sandi. Catat kata sandi yang ditampilkan. Anda membutuhkannya di langkah 10.
8. Pilih Unduh File Desktop Jarak Jauh, lalu buka file tersebut.
9. Jika Anda diminta untuk terhubung meskipun penerbit koneksi jarak jauh tidak dapat diidentifikasi, lanjutkan.
10. Ketik kata sandi yang Anda catat di langkah 7, lalu lanjutkan. (Jika aplikasi klien koneksi RDP Anda meminta Anda untuk nama pengguna, ketik.) **`Administrator`**
11. Jika Anda diminta untuk terhubung meskipun identitas komputer jarak jauh tidak dapat diverifikasi, lanjutkan.
12. Setelah Anda terhubung, desktop instans Amazon EC2 yang menjalankan Windows Server ditampilkan.
13. Anda sekarang dapat memutuskan sambungan dari instans Amazon EC2.

> **Warning**
>
> Jangan menghentikan atau menghentikan instance. Jika tidak, tidak CodeDeploy dapat menerapkannya.

Tambahkan aturan masuk yang memungkinkan lalu lintas HTTP ke instans Amazon EC2 Windows Server Anda

Langkah selanjutnya mengonfirmasi instans Amazon EC2 Anda memiliki port HTTP terbuka sehingga Anda dapat melihat halaman web yang digunakan pada instans Amazon EC2 Windows Server Anda di browser.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Pilih Instans, lalu pilih instans Anda.
3. Pada tab Deskripsi, di bawah Grup keamanan, pilih lihat aturan masuk.

   Anda akan melihat daftar aturan di grup keamanan Anda seperti berikut:

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

4. Di bawah Grup keamanan, pilih grup keamanan untuk instans Amazon EC2 Anda. Itu bisa diberi nama**launch-wizard-*N***. ***N***Dalam nama adalah nomor yang ditetapkan ke grup keamanan Anda saat instance Anda dibuat.

   Pilih tab Masuk. Jika grup keamanan untuk instans Anda dikonfigurasi dengan benar, Anda akan melihat aturan dengan nilai-nilai berikut:

   - Jenis: HTTP
   - Protokol: TCP
   - Rentang Port: 80
   - Sumber: 0.0.0. 0/0

5. Jika Anda tidak melihat aturan dengan nilai-nilai ini, gunakan prosedur dalam Menambahkan Aturan ke Grup Keamanan untuk menambahkannya ke aturan keamanan baru.

Langkah 2: Konfigurasikan konten sumber Anda untuk menyebarkan ke instans Windows Server Amazon EC2

Sekarang saatnya untuk mengonfigurasi konten sumber aplikasi Anda sehingga Anda memiliki sesuatu yang dapat Anda terapkan ke instans Amazon EC2. Untuk tutorial ini, Anda akan menyebarkan satu halaman web ke instans Amazon EC2 yang menjalankan Windows Server, yang akan menjalankan Internet Information Services (IIS) sebagai server webnya. Halaman web ini akan menampilkan “Halo, Dunia!” sederhana pesan.

Topik

- Buat halaman web
- Buat skrip untuk menjalankan aplikasi Anda
- Tambahkan file spesifikasi aplikasi

Buat halaman web

1. Buat subdirektori (subfolder) bernama `HelloWorldApp` di `c:\temp` folder Anda, dan kemudian beralih ke folder itu.

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

> **Note**
>
> Anda tidak harus menggunakan lokasi `c:\temp` atau nama subfolder. `HelloWorldApp` Jika Anda menggunakan lokasi atau nama subfolder yang berbeda, pastikan untuk menggunakannya di seluruh tutorial ini.

2. Gunakan editor teks untuk membuat file di dalam folder. Beri nama file`index.html`.

```
notepad index.html
```

3. Tambahkan kode HTML berikut ke file, lalu simpan file tersebut.

```
<!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>
```

Buat skrip untuk menjalankan aplikasi Anda

Selanjutnya, Anda akan membuat skrip yang CodeDeploy akan digunakan untuk mengatur server web pada instans Amazon EC2 target.

1. Di subfolder yang sama tempat `index.html` file disimpan, gunakan editor teks untuk membuat file lain. Beri nama file`before-install.bat`.

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

2. Tambahkan kode skrip batch berikut ke file, lalu simpan file.

   ```
   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
   ```

Tambahkan file spesifikasi aplikasi

Selanjutnya, Anda akan menambahkan file spesifikasi aplikasi (AppSpec file) selain halaman web dan file skrip batch. AppSpec File tersebut adalah file berformat YAMAL yang digunakan oleh untuk: CodeDeploy

- Petakan file sumber dalam revisi aplikasi Anda ke tujuan mereka pada instance.
- Tentukan skrip yang akan dijalankan pada instance selama penerapan.

AppSpec File harus diberi nama`appspec.yml`. Itu harus ditempatkan di folder root kode sumber aplikasi.

1. Di subfolder yang sama tempat `before-install.bat` file `index.html` dan disimpan, gunakan editor teks untuk membuat file lain. Beri nama file`appspec.yml`.

   ```
   notepad appspec.yml
   ```

2. Tambahkan kode YAMAL berikut ke file, lalu simpan file tersebut.

   ```
   version: 0.0
   os: windows
   ```

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

CodeDeploy akan menggunakan AppSpec file ini untuk menyalin `index.html` file di folder root kode sumber aplikasi ke `c:\inetpub\wwwroot` folder pada instans Amazon EC2 target. Selama penerapan, CodeDeploy akan menjalankan skrip `before-install.bat` batch pada instans Amazon EC2 target selama peristiwa siklus hidup **`BeforeInstall`** penerapan. Jika skrip ini membutuhkan waktu lebih dari 900 detik (15 menit) untuk dijalankan, CodeDeploy akan menghentikan penerapan dan menandai penerapan ke instans Amazon EC2 sebagai gagal.

Untuk informasi selengkapnya tentang pengaturan ini, lihatCodeDeploy AppSpec referensi file.

> **Important**
>
> Lokasi dan jumlah spasi antara masing-masing item dalam file ini penting. Jika spasi salah, CodeDeploy akan memunculkan kesalahan yang mungkin sulit untuk di-debug. Lihat informasi yang lebih lengkap di AppSpec Spasi file.

Langkah 3: Unggah “halo, dunia!” aplikasi ke Amazon S3

Sekarang Anda akan mempersiapkan dan mengunggah konten sumber Anda ke lokasi yang CodeDeploy dapat menyebarkannya. Petunjuk berikut menunjukkan cara menyediakan bucket Amazon S3, menyiapkan file revisi aplikasi untuk bucket, menggabungkan file revisi, lalu mendorong revisi ke bucket.

> **Note**
>
> Meskipun tidak tercakup dalam tutorial ini, Anda dapat menggunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub repositori ke instance. Untuk informasi selengkapnya, lihat Integrasi dengan CodeDeploy GitHub.

Topik

- Menyediakan bucket Amazon S3
- Siapkan file aplikasi untuk ember
- Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip

Menyediakan bucket Amazon S3

Buat wadah penyimpanan atau bucket di Amazon S3—atau gunakan bucket yang sudah ada. Pastikan Anda dapat mengunggah revisi ke bucket dan instans Amazon EC2 yang digunakan dalam penerapan dapat mengunduh revisi dari bucket.

Anda dapat menggunakan AWS CLI, konsol Amazon S3, atau API Amazon S3 untuk membuat bucket Amazon S3. Setelah Anda membuat bucket, pastikan untuk memberikan izin akses ke bucket dan CodeDeploy pengguna Anda.

> **Note**
>
> Nama bucket harus unik di Amazon S3 untuk semua AWS akun. Jika Anda tidak dapat menggunakannya**`amzn-s3-demo-bucket`**, coba nama bucket yang berbeda, seperti **`amzn-s3-demo-bucket`** diikuti dengan tanda hubung dan inisiasi Anda atau pengenal unik lainnya. Kemudian pastikan untuk mengganti nama bucket Anda ke **`amzn-s3-demo-bucket`** mana pun Anda melihatnya di seluruh tutorial ini.
> Bucket Amazon S3 harus dibuat di AWS Wilayah yang sama tempat instans Amazon EC2 target Anda diluncurkan. Misalnya, jika Anda membuat bucket di Wilayah AS Timur (Virginia N.), instans Amazon EC2 target Anda harus diluncurkan di Wilayah AS Timur (Virginia Utara).

Topik

- Untuk membuat bucket Amazon S3 (CLI)
- Untuk membuat bucket Amazon S3 (konsol)
- Berikan izin ke bucket Amazon S3 dan Anda AWS akun

Untuk membuat bucket Amazon S3 (CLI)

Panggil mb perintah untuk membuat bucket Amazon S3 bernama: **`amzn-s3-demo-bucket`**

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

Untuk membuat bucket Amazon S3 (konsol)

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di konsol Amazon S3, pilih Buat ember.
3. Di kotak nama Bucket, ketikkan nama untuk bucket.
4. Dalam daftar Wilayah, pilih Wilayah target, lalu pilih Buat.

Berikan izin ke bucket Amazon S3 dan Anda AWS akun

Anda harus memiliki izin untuk mengunggah ke bucket Amazon S3. Anda dapat menentukan izin ini melalui kebijakan bucket Amazon S3. Misalnya, dalam kebijakan bucket Amazon S3 berikut, menggunakan karakter wildcard (*) memungkinkan AWS akun `111122223333` untuk mengunggah file ke direktori apa pun di bucket Amazon S3 bernama: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Untuk melihat ID AWS akun, lihat Menemukan ID AWS akun Anda.

Sekarang adalah saat yang tepat untuk memverifikasi bucket Amazon S3 akan memungkinkan permintaan unduhan dari setiap instans Amazon EC2 yang berpartisipasi. Anda dapat menentukan ini melalui kebijakan bucket Amazon S3. Misalnya, dalam kebijakan bucket Amazon S3 berikut, menggunakan karakter wildcard (*) memungkinkan instans Amazon EC2 dengan profil instans IAM terlampir yang berisi ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` untuk mengunduh file dari direktori apa pun di bucket Amazon S3 bernama: `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"
                ]
            }
        }
    ]
}
```

Untuk informasi tentang cara membuat dan melampirkan kebijakan bucket Amazon S3, lihat contoh kebijakan Bucket.

Pengguna CodeDeploy admin yang Anda buat juga Langkah 1: Menyiapkan harus memiliki izin untuk mengunggah revisi ke bucket Amazon S3. Salah satu cara untuk menentukan ini adalah melalui kebijakan IAM, yang Anda tambahkan ke set izin pengguna, atau ke peran IAM (yang Anda izinkan untuk diasumsikan oleh pengguna). Kebijakan IAM berikut memungkinkan pengguna untuk mengunggah revisi di mana saja di bucket Amazon S3 bernama: `amzn-s3-demo-bucket`

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":["s3:PutObject"],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

Untuk informasi tentang cara membuat kebijakan IAM, lihat Membuat kebijakan IAM di Panduan Pengguna IAM. Untuk informasi tentang menambahkan kebijakan ke set izin, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

Siapkan file aplikasi untuk ember

Pastikan halaman web, AppSpec file, dan skrip diatur pada mesin pengembangan Anda seperti ini:

```
c:\
  |-- temp\
        |--HelloWorldApp\
             |-- appspec.yml
             |-- before-install.bat
             |-- index.html
```

Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip

Bundel file ke dalam file arsip (dikenal sebagai revisi aplikasi).

> **Note**
>
> Anda mungkin dikenakan biaya untuk menyimpan objek dalam ember dan untuk mentransfer revisi aplikasi ke dalam dan keluar dari ember. Untuk informasi selengkapnya, lihat Harga Amazon S3.

1. Pada mesin pengembangan, beralih ke folder tempat file disimpan:

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

   > **Note**
   >
   > Jika Anda tidak beralih ke folder ini, maka bundling file akan dimulai di folder Anda saat ini. Misalnya, jika folder Anda saat ini `c:\temp` bukan`c:\temp\HelloWorldApp`, bundling akan dimulai dengan file dan subfolder di `c:\temp` folder, yang mungkin mencakup lebih dari subfolder`HelloWorldApp`.

2. Panggil create-application perintah untuk mendaftarkan aplikasi baru bernama **`HelloWorld_App`** dengan CodeDeploy:

```
aws deploy create-application --application-name HelloWorld_App
```

3. Panggil perintah CodeDeploy push untuk menggabungkan file bersama-sama, mengunggah revisi ke Amazon S3, dan mendaftarkan informasi tentang revisi CodeDeploy yang diunggah, semuanya dalam satu tindakan.

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

   Perintah ini menggabungkan file dari direktori saat ini (tidak termasuk file tersembunyi) ke dalam satu file arsip bernama`HelloWorld_App.zip`, mengunggah revisi ke **`amzn-s3-demo-bucket`** bucket, dan mendaftarkan informasi tentang revisi yang CodeDeploy diunggah.

Langkah 4: Menyebarkan aplikasi Hello World Anda

Sekarang Anda menerapkan contoh revisi aplikasi Hello World yang Anda unggah ke Amazon S3. Anda menggunakan AWS CLI atau CodeDeploy konsol untuk menerapkan revisi dan memantau kemajuan penerapan. Setelah revisi aplikasi berhasil digunakan, Anda memeriksa hasilnya.

Topik

- Terapkan revisi aplikasi Anda dengan CodeDeploy
- Memantau dan memecahkan masalah penyebaran Anda
- Verifikasi penerapan Anda

Terapkan revisi aplikasi Anda dengan CodeDeploy

Anda dapat menerapkan aplikasi Anda menggunakan CLI atau konsol.

Topik

- Untuk menerapkan revisi aplikasi Anda (CLI)
- Untuk menerapkan revisi aplikasi Anda (konsol)

Untuk menerapkan revisi aplikasi Anda (CLI)

1. Pertama, penyebaran membutuhkan grup penyebaran. Namun, sebelum Anda membuat grup penyebaran, Anda memerlukan peran layanan ARN. Peran layanan adalah peran IAM

yang memberikan izin layanan untuk bertindak atas nama Anda. Dalam hal ini, peran layanan memberikan CodeDeploy izin untuk mengakses instans Amazon EC2 Anda untuk memperluas (membaca) tag instans Amazon EC2 mereka.

Anda seharusnya sudah mengikuti instruksi Buat peran layanan (CLI) untuk membuat peran layanan. Untuk mendapatkan ARN dari peran layanan, lihat. Dapatkan peran layanan ARN (CLI)

2. Sekarang setelah Anda memiliki ARN, panggil create-deployment-group perintah untuk membuat grup penyebaran bernama**`HelloWorld_DepGroup`**, terkait dengan aplikasi bernama**`HelloWorld_App`**, menggunakan tag instans Amazon EC2 bernama **`CodeDeployDemo`** dan konfigurasi penerapan bernama, **`CodeDeployDefault.OneAtATime`** dengan peran layanan ARN:

```
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**
>
> Perintah create-deployment-group menyediakan dukungan untuk membuat pemicu yang menghasilkan pengiriman notifikasi Amazon SNS ke pelanggan topik tentang peristiwa tertentu dalam penerapan dan instance. Perintah ini juga mendukung opsi untuk memutar kembali penerapan secara otomatis dan mengatur alarm untuk menghentikan penerapan saat memantau ambang batas di alarm Amazon terpenuhi. CloudWatch Perintah untuk tindakan ini tidak termasuk dalam tutorial ini.

3. Sebelum Anda membuat penerapan, instance dalam grup penyebaran Anda harus memiliki agen yang diinstal. CodeDeploy Anda dapat menginstal agen dari baris perintah AWS Systems Manager dengan perintah berikut:

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

Perintah ini menciptakan asosiasi di Systems Manager State Manager yang akan menginstal CodeDeploy agen dan kemudian mencoba memperbaruinya pada pukul 2:00 setiap Minggu pagi. Untuk informasi lebih lanjut tentang CodeDeploy agen, lihat Bekerja dengan CodeDeploy

agen. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu AWS Systems Manager.

4. Sekarang panggil create-deployment perintah untuk membuat deployment yang terkait dengan aplikasi bernama**`HelloWorld_App`**, konfigurasi deployment bernama**`CodeDeployDefault.OneAtATime`**, dan grup deployment bernama**`HelloWorld_DepGroup`**, menggunakan revisi aplikasi yang dinamai **`HelloWorld_App.zip`** dalam bucket bernama: **`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
```

Untuk menerapkan revisi aplikasi Anda (konsol)

1. Sebelum Anda menggunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda, Anda memerlukan peran layanan ARN. Peran layanan adalah peran IAM yang memberikan izin layanan untuk bertindak atas nama Anda. Dalam hal ini, peran layanan memberikan CodeDeploy izin untuk mengakses instans Amazon EC2 Anda untuk memperluas (membaca) tag instans Amazon EC2 mereka.

   Anda seharusnya sudah mengikuti instruksi Buat peran layanan (konsol) untuk membuat peran layanan. Untuk mendapatkan ARN dari peran layanan, lihat. Dapatkan peran layanan ARN (konsol)
2. Sekarang Anda memiliki ARN, Anda dapat menggunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda.

   Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

3. Di panel navigasi, perluas Deploy, lalu pilih Applications.
4. Pilih HelloWorld_App.
5. Pada tab Deployment groups, pilih Create deployment group.

6. Dalam nama grup Deployment, masukkan**`HelloWorld_DepGroup`**.
7. Di Peran Layanan, pilih nama peran layanan.
8. Dalam jenis Deployment, pilih In-place.
9. Dalam konfigurasi Lingkungan, pilih instans Amazon EC2.
10. Dalam konfigurasi Agen dengan AWS Systems Manager, pertahankan defaultnya.
11. Di Key, masukkan**`Name`**.
12. Dalam Value (Nilai), masukkan **`CodeDeployDemo`**.
13. Dalam konfigurasi Deployment, pilih CodeDeployDefault.OneAtATime.
14. Di Load Balancer, hapus Aktifkan penyeimbangan beban.
15. Pilih Buat grup penyebaran.
16. Pilih Buat penerapan.
17. Di grup Deployment, pilih _ HelloWorld DepGroup
18. Pada tipe Revisi, pilih Aplikasi saya disimpan di Amazon S3, lalu di lokasi Revisi, masukkan lokasi contoh revisi aplikasi Hello World yang sebelumnya Anda unggah ke Amazon S3. Untuk mendapatkan lokasi:
    a. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
    b. Dalam daftar bucket, pilih amzn-s3-demo-bucket (atau nama bucket tempat Anda mengunggah revisi aplikasi).
    c. Dalam daftar objek, pilih HelloWorld_ App.zip.
    d. Di tab Ikhtisar, pilih Salin jalur.
    e. Kembali ke CodeDeploy konsol, dan di Lokasi Revisi, tempel nilai bidang Tautan.
19. Untuk jenis file Revisi, pilih.zip.
20. (Opsional) Masukkan komentar di Deskripsi Deployment.
21. Pilih Buat penerapan. Informasi tentang penerapan yang baru dibuat muncul di halaman Deployment.

Memantau dan memecahkan masalah penyebaran Anda

Gunakan AWS CLI atau konsol untuk memantau dan memecahkan masalah penyebaran Anda.

Topik

- Untuk memantau dan memecahkan masalah penerapan Anda (CLI)
- Untuk memantau dan memecahkan masalah penerapan Anda (konsol)

Untuk memantau dan memecahkan masalah penerapan Anda (CLI)

1. Dapatkan ID penerapan dengan memanggil list-deployments perintah terhadap aplikasi bernama **`HelloWorld_App`** dan grup penyebaran bernama: **`HelloWorld_DepGroup`**

   ```
   aws deploy list-deployments --application-name HelloWorld_App --deployment-group-
   name HelloWorld_DepGroup --query "deployments" --output text
   ```

2. Panggil get-deployment perintah dengan ID penyebaran:

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

3. Perintah mengembalikan status keseluruhan penerapan. Jika berhasil, nilainya`Succeeded`.

   Jika status keseluruhannya`Failed`, Anda dapat memanggil perintah seperti list-deployment-instance dan get-deployment-instance untuk memecahkan masalah. Untuk opsi pemecahan masalah lainnya, lihat. Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance

Untuk memantau dan memecahkan masalah penerapan Anda (konsol)

Pada halaman Deployment di CodeDeploy konsol, Anda dapat memantau status penerapan di kolom Status.

Untuk mendapatkan informasi selengkapnya tentang penerapan Anda, terutama jika nilai kolom Status memiliki nilai selain Berhasil:

1. Di tabel Deployment, pilih ID deployment Anda. Setelah penerapan gagal, pesan yang menjelaskan alasan kegagalan muncul di halaman detail penerapan.
2. . Informasi lebih lanjut tentang instance penerapan ditampilkan. Setelah penerapan gagal, Anda mungkin dapat menentukan instans Amazon EC2 mana dan pada langkah mana penerapan gagal.
3. Jika Anda ingin melakukan lebih banyak pemecahan masalah, Anda dapat menggunakan teknik seperti. View Instance Details Anda juga dapat menganalisis file log penyebaran pada

instans Amazon EC2. Untuk informasi selengkapnya, lihat Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance.

Verifikasi penerapan Anda

Setelah penerapan Anda berhasil, verifikasi instalasi Anda berfungsi. Gunakan alamat DNS publik instans Amazon EC2 untuk melihat halaman web di browser web. (Untuk mendapatkan nilai DNS publik, di konsol Amazon EC2, pilih instans Amazon EC2, dan pada tab Deskripsi, cari nilainya di DNS Publik.)

Misalnya, jika alamat DNS publik instans **`ec2-01-234-567-890.compute-1.amazonaws.com`** Amazon EC2 Anda, Anda akan menggunakan URL berikut:

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

Jika berhasil, Anda harus melihat halaman web Hello World.

Langkah 5: Perbarui dan gunakan kembali “halo, dunia!” application

Sekarang Anda telah berhasil menerapkan revisi aplikasi Anda, pada mesin pengembangan, buat pembaruan ke kode halaman web, dan kemudian gunakan CodeDeploy untuk menyebarkan ulang situs. Setelah pemindahan, Anda akan dapat melihat perubahan pada instans Amazon EC2.

Topik

- Ubah halaman web
- Menerapkan ulang situs

Ubah halaman web

1. Buka `c:\temp\HelloWorldApp` subfolder Anda dan gunakan editor teks untuk memodifikasi `index.html` file:

   ```
   cd c:\temp\HelloWorldApp
   notepad index.html
   ```

2. Merevisi isi `index.html` file untuk mengubah warna latar belakang dan beberapa teks pada halaman web, dan kemudian menyimpan file:

```
<!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>
```

Menerapkan ulang situs

Sekarang setelah Anda memodifikasi kode, gunakan Amazon S3 dan CodeDeploy untuk menerapkan ulang halaman web.

Bundel dan unggah perubahan ke Amazon S3 seperti yang dijelaskan dalam. Bundel file aplikasi ke dalam satu file arsip dan dorong file arsip (Saat Anda mengikuti instruksi tersebut, Anda tidak perlu membuat aplikasi baru.) Berikan revisi kunci yang sama seperti sebelumnya (**`HelloWorld_App.zip`**). Unggah ke bucket Amazon S3 yang sama yang Anda buat sebelumnya (misalnya,**`amzn-s3-demo-bucket`**).

Gunakan AWS CLI atau CodeDeploy konsol untuk menyebarkan kembali situs.

Topik

- Untuk menyebarkan kembali situs (CLI)

- Untuk menyebarkan kembali situs (konsol)

Untuk menyebarkan kembali situs (CLI)

Panggil create-deployment perintah untuk membuat penerapan berdasarkan revisi yang diunggah, sekali lagi menggunakan aplikasi bernama**HelloWorld_App**, konfigurasi penerapan bernama, grup penyebaran bernama **CodeDeployDefault.OneAtATimeHelloWorld_DepGroup**, dan revisi yang disebutkan dalam bucket bernama: **HelloWorld_App.zip 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
```

Anda dapat memeriksa status penerapan baru, seperti yang dijelaskan dalamMemantau dan memecahkan masalah penyebaran Anda.

Ketika CodeDeploy telah memindahkan situs, kunjungi kembali situs di browser web Anda untuk memverifikasi bahwa warna latar belakang dan teks pada halaman web telah diubah. (Anda mungkin perlu menyegarkan browser Anda.) Jika warna latar belakang dan teks telah diubah, maka selamat! Anda telah memodifikasi dan memindahkan situs Anda!

Untuk menyebarkan kembali situs (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Pada panel navigasi, pilih Aplikasi.
3. Dalam daftar Aplikasi, pilih HelloWorld_App.
4. Di tab Deployment, pilih Create deployment.

   a. Dalam daftar grup Deployment, pilih HelloWorld_ DepGroup.

   b. Di lokasi Revisi, masukkan tautan Amazon S3 untuk revisi Anda.

      Untuk menemukan nilai tautan:

i. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Jelajahi dan buka amzn-s3-demo-bucket, lalu pilih revisi Anda, **`HelloWorld_App.zip`** di konsol Amazon S3.

ii. Jika panel Properties tidak terlihat di konsol Amazon S3, pilih tombol Properties.

iii. Di panel Properties, salin nilai bidang Link.

iv. Kembali ke CodeDeploy konsol, lalu tempel tautan ke lokasi Revisi.

v.

c. Dalam jenis file revisi, jika muncul pesan yang menyatakan bahwa jenis file tidak dapat dideteksi, pilih.zip.

d. Biarkan deskripsi Deployment kosong.

e. Perluas penggantian grup Deployment Dalam daftar konfigurasi Deployment, pilih CodeDeployDefault.OneAtATime, lalu pilih Create deployment.

   Anda dapat memeriksa status penerapan seperti yang dijelaskan dalamMemantau dan memecahkan masalah penyebaran Anda.

   Ketika CodeDeploy telah memindahkan situs, kunjungi kembali situs di browser web Anda untuk memverifikasi bahwa warna latar belakang dan teks pada halaman web telah diubah. (Anda mungkin perlu menyegarkan browser Anda.) Jika warna latar belakang dan teks telah diubah, selamat! Anda telah memodifikasi dan memindahkan situs Anda!

Langkah 6: Bersihkan “halo, dunia!” aplikasi dan sumber daya terkait

Anda sekarang telah berhasil membuat pembaruan ke “Hello, World!” kode dan digunakan kembali situs. Untuk menghindari biaya berkelanjutan untuk sumber daya yang Anda buat untuk menyelesaikan tutorial ini, Anda harus menghapus:

- Setiap CloudFormation tumpukan (atau menghentikan instans Amazon EC2 apa pun, jika Anda membuatnya di luar). CloudFormation
- Ember Amazon S3 apa pun.
- `HelloWorld_App`Aplikasi di CodeDeploy.
- Asosiasi Manajer AWS Systems Manager Negara untuk CodeDeploy agen.

Anda dapat menggunakan AWS CLI, Amazon S3 CloudFormation, Amazon EC2, CodeDeploy dan konsol, atau API untuk melakukan AWS pembersihan.

Topik

- Untuk menggunakan sumber daya pembersih (CLI)
- Untuk membersihkan sumber daya (konsol)
- Apa selanjutnya?

Untuk menggunakan sumber daya pembersih (CLI)

1. Jika Anda menggunakan CloudFormation tumpukan untuk tutorial ini, hapus tumpukan dengan memanggil delete-stack perintah terhadap tumpukan bernama**`CodeDeployDemoStack`**. Ini mengakhiri semua instans Amazon EC2 yang menyertainya dan menghapus semua peran IAM yang menyertainya yang awalnya dibuat oleh tumpukan.

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

2. Untuk menghapus bucket Amazon S3, panggil rm perintah dengan --recursive sakelar terhadap bucket bernama. **`amzn-s3-demo-bucket`** Ini menghapus ember dan semua benda di ember.

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

3. Untuk menghapus `HelloWorld_App` aplikasi dari CodeDeploy, panggil delete-application perintah. Ini menghapus semua catatan grup penyebaran terkait dan catatan penyebaran untuk aplikasi.

   ```
   aws deploy delete-application --application-name HelloWorld_App
   ```

4. Untuk menghapus asosiasi Manajer Negara Systems Manager, panggil delete-association perintah.

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

   Anda bisa mendapatkan *`association-id`* dengan memanggil describe-association perintah.

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

5. Jika Anda tidak menggunakan CloudFormation tumpukan untuk tutorial ini, panggil terminate-instances perintah untuk menghentikan instans Amazon EC2 yang Anda buat secara manual. Berikan ID instans Amazon EC2 untuk dihentikan.

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

Untuk membersihkan sumber daya (konsol)

Jika Anda menggunakan CloudFormation template kami untuk tutorial ini, hapus CloudFormation tumpukan terkait.

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di kotak pencarian, ketik nama CloudFormation tumpukan (misalnya,**CodeDeployDemoStack**).
3. Pilih kotak di samping nama tumpukan.
4. Di menu Actions, pilih Delete Stack. Ini menghapus tumpukan, menghentikan semua instans Amazon EC2 yang menyertainya, dan menghapus semua peran IAM yang menyertainya.

Untuk menghentikan instans Amazon EC2 yang Anda buat di luar tumpukan: CloudFormation

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di area Instances, pilih Instances.
3. Di kotak pencarian, ketik nama instans Amazon EC2 yang ingin Anda akhiri, lalu tekan Enter.
4. Pilih instans Amazon EC2.
5. Pilih Actions, arahkan ke Instance State, dan kemudian pilih Terminate. Saat diminta, pilih Ya, Hentikan. Ulangi langkah-langkah ini untuk setiap instans Amazon EC2 tambahan.

Untuk menghapus bucket Amazon S3:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dalam daftar ember, telusuri ke dan pilih nama bucket Amazon S3 (misalnya**amzn-s3-demo-bucket**,).

3. Sebelum Anda dapat menghapus ember, Anda harus terlebih dahulu menghapus isinya. Pilih semua file dalam ember, seperti**`HelloWorld_App.zip`**. Di menu Tindakan, pilih Hapus. Ketika diminta untuk mengkonfirmasi penghapusan, pilih OK.
4. Setelah ember kosong, Anda dapat menghapus ember. Dalam daftar ember, pilih baris ember (tetapi bukan nama ember). Pilih Hapus bucket, dan ketika diminta untuk mengonfirmasi, pilih OK.

Untuk menghapus `HelloWorld_App` aplikasi dari CodeDeploy:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pilih **`HelloWorld_App`**.
4. Pilih Hapus aplikasi.
5. Saat diminta, masukkan **`Delete`**, lalu pilih Hapus.

Untuk menghapus asosiasi Manajer Negara Systems Manager:

1. Buka AWS Systems Manager konsol di https://console.aws.amazon.com/systems-manager.
2. Di panel navigasi, pilih State Manager.
3. Pilih asosiasi yang Anda buat dan pilih Hapus.

Apa selanjutnya?

Jika Anda telah tiba di sini, Anda telah berhasil menyelesaikan penerapan dengan CodeDeploy. Selamat!

Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)

Tutorial ini membantu Anda mendapatkan pengalaman CodeDeploy dengan memandu Anda melalui penerapan contoh revisi aplikasi ke satu instans lokal — yaitu, perangkat fisik yang bukan instans Amazon EC2 — menjalankan Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux (RHEL). Untuk informasi tentang instans lokal dan cara kerjanya CodeDeploy, lihat. Working with On-Premises Instances

Bukan apa yang Anda cari?

- Untuk mempraktikkan penerapan ke instans Amazon EC2 yang menjalankan Amazon Linux atau RHEL, lihat. Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix)
- Untuk mempraktikkan penerapan ke instans Amazon EC2 yang menjalankan Windows Server, lihat. Tutorial: Menyebarkan “halo, dunia!” aplikasi dengan CodeDeploy (Windows Server)

Topik

- Prasyarat
- Langkah 1: Konfigurasikan instans lokal
- Langkah 2: Buat contoh revisi aplikasi
- Langkah 3: Bundel dan unggah revisi aplikasi Anda ke Amazon S3
- Langkah 4: Terapkan revisi aplikasi Anda
- Langkah 5: Verifikasi penerapan Anda
- Langkah 6: Bersihkan Sumber Daya

Prasyarat

Sebelum Anda memulai tutorial ini, Anda harus menyelesaikan prasyarat diMemulai dengan CodeDeploy, yang meliputi mengkonfigurasi pengguna, menginstal atau meningkatkan, dan membuat peran layanan AWS CLI. Anda tidak perlu membuat profil instans IAM seperti yang dijelaskan dalam prasyarat. On-premises instance tidak menggunakan profil instans IAM.

Perangkat fisik yang akan Anda konfigurasikan sebagai instans lokal harus menjalankan salah satu sistem operasi yang tercantum. Sistem operasi yang didukung oleh CodeDeploy agen

Langkah 1: Konfigurasikan instans lokal

Sebelum dapat menerapkan ke instans lokal, Anda harus mengonfigurasinya. Ikuti instruksi diWorking with On-Premises Instances, dan kemudian kembali ke halaman ini.

Instal CodeDeploy agen

Setelah mengonfigurasi instans lokal, ikuti langkah-langkah untuk instans lokal di Instal CodeDeploy agen dan kembali ke halaman ini.

Langkah 2: Buat contoh revisi aplikasi

Pada langkah ini, Anda membuat contoh revisi aplikasi untuk diterapkan ke instans lokal Anda.

Karena sulit untuk mengetahui perangkat lunak dan fitur mana yang sudah terinstal—atau diizinkan untuk diinstal oleh kebijakan organisasi Anda—pada instans lokal Anda, contoh revisi aplikasi yang kami tawarkan di sini hanya menggunakan skrip batch (untuk Windows Server) atau skrip shell (untuk Server Ubuntu dan RHEL) untuk menulis file teks ke lokasi di instans lokal Anda. Satu file ditulis untuk masing-masing dari beberapa peristiwa siklus hidup CodeDeploy penerapan, termasuk Instal,, AfterInstall, ApplicationStartdan. ValidateService Selama peristiwa siklus hidup BeforeInstallpenerapan, skrip akan berjalan untuk menghapus file lama yang ditulis selama penerapan sampel ini sebelumnya dan membuat lokasi pada instance lokal untuk menulis file baru.

> **Note**
>
> Contoh revisi aplikasi ini mungkin gagal diterapkan jika salah satu dari berikut ini benar:
>
> - Pengguna yang memulai CodeDeploy agen pada instans lokal tidak memiliki izin untuk mengeksekusi skrip.
> - Pengguna tidak memiliki izin untuk membuat atau menghapus folder di lokasi yang tercantum dalam skrip.
> - Pengguna tidak memiliki izin untuk membuat file teks di lokasi yang tercantum dalam skrip.

> **Note**
>
> Jika Anda mengonfigurasi instance Windows Server dan ingin menyebarkan sampel yang berbeda, Anda mungkin ingin menggunakannya Langkah 2: Konfigurasikan konten sumber

Anda untuk menyebarkan ke instans Windows Server Amazon EC2 di Tutorial: Menyebarkan "halo, dunia!" aplikasi dengan CodeDeploy (Windows Server) tutorial.
Jika Anda mengonfigurasi instance RHEL dan ingin menerapkan sampel yang berbeda, Anda mungkin ingin menggunakan contoh yang ada Langkah 2: Konfigurasikan konten sumber Anda untuk disebarkan ke instans Amazon EC2 Amazon Linux atau Red Hat Enterprise Linux Amazon EC2 di Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix) tutorial.
Saat ini, tidak ada contoh alternatif untuk Ubuntu Server.

1. Pada mesin pengembangan Anda, buat subdirektori (subfolder) bernama `CodeDeployDemo-OnPrem` yang akan menyimpan file revisi aplikasi sampel, dan kemudian beralih ke subfolder. Untuk contoh ini, kami berasumsi Anda akan menggunakan `c:\temp` folder sebagai folder root untuk Windows Server atau `/tmp` folder sebagai folder root untuk Ubuntu Server dan RHEL. Jika Anda menggunakan folder yang berbeda, pastikan untuk menggantinya dengan folder kami di seluruh tutorial ini:

   Untuk Windows:

   ```
   mkdir c:\temp\CodeDeployDemo-OnPrem
   cd c:\temp\CodeDeployDemo-OnPrem
   ```

   Untuk Linux, macOS, atau Unix:

   ```
   mkdir /tmp/CodeDeployDemo-OnPrem
   cd /tmp/CodeDeployDemo-OnPrem
   ```

2. Di root `CodeDeployDemo-OnPrem` subfolder, gunakan editor teks untuk membuat dua file bernama `appspec.yml` dan`install.txt`:

   `appspec.yml`untuk 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`untuk Server Ubuntu dan 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
```

Untuk informasi selengkapnya tentang AppSpec file, lihat Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy dan CodeDeploy AppSpec referensi file.

`install.txt`:

```
The Install deployment lifecycle event successfully completed.
```

3. Di bawah root `CodeDeployDemo-OnPrem` subfolder, buat `scripts` subfolder, lalu beralih ke sana:

Untuk Windows:

```
mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
cd c:\temp\CodeDeployDemo-OnPrem\scripts
```

Untuk Linux, macOS, atau Unix:

```
mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
cd /tmp/CodeDeployDemo-OnPrem/scripts
```

4. Di root `scripts` subfolder, gunakan editor teks untuk membuat empat file bernama`before-install.bat`,,, dan `validate-service.bat` untuk Windows Server `after-install.batapplication-start.bat`, atau,, `before-install.sh` `after-install.shapplication-start.sh`, dan `validate-service.sh` untuk Ubuntu Server dan RHEL:

   Untuk 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
```

Untuk Server Ubuntu dan 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. Untuk Ubuntu Server dan RHEL saja, pastikan keempat skrip shell memiliki izin eksekusi:

```
chmod +x ./scripts/*
```

Langkah 3: Bundel dan unggah revisi aplikasi Anda ke Amazon S3

Sebelum dapat menerapkan revisi aplikasi, Anda harus menggabungkan file, lalu mengunggah bundel file ke bucket Amazon S3. Ikuti instruksi di Buat aplikasi dengan CodeDeploy danDorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan). (Meskipun Anda dapat memberikan nama aplikasi dan grup penyebaran apa pun, kami sarankan Anda menggunakan `CodeDeploy-OnPrem-App` nama aplikasi dan `CodeDeploy-OnPrem-DG` untuk nama grup penyebaran.) Setelah Anda menyelesaikan instruksi tersebut, kembali ke halaman ini.

> **Note**
>
> Atau, Anda dapat mengunggah bundel file ke GitHub repositori dan menyebarkannya dari sana. Untuk informasi selengkapnya, lihat Integrasi dengan CodeDeploy GitHub.

Langkah 4: Terapkan revisi aplikasi Anda

Setelah mengunggah revisi aplikasi ke bucket Amazon S3, coba terapkan ke instans lokal. Ikuti instruksi diBuat penerapan dengan CodeDeploy, dan kemudian kembali ke halaman ini.

Langkah 5: Verifikasi penerapan Anda

Untuk memverifikasi penerapan berhasil, ikuti instruksi diLihat CodeDeploy detail penerapan, lalu kembali ke halaman ini.

Jika penyebaran berhasil, Anda akan menemukan empat file teks dalam `c:\temp \CodeDeployExample` folder (untuk Windows Server) atau `/tmp/CodeDeployExample` (untuk Ubuntu Server dan RHEL).

Jika penerapan gagal, ikuti langkah-langkah pemecahan masalah di dan. View Instance Details Memecahkan masalah contoh Buat perbaikan yang diperlukan, rebundle dan unggah revisi aplikasi Anda, lalu coba penerapan lagi.

Langkah 6: Bersihkan Sumber Daya

Untuk menghindari biaya berkelanjutan untuk sumber daya yang Anda buat untuk tutorial ini, hapus bucket Amazon S3 jika Anda tidak lagi menggunakannya. Anda juga dapat membersihkan sumber daya terkait, seperti catatan grup aplikasi dan penyebaran di CodeDeploy dan instans lokal.

Anda dapat menggunakan AWS CLI atau kombinasi konsol Amazon S3 CodeDeploy dan AWS CLI untuk membersihkan sumber daya.

Membersihkan sumber daya (CLI)

Untuk menghapus bucket Amazon S3

- Panggil perintah rm bersama dengan `--recursive` sakelar terhadap ember (misalnya,`amzn-s3-demo-bucket`). Ember dan semua benda di ember akan dihapus.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Untuk menghapus catatan grup aplikasi dan penyebaran di CodeDeploy

- Panggil perintah hapus-aplikasi terhadap aplikasi (misalnya,`CodeDeploy-OnPrem-App`). Catatan untuk grup penyebaran dan penyebaran akan dihapus.

```
aws deploy delete-application --application-name your-application-name
```

Untuk membatalkan pendaftaran instans lokal dan menghapus pengguna IAM

- Panggil perintah deregister terhadap instance dan wilayah lokal:

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --
region your-region
```

> **Note**
>
> Jika Anda tidak ingin menghapus pengguna IAM yang terkait dengan instans lokal ini, gunakan `--no-delete-iam-user` opsi sebagai gantinya.

Untuk menghapus instalan CodeDeploy agen dan menghapus file konfigurasi dari instans lokal

- Dari instance lokal, panggil perintah uninstall:

```
aws deploy uninstall
```

Anda sekarang telah menyelesaikan semua langkah untuk membersihkan sumber daya yang digunakan untuk tutorial ini.

Bersihkan sumber daya (konsol)

Untuk menghapus bucket Amazon S3

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih ikon di sebelah ember yang ingin Anda hapus (misalnya,`amzn-s3-demo-bucket`), tetapi jangan memilih ember itu sendiri.
3. Pilih Tindakan, lalu pilih Hapus.
4. Saat diminta untuk menghapus ember, pilih OK.

Untuk menghapus catatan grup aplikasi dan penyebaran di CodeDeploy

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi pilih Aplikasi.
3. Pilih nama aplikasi yang ingin Anda hapus (misalnya,`CodeDeploy-OnPrem-App`) dan kemudian pilih Hapus aplikasi.
4. Saat diminta, masukkan nama aplikasi untuk mengonfirmasi bahwa Anda ingin menghapusnya, lalu pilih Hapus.

Anda tidak dapat menggunakan AWS CodeDeploy konsol untuk membatalkan pendaftaran instans lokal atau menghapus instalan agen. CodeDeploy Ikuti petunjuk dalam Untuk membatalkan pendaftaran instans lokal dan menghapus pengguna IAM.

Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling

Dalam tutorial ini, Anda akan menggunakan CodeDeploy untuk menyebarkan revisi aplikasi ke grup Auto Scaling. Amazon EC2 Auto Scaling meluncurkan instans Amazon EC2 menggunakan kondisi yang telah ditentukan sebelumnya, dan kemudian menghentikan instans tersebut saat tidak lagi diperlukan. Amazon EC2 Auto Scaling CodeDeploy dapat membantu penskalaan dengan memastikannya selalu memiliki jumlah instans Amazon EC2 yang benar yang tersedia untuk menangani beban penerapan. Untuk informasi tentang CodeDeploy integrasi Amazon EC2 Auto Scaling dengan, lihat. Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling

Topik

- Prasyarat
- Langkah 1: Buat dan konfigurasikan grup Auto Scaling
- Langkah 2: Menyebarkan aplikasi ke grup Auto Scaling
- Langkah 3: Periksa hasil Anda
- Langkah 4: Tingkatkan jumlah instans Amazon EC2 di grup Auto Scaling
- Langkah 5: Periksa kembali hasil Anda
- Langkah 6: Bersihkan

Prasyarat

Untuk mengikuti dalam tutorial ini:

- Selesaikan semua langkahMemulai dengan CodeDeploy, termasuk menyiapkan dan mengonfigurasi AWS CLI dan membuat profil instance IAM (**`CodeDeployDemo-EC2-Instance-Profile`**) dan peran layanan (**`CodeDeployDemo`**). Peran layanan adalah jenis peran IAM khusus yang memberikan izin layanan untuk bertindak atas nama Anda.
- Jika Anda membuat grup Auto Scaling dengan template peluncuran, Anda harus menambahkan izin berikut:
  - `ec2:RunInstances`
  - `ec2:CreateTags`
  - `iam:PassRole`

  Untuk informasi selengkapnya, lihatLangkah 2: Buat peran layanan, Membuat template peluncuran untuk grup Auto Scaling, dan Luncurkan dukungan template di Panduan Pengguna Amazon EC2 Auto Scaling.
- Buat dan gunakan revisi yang kompatibel dengan instance Server Ubuntu dan CodeDeploy. Untuk revisi Anda, Anda dapat melakukan salah satu hal berikut:
  - Buat dan gunakan revisi sampel Langkah 2: Buat contoh revisi aplikasi dalam Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux) tutorial.
  - Buat revisi sendiri, lihatBekerja dengan revisi aplikasi untuk CodeDeploy.
- Buat Grup Keamanan bernama **`CodeDeployDemo-AS-SG`** dengan aturan Inbound berikut:
  - Jenis: HTTP
  - Sumber: Di mana saja

  Ini diperlukan untuk melihat aplikasi Anda dan memverifikasi keberhasilan penerapan. Untuk informasi tentang cara membuat Grup Keamanan, lihat Membuat grup keamanan di panduan pengguna Amazon EC2.

Langkah 1: Buat dan konfigurasikan grup Auto Scaling

Pada langkah ini, Anda akan membuat grup Auto Scaling yang berisi satu instans Amazon Linux, RHEL, atau Windows Server Amazon EC2. Pada langkah selanjutnya, Anda akan menginstruksikan

Amazon EC2 Auto Scaling untuk menambahkan satu lagi instans Amazon EC2, CodeDeploy dan akan menerapkan revisi Anda ke dalamnya.

Topik

- Untuk membuat dan mengkonfigurasi grup Auto Scaling (CLI)
- Untuk membuat dan mengkonfigurasi grup Auto Scaling (konsol)

Untuk membuat dan mengkonfigurasi grup Auto Scaling (CLI)

1. Panggil create-launch-template perintah untuk membuat template peluncuran Amazon EC2.

   Sebelum Anda memanggil perintah ini, Anda memerlukan ID AMI yang berfungsi untuk tutorial ini, diwakili oleh placeholder*`image-id`*. Anda juga memerlukan nama key pair instans Amazon EC2 untuk mengaktifkan akses ke instans Amazon EC2, yang diwakili oleh placeholder. *`key-name`*

   Untuk mendapatkan ID AMI yang berfungsi dengan tutorial ini:

   a. Buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/

   b. Di panel navigasi, di bawah Instans, pilih Instans, lalu pilih Launch Instance.

   c. Pada tab Mulai Cepat pada halaman Pilih Gambar Mesin Amazon, catat ID AMI di sebelah Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS, atau Microsoft Windows Server 2012 R2.

   > **Note**
   >
   > Jika Anda memiliki versi khusus AMI yang kompatibel dengannya CodeDeploy, pilih di sini alih-alih menjelajah melalui tab Mulai Cepat. Untuk informasi tentang menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Scaling Scaling, lihat.

   Untuk key pair instans Amazon EC2, gunakan nama key pair instans Amazon EC2 Anda.

   Panggil create-launch-template perintahnya.

   Di mesin Linux, macOS, atau Unix lokal:

```
aws ec2 create-launch-template \
  --launch-template-name CodeDeployDemo-AS-Launch-Template \
  --launch-template-data file://config.json
```

Isi `config.json` file:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Di mesin Windows lokal:

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-
Template --launch-template-data file://config.json
```

Isi `config.json` file:

```
{
  "InstanceType":"t1.micro",
  "ImageId":"image-id",
  "IamInstanceProfile":{
    "Name":"CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName":"key-name"
}
```

Perintah ini, bersama dengan `config.json` file, membuat template peluncuran Amazon EC2 yang dinamai CodeDeployDemo-AS-Launch-Template untuk grup Auto Scaling Anda yang akan dibuat dalam langkah berikut berdasarkan jenis instans Amazon EC2 t1.micro. Berdasarkan masukan Anda untuk`ImageId`,`IamInstanceProfile`, dan`KeyName`, template peluncuran juga menentukan ID AMI, nama profil instance yang terkait dengan peran IAM untuk diteruskan ke instance saat peluncuran, dan key pair Amazon EC2 yang akan digunakan saat menghubungkan ke instance.

2. Panggil create-auto-scaling-group perintah untuk membuat grup Auto Scaling. Anda akan memerlukan nama salah satu Availability Zone di salah satu Wilayah yang terdaftar di Wilayah dan titik akhir di Referensi Umum AWS, yang diwakili oleh *availability-zone* placeholder.

   > **Note**
   >
   > Untuk melihat daftar Availability Zone di Region, hubungi:
   >
   > ```
   > aws ec2 describe-availability-zones --region region-name
   > ```
   >
   > Misalnya, untuk melihat daftar Availability Zone di Wilayah AS Barat (Oregon), hubungi:
   >
   > ```
   > aws ec2 describe-availability-zones --region us-west-2
   > ```
   >
   > Untuk daftar pengenal nama Wilayah, lihatNama bucket kit sumber daya menurut Wilayah.

   Di mesin Linux, macOS, atau Unix lokal:

   ```
   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
   ```

   Di mesin Windows lokal:

   ```
   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
   ```

   Perintah ini membuat grup Auto Scaling bernama **CodeDeployDemo-AS-Group** berdasarkan template peluncuran Amazon EC2 yang diberi nama. **CodeDeployDemo-AS-Launch-**

`Template` Grup Auto Scaling ini hanya memiliki satu instans Amazon EC2, dan dibuat di Availability Zone yang ditentukan. Setiap instance dalam grup Auto Scaling ini akan memiliki tag. `Name=CodeDeployDemo` Tag akan digunakan saat menginstal CodeDeploy agen nanti.

3. Panggil describe-auto-scaling-groups perintah terhadap**`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
 LifecycleState]" --output text
```

Jangan lanjutkan sampai nilai yang dikembalikan menunjukkan `Healthy` dan`InService`.

4. Instance dalam grup Auto Scaling Anda harus memiliki agen CodeDeploy yang diinstal untuk digunakan CodeDeploy dalam penerapan. Instal CodeDeploy agen dengan memanggil create-association perintah dari AWS Systems Manager tag yang ditambahkan saat grup Auto Scaling dibuat.

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

 --parameters action=Install, name=AWSCodeDeployAgent \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Perintah ini membuat asosiasi di Systems Manager State Manager yang akan menginstal CodeDeploy agen pada semua instance di grup Auto Scaling dan kemudian mencoba memperbaruinya pada pukul 2:00 setiap Minggu pagi. Untuk informasi lebih lanjut tentang CodeDeploy agen, lihat Bekerja dengan CodeDeploy agen. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu AWS Systems Manager.

Untuk membuat dan mengkonfigurasi grup Auto Scaling (konsol)

1. Buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di bilah navigasi global, pastikan salah satu Wilayah yang terdaftar di Wilayah dan titik akhir di Referensi Umum AWSdipilih. Sumber daya Amazon EC2 Auto Scaling terikat ke Wilayah yang Anda tentukan CodeDeploy , dan hanya didukung di Wilayah tertentu.
3. Di bilah navigasi, di bawah Instans, pilih Luncurkan Template.
4. Pilih Buat templat peluncuran.

5. Dalam dialog Launch template name and description, untuk Launch nama template, masukkan**`CodeDeployDemo-AS-Launch-Template`**. Biarkan default untuk bidang lainnya.
6. Dalam dialog Amazon machine image (AMI), klik dropdown di bawah AMI, pilih AMI yang bekerja dengan tutorial ini:

   - Pada tab Mulai Cepat dari dropdown AMI, pilih salah satu dari yang berikut: Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS, atau Microsoft Windows Server 2012 R2.

   > **Note**
   >
   > Jika Anda memiliki versi khusus AMI yang kompatibel dengannya CodeDeploy, pilih di sini alih-alih menjelajah melalui tab Mulai Cepat. Untuk informasi tentang menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Menggunakan AMI khusus dengan CodeDeploy dan Amazon EC2 Auto Scaling Scaling, lihat.

7. Dalam tipe Instance, pilih dropdown dan pilih t1.micro. Anda dapat menggunakan bilah pencarian untuk menemukannya lebih cepat.
8. Dalam kotak dialog Key pair (login), pilih Choose an existing key pair. Dalam daftar drop-down Select a key pair, pilih key pair instans Amazon EC2 yang Anda buat atau gunakan pada langkah sebelumnya.
9. Di kotak dialog Pengaturan jaringan, pilih Virtual Public Cloud (VPC).

   Di dropdown Grup keamanan, pilih grup keamanan yang Anda buat di bagian prasyarat tutorial (). **`CodeDeployDemo-AS-SG`**
10. Perluas kotak dialog Detail lanjutan. Di menu tarik-turun profil instans IAM, pilih peran IAM yang Anda buat sebelumnya (**`CodeDeployDemo-EC2-Instance-Profile`**) di bawah profil instans IAM.

    Tinggalkan sisa default.
11. Pilih Buat templat peluncuran.
12. Dalam kotak dialog Langkah berikutnya, pilih Buat grup Auto Scaling.
13. Pada halaman Choose launch template atau konfigurasi, untuk nama grup Auto Scaling, ketik. **`CodeDeployDemo-AS-Group`**

14. Di kotak dialog Launch template, template peluncuran Anda (**`CodeDeployDemo-AS-Launch-Template`**) harus diisi, jika tidak, pilih dari menu tarik-turun. Tinggalkan default dan pilih Berikutnya.
15. Pada halaman Pilih opsi peluncuran instance, di bagian Jaringan, untuk VPC, pilih VPC default. Kemudian untuk Availability Zones dan subnet, pilih subnet default. Anda harus membuat VPC jika Anda tidak dapat memilih default. Untuk informasi selengkapnya, lihat Memulai Amazon VPC.
16. Di bagian Persyaratan tipe instans, gunakan pengaturan default untuk menyederhanakan langkah ini. (Jangan mengganti template peluncuran.) Untuk tutorial ini, Anda hanya akan meluncurkan On-Demand Instans menggunakan jenis instance yang ditentukan dalam template peluncuran Anda.
17. Pilih Berikutnya untuk pergi ke halaman Configure advanced options.
18. Pertahankan nilai default dan pilih Berikutnya.
19. Pada halaman Konfigurasi ukuran grup dan kebijakan penskalaan, pertahankan nilai ukuran Grup default 1. Pilih Berikutnya.
20. Lewati langkah untuk mengonfigurasi notifikasi, dan pilih Berikutnya.
21. Pada halaman Tambahkan tag, tambahkan tag yang akan digunakan saat menginstal CodeDeploy agen nanti. Pilih Tambahkan tanda.

    a. Di Key, masukkan**`Name`**.

    b. Dalam Value (Nilai), masukkan **`CodeDeployDemo`**.

    Pilih Berikutnya.
22. Tinjau informasi grup Auto Scaling Anda di halaman Tinjauan, lalu pilih grup Buat Auto Scaling.
23. Di bilah navigasi, dengan Grup Auto Scaling dipilih, pilih**`CodeDeployDemo-AS-Group`**, lalu pilih tab Manajemen Instance. Jangan lanjutkan sampai nilai InServicemuncul di kolom Siklus Hidup dan nilai Sehat muncul di kolom Status Kesehatan.
24. Instal CodeDeploy agen dengan mengikuti langkah-langkah di Instal CodeDeploy agen dan menggunakan tag `Name=CodeDeployDemo` instance.

Langkah 2: Menyebarkan aplikasi ke grup Auto Scaling

Pada langkah ini, Anda akan menerapkan revisi ke instans Amazon EC2 tunggal di grup Auto Scaling.

Topik

- Untuk membuat penerapan (CLI)
- Untuk membuat penyebaran (konsol)

Untuk membuat penerapan (CLI)

1. Panggil create-application perintah untuk membuat aplikasi bernama**`SimpleDemoApp`**:

   ```
   aws deploy create-application --application-name SimpleDemoApp
   ```

2. Anda seharusnya sudah membuat peran layanan dengan mengikuti instruksi diLangkah 2: Buat peran layanan untuk CodeDeploy. Peran layanan akan memberikan CodeDeploy izin untuk mengakses instans Amazon EC2 Anda untuk memperluas (membaca) tag mereka. Anda akan membutuhkan peran layanan ARN. Untuk mendapatkan peran layanan ARN, ikuti instruksi di. Dapatkan peran layanan ARN (CLI)

3. Sekarang setelah Anda memiliki peran layanan ARN, panggil create-deployment-group perintah untuk membuat grup penyebaran bernama**`SimpleDemoDG`**, terkait dengan aplikasi bernama**`SimpleDemoApp`**, menggunakan grup Auto Scaling bernama **`CodeDeployDemo-AS-Group`** dan konfigurasi penyebaran bernama**`CodeDeployDefault.OneAtATime`**, dengan peran layanan yang ditentukan ARN.

   > **Note**
   >
   > Perintah create-deployment-group menyediakan dukungan untuk membuat pemicu yang menghasilkan pengiriman notifikasi Amazon SNS ke pelanggan topik tentang peristiwa tertentu dalam penerapan dan instance. Perintah ini juga mendukung opsi untuk memutar kembali penerapan secara otomatis dan mengatur alarm untuk menghentikan penerapan saat memantau ambang batas di alarm Amazon terpenuhi. CloudWatch Perintah untuk tindakan ini tidak termasuk dalam tutorial ini.

   Di mesin Linux, macOS, atau Unix lokal:

   ```
   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
```

Di mesin Windows lokal:

```
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. Panggil create-deployment perintah untuk membuat penyebaran yang terkait dengan aplikasi bernama**SimpleDemoApp**, konfigurasi penyebaran bernama, grup penyebaran bernama **CodeDeployDefault.OneAtATimeSimpleDemoDG**, menggunakan revisi di lokasi yang ditentukan.

   Untuk instans Amazon Linux dan RHEL Amazon EC2, menelepon dari mesin Linux, macOS, atau Unix lokal

```
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*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda. Misalnya, untuk Wilayah Timur AS (Ohio), ganti *bucket-name* dengan`aws-codedeploy-us-east-2`. Untuk daftar nama bucket, lihatNama bucket kit sumber daya menurut Wilayah.

   Untuk instans Amazon Linux dan RHEL Amazon EC2, menelepon dari mesin Windows lokal

```
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*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda. Misalnya, untuk Wilayah Timur AS (Ohio), ganti *bucket-name* dengan`aws-`

codedeploy-us-east-2. Untuk daftar nama bucket, lihatNama bucket kit sumber daya menurut Wilayah.

Untuk instans Windows Server Amazon EC2, panggilan dari mesin Linux, macOS, atau Unix lokal

```
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*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda. Misalnya, untuk Wilayah Timur AS (Ohio), ganti *bucket-name* denganaws-codedeploy-us-east-2. Untuk daftar nama bucket, lihatNama bucket kit sumber daya menurut Wilayah.

Untuk instans Windows Server Amazon EC2, menelepon dari mesin Windows lokal

```
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*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda. Misalnya, untuk Wilayah Timur AS (Ohio), ganti *bucket-name* denganaws-codedeploy-us-east-2. Untuk daftar nama bucket, lihatNama bucket kit sumber daya menurut Wilayah.

> **Note**
>
> Saat ini, CodeDeploy tidak menyediakan contoh revisi untuk menyebarkan ke instans Ubuntu Server Amazon EC2. Untuk membuat revisi sendiri, lihatBekerja dengan revisi aplikasi untuk CodeDeploy.

5. Panggil get-deployment perintah untuk memastikan penerapan berhasil.

Sebelum Anda memanggil perintah ini, Anda akan memerlukan ID penyebaran, yang seharusnya dikembalikan oleh panggilan ke create-deployment perintah. Jika Anda perlu mendapatkan ID penyebaran lagi, panggil list-deployments perintah terhadap aplikasi bernama **`SimpleDemoApp`** dan grup penyebaran bernama: **`SimpleDemoDG`**

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
name SimpleDemoDG --query "deployments" --output text
```

Sekarang, panggil get-deployment perintah menggunakan ID penyebaran:

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

Jangan lanjutkan sampai nilai yang dikembalikan`Succeeded`.

Untuk membuat penyebaran (konsol)

1. Anda seharusnya sudah membuat peran layanan dengan mengikuti instruksi di[Langkah 2: Buat peran layanan untuk CodeDeploy](). Peran layanan akan memberikan CodeDeploy izin untuk mengakses instance Anda untuk memperluas (membaca) tag mereka. Sebelum Anda menggunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda, Anda akan memerlukan peran layanan ARN. Untuk mendapatkan peran layanan ARN, ikuti instruksi di. [Dapatkan peran layanan ARN (konsol)]()
2. Sekarang setelah Anda memiliki peran layanan ARN, Anda dapat menggunakan CodeDeploy konsol untuk menyebarkan revisi aplikasi Anda.

   Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda atur[Memulai dengan CodeDeploy]().

3. Di panel navigasi, perluas Deploy, lalu pilih Applications.
4. Pilih Create application (Buat aplikasi).

5. Pilih aplikasi Kustom.
6. Dalam nama Aplikasi, masukkan**`SimpleDemoApp`**.
7. Di platform Compute, pilih EC2/On-premises.
8. Pilih Create application (Buat aplikasi).
9. Pada tab Deployment groups, pilih Create deployment group.
10. Dalam nama grup Deployment, masukkan**`SimpleDemoDG`**.
11. Di Peran Layanan, pilih nama peran layanan Anda.
12. Dalam jenis Deployment, pilih In-place.
13. Dalam konfigurasi Lingkungan pilih grup Auto Scaling, lalu pilih. **`CodeDeployDemo-AS-Group`**
14. Dalam konfigurasi Deployment, pilih CodeDeployDefault.OneAtATime.
15. Hapus Aktifkan penyeimbangan beban.
16. Pilih Buat grup penyebaran.
17. Di halaman grup penyebaran, pilih Buat penyebaran.
18. Dalam jenis Revisi, pilih Aplikasi saya disimpan di Amazon S3.
19. Di lokasi Revisi, masukkan lokasi aplikasi sampel untuk sistem operasi dan Wilayah Anda.

    Untuk instans Amazon Linux dan RHEL Amazon EC2

Region	Lokasi aplikasi sampel
Wilayah AS Timur (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip`
Wilayah AS Timur (Virginia Utara)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip`
Wilayah AS Barat (California Utara)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip`

Region	Lokasi aplikasi sampel
Wilayah AS Barat (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip`
Wilayah Kanada (Pusat)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip`
Wilayah Eropa (Irlandia)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip`
Wilayah Eropa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip`
Wilayah Eropa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip`
Wilayah Eropa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip`
Wilayah Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`

Region	Lokasi aplikasi sampel
Wilayah Asia Pasifik (Hong Kong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pacific (Tokyo)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pasifik (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pasifik (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pacific (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pasifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip`
Wilayah Asia Pasifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip`

Region	Lokasi aplikasi sampel
Wilayah Amerika Selatan (Sao Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip`

Untuk instans Windows Server Amazon EC2

Region	Lokasi aplikasi sampel
Wilayah AS Timur (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip`
Wilayah AS Timur (Virginia Utara)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip`
Wilayah AS Barat (California Utara)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip`
Wilayah AS Barat (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip`
Wilayah Kanada (Pusat)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip`

Region	Lokasi aplikasi sampel
Wilayah Eropa (Irlandia)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip`
Wilayah Eropa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip`
Wilayah Eropa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip`
Wilayah Eropa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip`
Wilayah Israel (Tel Aviv)	`https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Wilayah Asia Pasifik (Hong Kong)	`https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Wilayah Asia Pacific (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip`

Region	Lokasi aplikasi sampel
Wilayah Asia Pasifik (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip`
Wilayah Asia Pacific (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip`
Wilayah Asia Pasifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip`
Wilayah Asia Pasifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip`
Wilayah Amerika Selatan (Sao Paulo)	`http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip`

Untuk instans Ubuntu Server Amazon EC2

Ketik lokasi revisi aplikasi kustom Anda yang disimpan di Amazon S3.

20. Biarkan deskripsi Deployment kosong.
21. Perluas Lanjutan.
22. Pilih Buat penerapan.

> **Note**
>
> Jika Gagal muncul alih-alih Berhasil, Anda mungkin ingin mencoba beberapa teknik di Pantau dan pecahkan masalah penerapan Anda (menggunakan nama aplikasi **SimpleDemoApp** dan nama grup penyebaran). **SimpleDemoDG**

Langkah 3: Periksa hasil Anda

Pada langkah ini, Anda akan memeriksa apakah CodeDeploy telah menginstal **SimpleDemoApp** revisi pada instans Amazon EC2 tunggal di grup Auto Scaling.

Topik

- Untuk memeriksa hasil (CLI)
- Untuk memeriksa hasil (konsol)

Untuk memeriksa hasil (CLI)

Pertama, Anda memerlukan DNS publik dari instans Amazon EC2.

Gunakan AWS CLI untuk mendapatkan DNS publik instans Amazon EC2 di grup Auto Scaling dengan memanggil perintah. describe-instances

Sebelum Anda memanggil perintah ini, Anda akan memerlukan ID instans Amazon EC2. Untuk mendapatkan ID, hubungi describe-auto-scaling-groups lawan **CodeDeployDemo-AS-Group** seperti yang Anda lakukan sebelumnya:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-
AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Sekarang panggil describe-instances perintah:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

Nilai yang dikembalikan adalah DNS publik dari instans Amazon EC2.

Menggunakan browser web, tampilkan SimpleDemoApp revisi yang diterapkan ke instans Amazon EC2 tersebut, menggunakan URL seperti berikut:

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

Jika Anda melihat halaman ucapan selamat, Anda telah berhasil menggunakan revisi CodeDeploy ke satu instans Amazon EC2 dalam grup Auto Scaling!

Selanjutnya, Anda akan menambahkan instans Amazon EC2 ke grup Auto Scaling. Setelah Amazon EC2 Auto Scaling menambahkan instans Amazon EC2, akan menerapkan revisi Anda ke CodeDeploy instans baru.

Untuk memeriksa hasil (konsol)

Pertama, Anda memerlukan DNS publik dari instans Amazon EC2.

Buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/

Di panel navigasi Amazon EC2, di bawah Auto Scaling, pilih Grup Auto Scaling, lalu pilih entri. **`CodeDeployDemo-AS-Group`**

Pada tab Instans, pilih ID instans Amazon EC2 dalam daftar.

Pada halaman Instans, pada tab Deskripsi, perhatikan nilai DNS Publik. Seharusnya terlihat seperti ini:**`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Menggunakan browser web, tampilkan SimpleDemoApp revisi yang diterapkan ke instans Amazon EC2 tersebut, menggunakan URL seperti berikut:

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

Jika Anda melihat halaman ucapan selamat, Anda telah berhasil menggunakan revisi CodeDeploy ke satu instans Amazon EC2 dalam grup Auto Scaling!

Selanjutnya, Anda menambahkan instans Amazon EC2 ke grup Auto Scaling. Setelah Amazon EC2 Auto Scaling menambahkan instans Amazon EC2, akan menerapkan revisi Anda ke CodeDeploy instans Amazon EC2 yang baru.

Langkah 4: Tingkatkan jumlah instans Amazon EC2 di grup Auto Scaling

Pada langkah ini, Anda menginstruksikan grup Auto Scaling untuk membuat instans Amazon EC2 tambahan. Setelah Amazon EC2 Auto Scaling membuat CodeDeploy instance, menerapkan revisi Anda ke instans tersebut.

Topik

- Untuk mengukur jumlah instans Amazon EC2 di grup Auto Scaling (CLI)
- Untuk memperkecil jumlah instans Amazon EC2 di grup penerapan (konsol)

Untuk mengukur jumlah instans Amazon EC2 di grup Auto Scaling (CLI)

1. Panggil update-auto-scaling-group perintah untuk meningkatkan instans Amazon EC2 di grup Auto Scaling yang diberi nama **`CodeDeployDemo-AS-Group`** dari satu menjadi dua.

   Di mesin Linux, macOS, atau Unix lokal:

   ```
   aws autoscaling update-auto-scaling-group \
     --auto-scaling-group-name CodeDeployDemo-AS-Group \
     --min-size 2 \
     --max-size 2 \
     --desired-capacity 2
   ```

   Di mesin Windows lokal:

   ```
   aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-
   AS-Group --min-size 2 --max-size 2 --desired-capacity 2
   ```

2. Pastikan grup Auto Scaling sekarang memiliki dua instans Amazon EC2. Panggil describe-auto-scaling-groups perintah terhadap**`CodeDeployDemo-AS-Group`**:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
    CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
    LifecycleState]" --output text
   ```

   Jangan lanjutkan sampai kedua nilai yang dikembalikan menunjukkan `Healthy` dan`InService`.

Untuk memperkecil jumlah instans Amazon EC2 di grup penerapan (konsol)

1. Di bilah navigasi Amazon EC2, di bawah Auto Scaling, pilih Grup Auto Scaling, lalu pilih. **`CodeDeployDemo-AS-Group`**
2. Pilih Tindakan, dan kemudian pilih Edit.
3. Pada tab Detail, di kotak Diinginkan, Min, dan Maks, ketik**2**, lalu pilih Simpan.
4. Pilih tab Instances. Instans Amazon EC2 baru akan muncul dalam daftar. (Jika instans tidak muncul, Anda mungkin perlu memilih tombol Refresh beberapa kali.) Jangan lanjutkan sampai nilai InServicemuncul di kolom Siklus Hidup dan nilai Sehat muncul di kolom Status Kesehatan.

Langkah 5: Periksa kembali hasil Anda

Pada langkah ini, Anda akan memeriksa untuk melihat apakah CodeDeploy menginstal SimpleDemoApp revisi pada instance baru di grup Auto Scaling.

Topik

- Untuk memeriksa hasil penerapan otomatis (CLI)
- Untuk memeriksa hasil penerapan otomatis (konsol)

Untuk memeriksa hasil penerapan otomatis (CLI)

1. Sebelum Anda memanggil get-deployment perintah, Anda akan memerlukan ID penyebaran otomatis. Untuk mendapatkan ID, panggil list-deployments perintah terhadap aplikasi bernama **`SimpleDemoApp`** dan grup penyebaran bernama**`SimpleDemoDG`**:

   ```
   aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-
   name SimpleDemoDG --query "deployments" --output text
   ```

   Harus ada dua ID penerapan. Gunakan yang belum Anda gunakan dalam panggilan ke get-deployment perintah:

   ```
   aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
   [status, creator]" --output text
   ```

   Selain status penyebaran, Anda akan melihat `autoScaling` di output perintah. (`autoScaling`berarti Amazon EC2 Auto Scaling membuat penerapan.)

Jangan lanjutkan sampai status penerapan ditampilkan`Succeeded`.

2. Sebelum Anda memanggil describe-instances perintah, Anda akan memerlukan ID instans Amazon EC2 baru. Untuk mendapatkan ID ini, lakukan panggilan lain ke describe-auto-scaling-groups perintah terhadap**`CodeDeployDemo-AS-Group`**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
 CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --
output text
```

Sekarang buat panggilan ke describe-instances perintah:

```
aws ec2 describe-instances --instance-id instance-id --query
 "Reservations[0].Instances[0].PublicDnsName" --output text
```

Dalam output describe-instances perintah, perhatikan DNS publik untuk instans Amazon EC2 baru.

3. Menggunakan browser web, tampilkan `SimpleDemoApp` revisi yang diterapkan ke instans Amazon EC2 tersebut, menggunakan URL seperti berikut:

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

Jika halaman ucapan selamat muncul, Anda telah menggunakan revisi CodeDeploy ke instans Amazon EC2 yang ditingkatkan dalam grup Auto Scaling!

Untuk memeriksa hasil penerapan otomatis (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Deployments.

3. Pilih ID penerapan penerapan yang dibuat oleh Amazon EC2 Auto Scaling.

   .

4. Halaman Deployment menampilkan informasi tentang penyebaran. Biasanya, Anda akan membuat penerapan sendiri, tetapi Amazon EC2 Auto Scaling membuatnya atas nama Anda untuk menerapkan revisi Anda ke instans Amazon EC2 yang baru.
5. Setelah Berhasil ditampilkan di bagian atas halaman, verifikasi hasil pada instance. Anda harus terlebih dahulu mendapatkan DNS publik dari instance:
6. Di panel navigasi Amazon EC2, di bawah Auto Scaling, pilih Grup Auto Scaling, lalu pilih entri. **`CodeDeployDemo-AS-Group`**
7. Pada tab Instans, pilih ID instans Amazon EC2 baru.
8. Pada halaman Instans, pada tab Deskripsi, perhatikan nilai DNS Publik. Seharusnya terlihat seperti ini:**`ec2-01-234-567-890.compute-1.amazonaws.com`**.

Tampilkan `SimpleDemoApp` revisi yang diterapkan ke instance menggunakan URL seperti berikut:

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

Jika halaman ucapan selamat muncul, Anda telah menggunakan revisi CodeDeploy ke instans Amazon EC2 yang ditingkatkan dalam grup Auto Scaling!

Langkah 6: Bersihkan

Pada langkah ini, Anda akan menghapus grup Auto Scaling untuk menghindari biaya berkelanjutan untuk sumber daya yang Anda gunakan selama tutorial ini. Secara opsional, Anda dapat menghapus konfigurasi Auto Scaling CodeDeploy dan catatan komponen deployment.

Topik

- Untuk membersihkan sumber daya (CLI)
- Untuk membersihkan sumber daya (konsol)

Untuk membersihkan sumber daya (CLI)

1. Hapus grup Auto Scaling dengan memanggil delete-auto-scaling-group perintah terhadap. **`CodeDeployDemo-AS-Group`** Ini juga akan mengakhiri instans Amazon EC2.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-
AS-Group --force-delete
```

2. Secara opsional, hapus template peluncuran Auto Scaling dengan memanggil perintah delete-launch-template terhadap konfigurasi peluncuran bernama: **CodeDeployDemo-AS-Launch-Template**

```
aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-
Template
```

3. Secara opsional, hapus aplikasi dari CodeDeploy dengan memanggil delete-application perintah terhadap aplikasi bernama**SimpleDemoApp**. Ini juga akan menghapus semua catatan penyebaran, grup penyebaran, dan revisi terkait.

```
aws deploy delete-application --application-name SimpleDemoApp
```

4. Untuk menghapus asosiasi Manajer Negara Systems Manager, panggil delete-association perintah.

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

Anda bisa mendapatkan *association-id* dengan memanggil describe-association perintah.

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

Untuk membersihkan sumber daya (konsol)

Untuk menghapus grup Auto Scaling, yang juga menghentikan instans Amazon EC2:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi Amazon EC2, di bawah Auto Scaling, pilih Grup Auto Scaling, lalu pilih entri. **CodeDeployDemo-AS-Group**
3. Pilih Tindakan, pilih Hapus, lalu pilih Ya, Hapus.

(Opsional) Untuk menghapus template peluncuran:

1. Di bilah navigasi, di bawah Auto Scaling, pilih Luncurkan Konfigurasi, lalu pilih. **`CodeDeployDemo-AS-Launch-Template`**
2. Pilih Tindakan, pilih Hapus konfigurasi peluncuran, lalu pilih Ya, Hapus.

1. Secara opsional, hapus aplikasi dari CodeDeploy. Ini juga akan menghapus semua catatan penyebaran, grup penyebaran, dan revisi terkait. Buka konsol CodeDeploy di https://console.aws.amazon.com/codedeploy.
2. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

   Di panel navigasi, perluas Deploy, lalu pilih Applications.

3. Dalam daftar aplikasi, pilih SimpleDemoApp.
4. Pada halaman Detail aplikasi, pilih Hapus aplikasi.
5. Saat diminta, masukkan **`Delete`**, lalu pilih Hapus.

Untuk menghapus asosiasi Manajer Negara Systems Manager:

1. Buka AWS Systems Manager konsol di https://console.aws.amazon.com/systems-manager.
2. Di panel navigasi, pilih State Manager.
3. Pilih asosiasi yang Anda buat dan pilih Hapus.

Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub

Dalam tutorial ini, Anda gunakan CodeDeploy untuk menerapkan contoh revisi aplikasi dari GitHub satu instans Amazon EC2 yang menjalankan Amazon Linux, satu instance Red Hat Enterprise

Linux (RHEL), atau satu instance Windows Server. Untuk informasi tentang GitHub integrasi dengan CodeDeploy, lihatIntegrasi dengan CodeDeploy GitHub.

> **Note**
>
> Anda juga dapat menggunakan CodeDeploy untuk menyebarkan revisi aplikasi dari GitHub ke instance Server Ubuntu. Anda dapat menggunakan contoh revisi yang dijelaskan Langkah 2: Buat contoh revisi aplikasi dalamTutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux), atau Anda dapat membuat revisi yang kompatibel dengan instance Ubuntu Server dan. CodeDeploy Untuk membuat revisi Anda sendiri, lihat Rencanakan revisi untuk CodeDeploy danTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.

Topik

- Prasyarat
- Langkah 1: Siapkan GitHub akun
- Langkah 2: Buat GitHub repositori
- Langkah 3: Unggah contoh aplikasi ke GitHub repositori Anda
- Langkah 4: Menyediakan sebuah instance
- Langkah 5: Buat grup aplikasi dan penyebaran
- Langkah 6: Menyebarkan aplikasi ke instance
- Langkah 7: Pantau dan verifikasi penyebaran
- Langkah 8: Membersihkan

Prasyarat

Sebelum Anda memulai tutorial ini, lakukan hal berikut:

- Instal Git di mesin lokal Anda. Untuk menginstal Git, lihat Unduhan Git.
- Selesaikan langkah-langkahnyaMemulai dengan CodeDeploy, termasuk menginstal dan mengonfigurasi file. AWS CLI Ini sangat penting jika Anda ingin menggunakan AWS CLI untuk menyebarkan revisi dari GitHub ke instance.

Langkah 1: Siapkan GitHub akun

Anda akan memerlukan GitHub akun untuk membuat GitHub repositori tempat revisi akan disimpan. Jika Anda sudah memiliki GitHub akun, lompat ke depanLangkah 2: Buat GitHub repositori.

1. Kunjungi https://github.com/join.
2. Ketik nama pengguna, alamat email Anda, dan kata sandi.
3. Pilih Daftar GitHub, lalu ikuti petunjuknya.

Langkah 2: Buat GitHub repositori

Anda akan memerlukan GitHub repositori untuk menyimpan revisi.

Jika Anda sudah memiliki GitHub repositori, pastikan untuk mengganti namanya untuk **`CodeDeployGitHubDemo`** seluruh tutorial ini, dan kemudian lompat ke depan. Langkah 3: Unggah contoh aplikasi ke GitHub repositori Anda

1. Di GitHub halaman beranda, lakukan salah satu hal berikut:
   - Di repositori Anda, pilih Repositori baru.
   - Pada bilah navigasi, pilih Buat baru (+), lalu pilih Repositori baru.
2. Di halaman Buat repositori baru, lakukan hal berikut:
   - Di kotak nama Repositori, masukkan. **`CodeDeployGitHubDemo`**
   - Pilih Publik.

     > **Note**
     >
     > Memilih opsi Publik default berarti siapa pun dapat melihat repositori ini. Anda dapat memilih opsi Private untuk membatasi siapa yang dapat melihat dan berkomitmen ke repositori.

   - Kosongkan kotak centang Inisialisasi repositori ini dengan README. Anda akan membuat `README.md` file secara manual di langkah berikutnya sebagai gantinya.
   - Pilih Buat repositori.
3. Ikuti instruksi untuk jenis mesin lokal Anda untuk menggunakan baris perintah untuk membuat repositori.

> **Note**
>
> Jika Anda telah mengaktifkan otentikasi dua faktor GitHub, pastikan Anda memasukkan token akses pribadi alih-alih kata sandi GitHub masuk Anda jika diminta kata sandi. Untuk selengkapnya, lihat Menyediakan kode autentikasi 2FA Anda.

Di mesin Linux, macOS, atau Unix lokal:

1. Dari terminal, jalankan perintah berikut, satu per satu, di *user-name* mana nama GitHub pengguna Anda:

```
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. Biarkan terminal terbuka di `/tmp/CodeDeployGitHubDemo` lokasi.

Di mesin Windows lokal:

1. Dari command prompt yang berjalan sebagai administrator, jalankan perintah berikut, satu per satu:

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. Di Notepad, simpan file. `README.md` Tutup Notepad. Jalankan perintah berikut, satu per satu, di *`user-name`* mana nama GitHub pengguna Anda:

```
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. Biarkan command prompt terbuka di `c:\temp\CodeDeployGitHubDemo` lokasi.

Langkah 3: Unggah contoh aplikasi ke GitHub repositori Anda

Pada langkah ini, Anda akan menyalin revisi sampel dari bucket Amazon S3 publik ke GitHub repositori Anda. (Untuk kesederhanaan, contoh revisi yang disediakan untuk tutorial ini adalah halaman web tunggal.)

> **Note**
>
> Jika Anda menggunakan salah satu revisi Anda alih-alih revisi sampel kami, revisi Anda harus:
>
> - Ikuti pedoman di Rencanakan revisi untuk CodeDeploy danTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.
> - Bekerja dengan jenis instance yang sesuai.

- Dapat diakses dari GitHub dasbor Anda.

Jika revisi Anda memenuhi persyaratan ini, lanjutkan keLangkah 5: Buat grup aplikasi dan penyebaran.
Jika Anda menerapkan ke instance Server Ubuntu, Anda harus mengunggah ke GitHub repositori Anda revisi yang kompatibel dengan instance Server Ubuntu dan. CodeDeploy Untuk informasi selengkapnya, lihat Rencanakan revisi untuk CodeDeploy dan Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.

Topik

- Dorong revisi sampel dari mesin Linux, macOS, atau Unix lokal
- Dorong revisi sampel dari mesin Windows lokal

Dorong revisi sampel dari mesin Linux, macOS, atau Unix lokal

Dengan terminal Anda masih terbuka, misalnya, `/tmp/CodeDeployGitHubDemo` lokasi, jalankan perintah berikut satu per satu:

Note

Jika Anda berencana untuk menyebarkan ke instance Windows Server, gantilah `SampleApp_Windows.zip` `SampleApp_Linux.zip` dalam perintah.

```
(Amazon S3 copy command)
```

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

*(Amazon S3 copy command)*Dimana salah satu dari berikut ini:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2`untuk Wilayah Timur AS (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1`untuk Wilayah AS Timur (Virginia N.)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1`untuk Wilayah AS Barat (California Utara)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2`untuk Wilayah Barat AS (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1`untuk Wilayah Kanada (Tengah)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1`untuk Wilayah Eropa (Irlandia)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2`untuk Wilayah Eropa (London)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3`untuk Wilayah Eropa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1`untuk Wilayah Eropa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1`untuk Wilayah Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1`untuk Wilayah Asia Pasifik (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1`untuk Wilayah Asia Pasifik (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2`untuk Wilayah Asia Pasifik (Seoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1`untuk Wilayah Asia Pasifik (Singapura)

- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2`untuk Wilayah Asia Pasifik (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4`untuk Wilayah Asia Pasifik (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1`untuk Wilayah Asia Pasifik (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1`untuk Wilayah Amerika Selatan (São Paulo)

Dorong revisi sampel dari mesin Windows lokal

Dengan prompt perintah Anda masih terbuka, misalnya, `c:\temp\CodeDeployGitHubDemo` lokasi, jalankan perintah berikut satu per satu:

> **Note**
>
> Jika Anda berencana untuk menyebarkan ke instans Amazon Linux atau RHEL, gantilah `SampleApp_Linux.zip` `SampleApp_Windows.zip` dalam perintah.

```
(Amazon S3 copy command)
```

Buka zip isi file ZIP langsung ke direktori lokal (misalnya`c:\temp\CodeDeployGitHubDemo`), bukan ke subdirektori baru. `the`

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

*`(Amazon S3 copy command)`*Dimana salah satu dari berikut ini:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2`untuk Wilayah Timur AS (Ohio)

- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1`untuk Wilayah AS Timur (Virginia N.)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1`untuk Wilayah AS Barat (California Utara)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2`untuk Wilayah Barat AS (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1`untuk Wilayah Kanada (Tengah)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1`untuk Wilayah Eropa (Irlandia)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2`untuk Wilayah Eropa (London)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3`untuk Wilayah Eropa (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1`untuk Wilayah Eropa (Frankfurt)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1`untuk Wilayah Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1`untuk Wilayah Asia Pasifik (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1`untuk Wilayah Asia Pasifik (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2`untuk Wilayah Asia Pasifik (Seoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1`untuk Wilayah Asia Pasifik (Singapura)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2`untuk Wilayah Asia Pasifik (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4`untuk Wilayah Asia Pasifik (Melbourne)

- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1`untuk Wilayah Asia Pasifik (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1`untuk Wilayah Amerika Selatan (São Paulo)

Untuk mendorong revisi Anda sendiri ke instance Server Ubuntu, salin revisi Anda ke repo lokal Anda, lalu panggil yang berikut ini:

```
git add .
git commit -m "Added Ubuntu app"
git push
```

Langkah 4: Menyediakan sebuah instance

Pada langkah ini, Anda akan membuat atau mengonfigurasi instance yang akan Anda gunakan untuk aplikasi sampel. Anda dapat menerapkan ke instans Amazon EC2 atau instans lokal yang menjalankan salah satu sistem operasi yang didukung oleh. CodeDeploy Untuk informasi lihatSistem operasi yang didukung oleh CodeDeploy agen. (Jika Anda sudah memiliki instance yang dikonfigurasi untuk digunakan dalam CodeDeploy penerapan, lewati ke langkah berikutnya.)

Untuk menyediakan sebuah instance

1. Ikuti instruksi Luncurkan instans Amazon EC2 (konsol) untuk menyediakan instance.
2. Saat meluncurkan instance, ingatlah untuk menentukan tag pada halaman Tambahkan tag. Untuk detail tentang cara menentukan tag, lihatLuncurkan instans Amazon EC2 (konsol).

Untuk memverifikasi bahwa CodeDeploy agen berjalan pada instance

- Ikuti instruksi Verifikasi CodeDeploy agen sedang berjalan untuk memverifikasi bahwa agen sedang berjalan.

Setelah Anda berhasil menyediakan instance dan memverifikasi bahwa CodeDeploy agen sedang berjalan, lanjutkan ke langkah berikutnya.

Langkah 5: Buat grup aplikasi dan penyebaran

Pada langkah ini, Anda akan menggunakan CodeDeploy konsol atau AWS CLI untuk membuat aplikasi dan grup penyebaran untuk digunakan untuk menyebarkan revisi sampel dari repositori Anda. GitHub

Buat grup aplikasi dan penyebaran (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pilih Buat aplikasi, lalu pilih Aplikasi kustom.
4. Dalam nama Aplikasi, masukkan**`CodeDeployGitHubDemo-App`**.
5. Di Compute Platform, pilih EC2/On-premises.
6. Pilih Create application (Buat aplikasi).
7. Pada tab Deployment groups, pilih Create deployment group.
8. Dalam nama grup Deployment, masukkan**`CodeDeployGitHubDemo-DepGrp`**.
9. Di Peran layanan, pilih nama peran CodeDeploy layanan yang Anda buat di Buat peran layanan CodeDeploy.
10. Dalam jenis Deployment, pilih In-place.
11. Dalam konfigurasi Lingkungan, tergantung pada jenis instans yang Anda gunakan, pilih instans atau instans Amazon EC2. On-premises Untuk Key dan Value, masukkan kunci tag instance dan nilai yang diterapkan ke instance Anda sebagai bagian dariLangkah 4: Menyediakan sebuah instance.
12. Dalam konfigurasi Deployment, pilih CodeDeployDefault.AllatOnce.
13. Di Load Balancer, hapus Aktifkan penyeimbangan beban.
14. Perluas Lanjutan.

15. Di Alarm, pilih Abaikan konfigurasi alarm.
16. Pilih Buat grup penyebaran, dan lanjutkan ke langkah berikutnya.

Buat grup aplikasi dan penyebaran (CLI)

1. Panggil create-application perintah untuk membuat aplikasi CodeDeploy bernama`CodeDeployGitHubDemo-App`:

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Panggil create-deployment-group perintah untuk membuat grup penyebaran bernama`CodeDeployGitHubDemo-DepGrp`:

   - Jika Anda menerapkan ke instans Amazon EC2*ec2-tag-key*, adalah kunci tag instans Amazon EC2 yang diterapkan ke instans Amazon EC2 Anda sebagai bagian dari. Langkah 4: Menyediakan sebuah instance
   - Jika Anda menerapkan ke instans Amazon EC2*ec2-tag-value*, adalah nilai tag instans Amazon EC2 yang diterapkan ke instans Amazon EC2 Anda sebagai bagian dari. Langkah 4: Menyediakan sebuah instance
   - Jika Anda menerapkan ke instans lokal, *on-premises-tag-key* adalah kunci tag instans lokal yang diterapkan ke instans lokal Anda sebagai bagian dari. Langkah 4: Menyediakan sebuah instance
   - Jika Anda menerapkan ke instans lokal, *on-premises-tag-value* adalah nilai tag instans lokal yang diterapkan ke instans lokal Anda sebagai bagian dari. Langkah 4: Menyediakan sebuah instance
   - *service-role-arn*adalah ARN peran layanan untuk peran layanan yang Anda buat di Buat peran layanan. CodeDeploy (Ikuti instruksi Dapatkan peran layanan ARN (CLI) untuk menemukan peran layanan ARN.)

```
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**
>
> Perintah create-deployment-group menyediakan dukungan untuk membuat pemicu yang menghasilkan pengiriman notifikasi Amazon SNS ke pelanggan topik tentang peristiwa tertentu dalam penerapan dan instance. Perintah ini juga mendukung opsi untuk memutar kembali penerapan secara otomatis dan mengatur alarm untuk menghentikan penerapan saat memantau ambang batas di alarm Amazon terpenuhi. CloudWatch Perintah untuk tindakan ini tidak termasuk dalam tutorial ini.

Langkah 6: Menyebarkan aplikasi ke instance

Pada langkah ini, Anda menggunakan CodeDeploy konsol atau AWS CLI untuk menerapkan revisi sampel dari GitHub repositori Anda ke instance Anda.

Untuk menyebarkan revisi (konsol)

1. Pada halaman Detail grup Deployment, pilih Create deployment.
2. Di grup Deployment, pilih **`CodeDeployGitHubDemo-DepGrp`**.
3. Dalam jenis Revisi, pilih GitHub.
4. Di Connect to GitHub, lakukan salah satu hal berikut:
   - Untuk membuat koneksi untuk CodeDeploy aplikasi ke GitHub akun, keluar dari GitHub tab browser web terpisah. Di GitHub akun, masukkan nama untuk mengidentifikasi koneksi ini, lalu pilih Connect to GitHub. Halaman web meminta Anda untuk mengotorisasi CodeDeploy untuk berinteraksi dengan aplikasi GitHub bernama. `CodeDeployGitHubDemo-App` Lanjutkan ke langkah 5.
   - Untuk menggunakan koneksi yang telah Anda buat, di GitHubakun, pilih namanya, lalu pilih Connect to GitHub. Lanjutkan ke langkah 7.
   - Untuk membuat koneksi ke GitHub akun lain, keluar dari GitHub tab browser web terpisah. Pilih Connect ke GitHub akun lain, lalu pilih Connect to GitHub. Lanjutkan ke langkah 5.
5. Ikuti petunjuk di halaman Masuk untuk masuk dengan GitHub akun Anda.
6. Pada halaman Otorisasi aplikasi, pilih Otorisasi aplikasi.
7. Pada halaman CodeDeploy Create deployment, dalam nama Repositori, masukkan nama GitHub pengguna yang Anda gunakan untuk masuk, diikuti dengan garis miring (`/`), diikuti

dengan nama repositori tempat Anda mendorong revisi aplikasi Anda (misalnya,). ***my-github-user-name*/CodeDeployGitHubDemo**

Jika Anda tidak yakin nilai yang akan dimasukkan, atau jika Anda ingin menentukan repositori yang berbeda:

a. Di tab browser web terpisah, buka GitHub dasbor Anda.

b. Di repositori Anda, arahkan kursor mouse Anda ke nama repositori target. Tooltip muncul, menampilkan nama GitHub pengguna atau organisasi, diikuti dengan garis miring (/), diikuti dengan nama repositori. Masukkan nilai ini ke dalam nama Repositori.

> **Note**
>
> Jika nama repositori target tidak ditampilkan di repositori Anda, gunakan GitHub kotak Pencarian untuk menemukan repositori target dan nama pengguna atau organisasi. GitHub

8. Di kotak Commit ID, masukkan ID komit yang terkait dengan push revisi aplikasi Anda. GitHub

   Jika Anda tidak yakin dengan nilai yang akan dimasukkan:

   a. Di tab browser web terpisah, buka GitHub dasbor Anda.

   b. Di repositori Anda, pilih. CodeDeployGitHubDemo

   c. Dalam daftar komit, temukan dan salin ID komit yang terkait dengan push revisi aplikasi Anda. GitHub ID ini biasanya panjangnya 40 karakter dan terdiri dari huruf dan angka. (Jangan gunakan versi yang lebih pendek dari ID komit, yang biasanya merupakan 10 karakter pertama dari versi yang lebih panjang.)

   d. Rekatkan ID komit ke dalam kotak ID Komit.

9. Pilih Deploy, dan lanjutkan ke langkah berikutnya.

Untuk menyebarkan revisi (CLI)

Sebelum Anda dapat memanggil AWS CLI perintah apa pun yang berinteraksi dengan GitHub (seperti create-deployment perintah, yang akan Anda panggil berikutnya), Anda harus memberikan CodeDeploy izin untuk menggunakan akun GitHub pengguna Anda GitHub untuk berinteraksi dengan `CodeDeployGitHubDemo-App` aplikasi. Saat ini, Anda harus menggunakan CodeDeploy konsol untuk melakukan ini.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pilih CodeDeployGitHubDemo-App.
4. Pada tab Deployment, pilih Create deployment.

   > **Note**
   >
   > Anda tidak akan membuat penyebaran baru. Ini saat ini satu-satunya cara untuk memberikan CodeDeploy izin untuk berinteraksi dengan GitHub atas nama akun GitHub pengguna Anda.

5. Dari grup Deployment, pilih CodeDeployGitHubDemo-DepGrp.
6. Dalam jenis Revisi, pilih GitHub.
7. Di Connect to GitHub, lakukan salah satu hal berikut:

   - Untuk membuat koneksi untuk CodeDeploy aplikasi ke GitHub akun, keluar dari GitHub tab browser web terpisah. Di GitHub akun, ketikkan nama untuk mengidentifikasi koneksi ini, lalu pilih Connect to GitHub. Halaman web meminta Anda untuk mengotorisasi CodeDeploy untuk berinteraksi dengan GitHub aplikasi bernama. `CodeDeployGitHubDemo-App` Lanjutkan ke langkah 8.
   - Untuk menggunakan koneksi yang telah Anda buat, di GitHubakun, pilih namanya, lalu pilih Connect to GitHub. Lanjutkan ke langkah 10.
   - Untuk membuat koneksi ke GitHub akun lain, keluar dari GitHub tab browser web terpisah. Pilih Connect ke GitHub akun lain, lalu pilih Connect to GitHub. Lanjutkan ke langkah 8.

8. Ikuti petunjuk pada halaman Masuk untuk masuk dengan nama GitHub pengguna atau email dan kata sandi Anda.
9. Pada halaman Otorisasi aplikasi, pilih Otorisasi aplikasi.
10. Pada halaman CodeDeploy Create deployment, pilih Cancel.

11. Panggil create-deployment perintah untuk menyebarkan revisi dari GitHub repositori Anda ke instance, di mana:

    - *repository*adalah nama GitHub akun Anda, diikuti dengan garis miring maju (/), diikuti dengan nama repositori Anda (CodeDeployGitHubDemo), misalnya,. MyGitHubUserName/CodeDeployGitHubDemo

      Jika Anda tidak yakin nilai yang akan digunakan, atau jika Anda ingin menentukan repositori yang berbeda:

      1. Di tab browser web terpisah, buka GitHub dasbor Anda.
      2. Di repositori Anda, arahkan kursor mouse Anda ke nama repositori target. Tooltip muncul, menampilkan nama GitHub pengguna atau organisasi, diikuti dengan garis miring (/), diikuti dengan nama repositori. Ini adalah nilai untuk digunakan.

      > **Note**
      >
      > Jika nama repositori target tidak muncul di repositori Anda, gunakan GitHub kotak Pencarian untuk menemukan repositori target dan nama pengguna atau organisasi yang sesuai. GitHub

    - *commit-id*adalah komit yang terkait dengan versi revisi aplikasi yang Anda dorong ke repositori Anda (misalnya,). f835159a...528eb76f

      Jika Anda tidak yakin dengan nilai yang akan digunakan:

      1. Di tab browser web terpisah, buka GitHub dasbor Anda.
      2. Di repositori Anda, pilih. CodeDeployGitHubDemo
      3. Dalam daftar komit, temukan ID komit yang terkait dengan push revisi aplikasi Anda. GitHub ID ini biasanya panjangnya 40 karakter dan terdiri dari huruf dan angka. (Jangan gunakan versi yang lebih pendek dari ID komit, yang biasanya merupakan 10 karakter pertama dari versi yang lebih panjang.) Gunakan nilai ini.

    Jika Anda bekerja pada mesin Linux, macOS, atau Unix lokal:

```
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
```

Jika Anda bekerja pada mesin Windows lokal:

```
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
```

Langkah 7: Pantau dan verifikasi penyebaran

Pada langkah ini, Anda akan menggunakan CodeDeploy konsol atau AWS CLI untuk memverifikasi keberhasilan penerapan. Anda akan menggunakan browser web Anda untuk melihat halaman web yang dikerahkan ke instance yang Anda buat atau konfigurasikan.

> **Note**
>
> Jika Anda menerapkan ke instance Server Ubuntu, gunakan strategi pengujian Anda sendiri untuk menentukan apakah revisi yang diterapkan berfungsi seperti yang diharapkan pada instance, dan kemudian lanjutkan ke langkah berikutnya.

Untuk memantau dan memverifikasi penyebaran (konsol)

1. Di panel navigasi, perluas Deploy, lalu pilih Deployments.
2. Dalam daftar penerapan, cari baris dengan nilai Aplikasi CodeDeployGitHubDemo-Appdan nilai Grup Deployment dari. CodeDeployGitHubDemo-DepGrp Jika Berhasil atau Gagal tidak muncul di kolom Status, pilih tombol Refresh secara berkala.
3. Jika Gagal muncul di kolom Status, ikuti petunjuk Lihat detail contoh (konsol) untuk memecahkan masalah penerapan.
4. Jika Berhasil muncul di kolom Status, Anda sekarang dapat memverifikasi penyebaran melalui browser web Anda. Revisi sampel kami menyebarkan satu halaman web ke instance. Jika Anda menerapkan ke instans Amazon EC2, di browser web Anda, buka `http://public-dns` misalnya (misalnya`http://ec2-01-234-567-890.compute-1.amazonaws.com`,).

5. Jika Anda dapat melihat halaman web, maka selamat! Sekarang setelah Anda AWS CodeDeploy berhasil menggunakan revisi dari GitHub, Anda dapat melompat ke depan. Langkah 8: Membersihkan

Untuk memantau dan memverifikasi penyebaran (CLI)

1. Panggil list-deployments perintah untuk mendapatkan ID penyebaran untuk aplikasi bernama `CodeDeployGitHubDemo-App` dan grup penyebaran bernama: `CodeDeployGitHubDemo-DepGrp`

```
aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --
deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output
 text
```

2. Panggil get-deployment perintah, memasok ID penyebaran dalam output dari perintah: list-deployments

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
[status, creator]" --output text
```

3. Jika Gagal dikembalikan, ikuti instruksi Lihat detail contoh (konsol) untuk memecahkan masalah penerapan.
4. Jika Succeeded dikembalikan, Anda sekarang dapat mencoba memverifikasi penyebaran melalui browser web Anda. Revisi sampel kami adalah satu halaman web yang digunakan untuk instance. Jika Anda menerapkan ke instans Amazon EC2, Anda dapat melihat halaman ini di browser web Anda dengan membuka `http://public-dns` instans Amazon EC2 (misalnya,). `http://ec2-01-234-567-890.compute-1.amazonaws.com`
5. Jika Anda dapat melihat halaman web, maka selamat! Anda telah berhasil digunakan AWS CodeDeploy untuk menyebarkan dari GitHub repositori Anda.

Langkah 8: Membersihkan

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, Anda harus menghentikan instans Amazon EC2 dan sumber daya yang terkait. Secara opsional, Anda dapat menghapus catatan komponen CodeDeploy penyebaran yang terkait dengan tutorial ini. Jika Anda menggunakan GitHub repositori hanya untuk tutorial ini, Anda dapat menghapusnya sekarang juga.

Untuk menghapus CloudFormation stack (jika Anda menggunakan CloudFormation template untuk membuat instans Amazon EC2)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di kolom Stacks, pilih tumpukan yang dimulai dengan`CodeDeploySampleStack`.
3. Pilih Hapus.
4. Saat diminta, pilih Hapus tumpukan. Instans Amazon EC2 dan profil instans IAM terkait serta peran layanan dihapus.

Untuk membatalkan pendaftaran dan membersihkan instans lokal secara manual (jika Anda menyediakan instans lokal)

1. Gunakan perintah AWS CLI untuk memanggil perintah deregister terhadap instance lokal yang diwakili di sini oleh *`your-instance-name`* dan wilayah terkait dengan: *`your-region`*

   ```
   aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --
   region your-region
   ```

2. Dari instance lokal, panggil perintah uninstall:

   ```
   aws deploy uninstall
   ```

Untuk menghentikan instans Amazon EC2 secara manual (jika Anda meluncurkan instans Amazon EC2 secara manual)

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi, di Instans, pilih Instans.
3. Pilih kotak di sebelah instans Amazon EC2 yang ingin Anda hentikan. Di menu Actions, arahkan ke Instance State, lalu pilih Terminate.
4. Saat diminta, pilih Ya, Hentikan.

Untuk menghapus catatan komponen CodeDeploy penyebaran

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.

3. Pilih CodeDeployGitHubDemo-App.
4. Pilih Hapus aplikasi.
5. Saat diminta, masukkan **`Delete`**, lalu pilih Hapus.

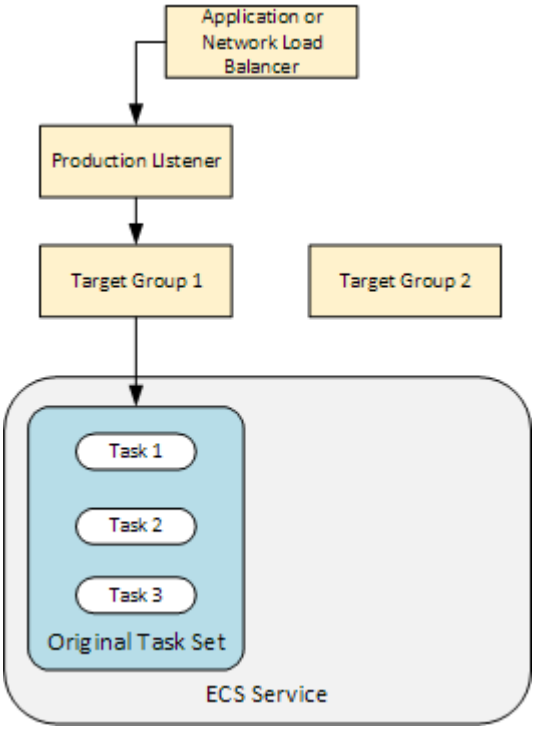
Untuk menghapus GitHub repositori Anda

Lihat Menghapus repositori dalam bantuan. GitHub

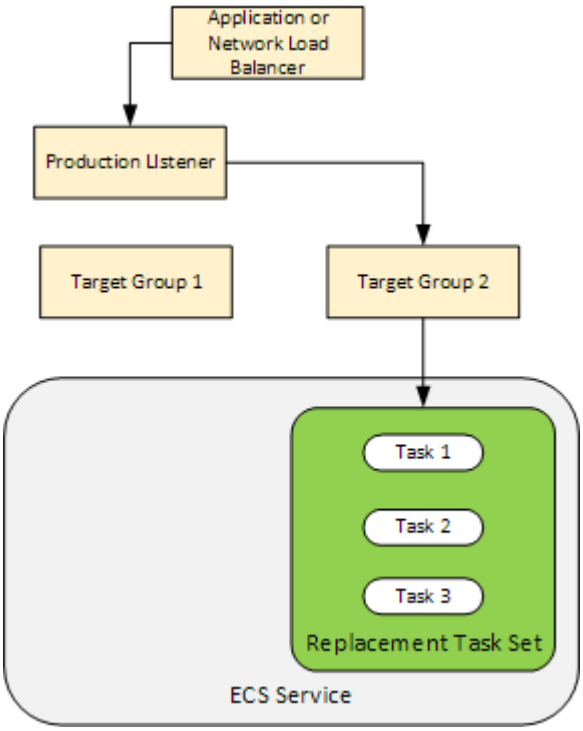
Tutorial: Menyebarkan aplikasi ke Amazon ECS

Dalam tutorial ini, Anda mempelajari cara menyebarkan aplikasi ke Amazon ECS menggunakan. CodeDeploy Anda mulai dengan aplikasi yang sudah Anda buat dan gunakan ke Amazon ECS. Langkah pertama adalah memperbarui aplikasi Anda dengan memodifikasi file definisi tugasnya dengan tag baru. Selanjutnya, Anda gunakan CodeDeploy untuk menyebarkan pembaruan. Selama penerapan, CodeDeploy instal pembaruan Anda ke set tugas pengganti baru. Kemudian, itu menggeser lalu lintas produksi dari versi asli aplikasi Amazon ECS Anda, yang ada dalam set tugas aslinya, ke versi terbaru dalam set tugas pengganti.

Selama penyebaran Amazon ECS, CodeDeploy gunakan penyeimbang beban yang dikonfigurasi dengan dua grup target dan satu pendengar lalu lintas produksi. Diagram berikut menunjukkan bagaimana penyeimbang beban, pendengar produksi, grup target, dan aplikasi Amazon ECS Anda terkait sebelum penerapan dimulai. Tutorial ini menggunakan Application Load Balancer. Anda juga dapat menggunakan Network Load Balancer.



Setelah penerapan berhasil, pendengar lalu lintas produksi menyajikan lalu lintas ke set tugas pengganti baru Anda dan set tugas asli dihentikan. Diagram berikut menunjukkan bagaimana sumber daya Anda terkait setelah penerapan berhasil. Untuk informasi selengkapnya, lihat Apa yang terjadi selama penyebaran Amazon ECS.



Untuk informasi tentang cara menggunakan aplikasi AWS CLI untuk menyebarkan aplikasi ke Amazon ECS, lihat Tutorial: Membuat layanan menggunakan penerapan. blue/green Untuk informasi tentang cara menggunakan CodePipeline untuk mendeteksi dan secara otomatis menerapkan perubahan ke layanan Amazon ECS dengan CodeDeploy, lihat Tutorial: Membuat pipeline dengan sumber dan penerapan Amazon ECR. ECS-to-CodeDeploy

Setelah Anda menyelesaikan tutorial ini, Anda dapat menggunakan CodeDeploy aplikasi dan grup penyebaran yang Anda buat untuk menambahkan uji validasi penerapan. Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi

Topik

- Prasyarat
- Langkah 1: Perbarui aplikasi Amazon ECS Anda
- Langkah 2: Buat AppSpec file
- Langkah 3: Gunakan CodeDeploy konsol untuk menyebarkan aplikasi Anda
- Langkah 4: Membersihkan

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Selesaikan langkah 2 dan 3 diMemulai dengan CodeDeploy.
- Buat Application Load Balancer yang dikonfigurasi dengan dua grup target dan satu pendengar. Untuk informasi tentang membuat penyeimbang beban menggunakan konsol, lihatSiapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS. Untuk informasi tentang membuat penyeimbang beban menggunakan AWS CLI, lihat Langkah 1: Membuat Application Load Balancer di Panduan Pengguna Amazon Elastic Container Service. Saat Anda membuat penyeimbang beban, buat catatan berikut untuk tutorial ini:
  - Nama penyeimbang beban Anda.
  - Nama-nama kelompok sasaran Anda.
  - Port yang digunakan oleh pendengar penyeimbang beban Anda.
- Buat kluster dan layanan Amazon ECS. Untuk informasi selengkapnya, lihat langkah 2, 3, dan 4 di Tutorial: Membuat layanan menggunakan blue/green penerapan di Panduan Pengguna Layanan Amazon Elastic Container. Catat hal-hal berikut untuk tutorial ini:
  - Nama cluster Amazon ECS Anda.

- ARN definisi tugas yang digunakan oleh layanan Amazon ECS Anda.
- Nama wadah yang digunakan oleh layanan Amazon ECS Anda.

- Buat bucket Amazon S3 untuk file Anda AppSpec .

Langkah 1: Perbarui aplikasi Amazon ECS Anda

Di bagian ini, Anda memperbarui aplikasi Amazon ECS Anda dengan revisi baru definisi tugasnya. Revisi yang diperbarui menambahkan pasangan kunci dan tag baru. DiLangkah 3: Gunakan CodeDeploy konsol untuk menyebarkan aplikasi Anda, Anda menerapkan versi terbaru dari aplikasi Amazon ECS Anda.

Untuk memperbarui definisi tugas

1. Buka konsol di https://console.aws.amazon.com/ecs/v2.
2. Di panel navigasi, pilih Ketentuan Tugas.
3. Pilih definisi tugas yang digunakan oleh layanan Amazon ECS Anda.
4. Pilih revisi definisi tugas, lalu pilih Buat revisi baru, Buat revisi baru.
5. Untuk tutorial ini, buat pembaruan kecil untuk definisi tugas dengan menambahkan tag. Di bagian bawah halaman, di bawah Tag, buat tag baru dengan memasukkan pasangan kunci dan nilai baru.
6. Pilih Buat.

   Nomor revisi definisi tugas Anda bertambah satu.
7. Pilih tab JSON. Catat hal-hal berikut karena Anda memerlukan informasi ini di langkah berikutnya.

   - Nilai untuk`taskDefinitionArn`. Formatnya adalah `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`*. Ini adalah ARN dari definisi tugas Anda yang diperbarui.
   - Dalam `containerDefinitions` elemen, nilai untuk`name`. Ini adalah nama wadah Anda.
   - Dalam `portMappings` elemen, nilai untuk`containerPort`. Ini adalah port untuk kontainer Anda.

Langkah 2: Buat AppSpec file

Di bagian ini, Anda membuat AppSpec file dan mengunggahnya ke bucket Amazon S3 yang Anda buat di bagian tersebutPrasyarat. AppSpec File untuk penerapan Amazon ECS menentukan definisi tugas, nama kontainer, dan port kontainer Anda. Untuk informasi selengkapnya, lihat AppSpec Contoh file untuk penyebaran Amazon ECS dan AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

Untuk membuat AppSpec file Anda

1. Jika Anda ingin membuat AppSpec file Anda menggunakan YAMG, buat file bernama`appspec.yml`. Jika Anda ingin membuat AppSpec file Anda menggunakan JSON, buat file bernama`appspec.json`.
2. Pilih tab yang sesuai, tergantung pada apakah Anda menggunakan YAMAL atau JSON untuk AppSpec file Anda, dan salin kontennya ke AppSpec file yang baru saja Anda buat. Untuk `TaskDefinition` properti, gunakan definisi tugas ARN yang Anda catat di Langkah 1: Perbarui aplikasi Amazon ECS Anda bagian.

   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**
>
> Set tugas pengganti Anda mewarisi subnet, grup keamanan, versi platform, dan nilai IP publik yang ditetapkan dari set tugas asli Anda. Anda dapat mengganti nilai ini untuk tugas pengganti yang ditetapkan dengan menyetel properti opsionalnya di AppSpec file Anda. Untuk informasi selengkapnya, lihat AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS dan AppSpec Contoh file untuk penyebaran Amazon ECS.

3. Unggah AppSpec file Anda ke bucket S3 yang Anda buat sebagai prasyarat untuk tutorial ini.

Langkah 3: Gunakan CodeDeploy konsol untuk menyebarkan aplikasi Anda

Di bagian ini, Anda membuat grup CodeDeploy aplikasi dan penyebaran untuk menyebarkan aplikasi yang diperbarui ke Amazon ECS. Selama penerapan, CodeDeploy menggeser lalu lintas produksi untuk aplikasi Anda ke versi barunya dalam set tugas pengganti yang baru. Untuk menyelesaikan langkah ini, Anda memerlukan item berikut:

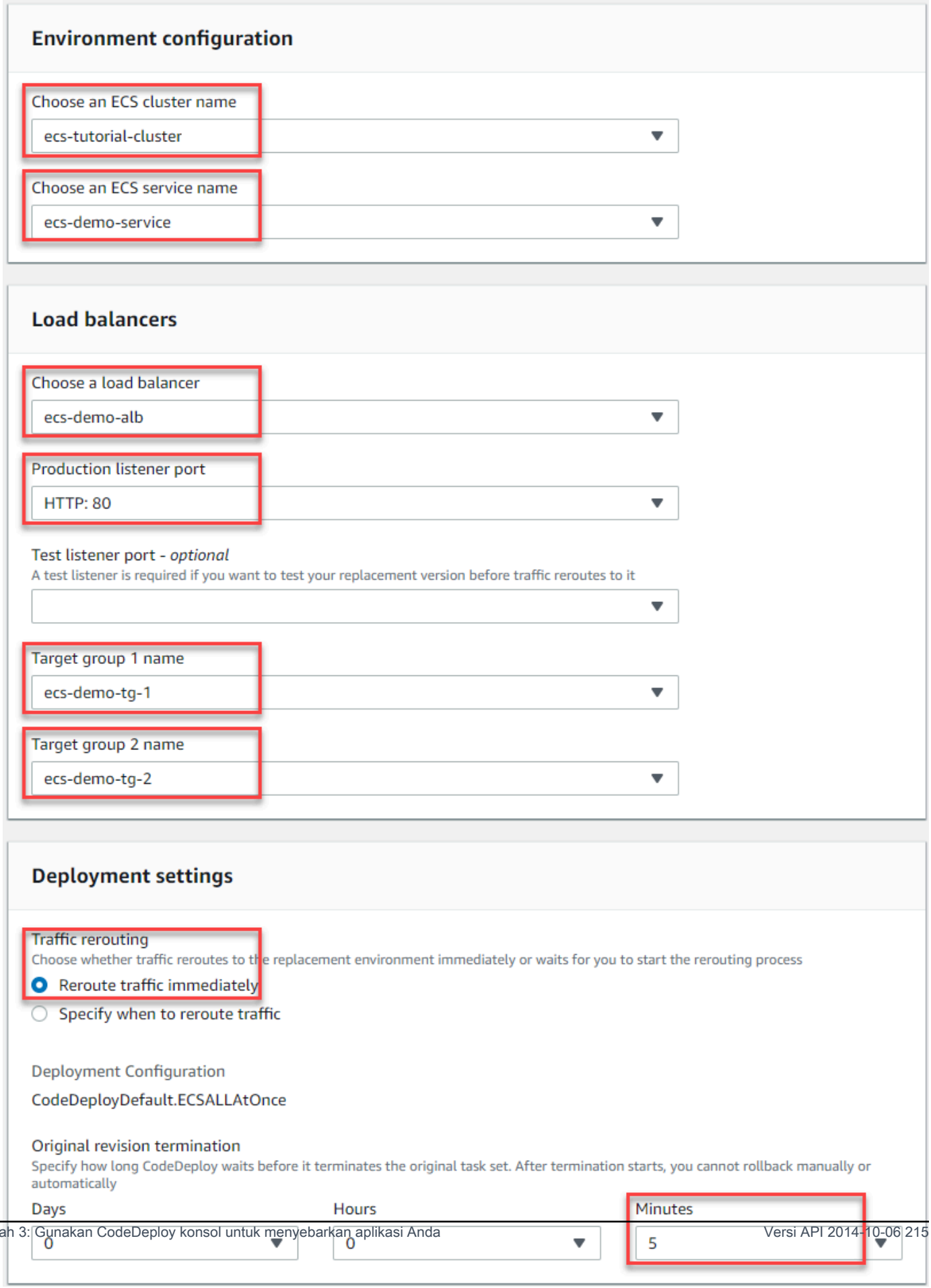
- Nama cluster Amazon ECS Anda.
- Nama layanan Amazon ECS Anda.
- Nama Application Load Balancer Anda.
- Port pendengar produksi Anda.
- Nama grup target Anda.
- Nama bucket S3 yang Anda buat.

Untuk membuat CodeDeploy aplikasi

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
2. Pilih Create application (Buat aplikasi).
3. Dalam nama Aplikasi, masukkan**`ecs-demo-codedeploy-app`**.
4. Di platform Compute, pilih Amazon ECS.
5. Pilih Create application (Buat aplikasi).

Untuk membuat CodeDeploy grup penyebaran

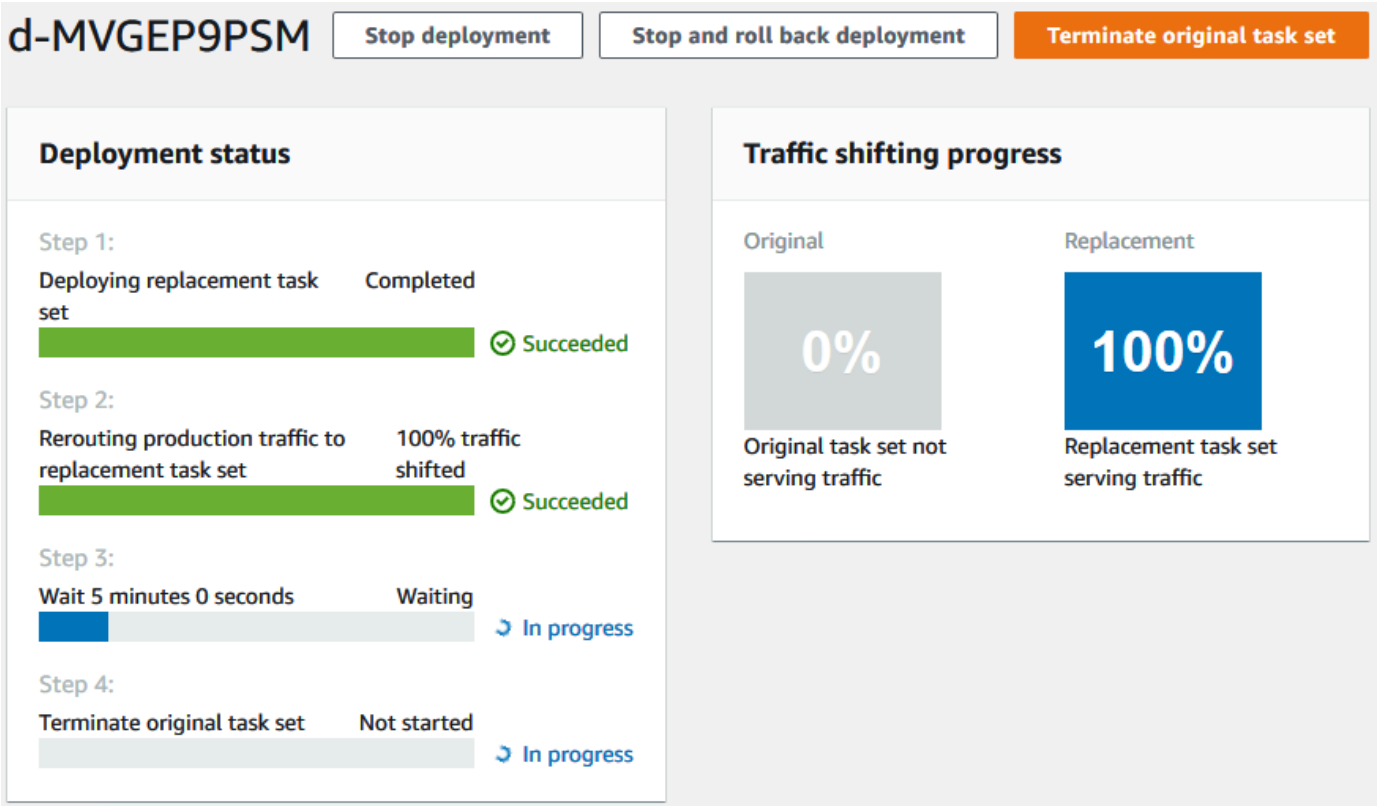
1. Pada tab grup Deployment di halaman aplikasi Anda, pilih Buat grup penyebaran.
2. Dalam nama grup Deployment, masukkan**`ecs-demo-dg`**.
3. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke Amazon ECS. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk AWS CodeDeploy.
4. Dalam konfigurasi Lingkungan, pilih nama cluster Amazon ECS dan nama layanan.
5. Dari Load balancer, pilih nama penyeimbang beban yang melayani lalu lintas ke layanan Amazon ECS Anda.
6. Dari port pendengar Produksi, pilih port dan protokol untuk pendengar yang menyajikan lalu lintas produksi ke layanan Amazon ECS Anda (misalnya, HTTP: 80). Tutorial ini tidak menyertakan pendengar uji opsional, jadi jangan memilih port dari port pendengar Uji.
7. Dari nama grup target 1 dan nama grup target 2, pilih dua grup target yang berbeda untuk merutekan lalu lintas selama penyebaran Anda. Pastikan bahwa ini adalah kelompok target yang Anda buat untuk penyeimbang beban Anda. Tidak masalah mana yang digunakan untuk kelompok target 1 dan yang digunakan untuk kelompok sasaran 2.
8. Pilih Reroute lalu lintas segera.
9. Untuk penghentian revisi Asli, pilih 0 hari, 0 jam, dan 5 menit. Ini memungkinkan Anda melihat penerapan Anda selesai lebih cepat daripada jika Anda menggunakan default (1 jam).


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. Pilih Buat grup penyebaran.

Untuk menyebarkan aplikasi Amazon ECS Anda

1. Dari halaman konsol grup penerapan Anda, pilih Buat penerapan.
2. Untuk grup Deployment, pilih ecs-demo-dg.
3. Untuk jenis Revisi, pilih Aplikasi saya disimpan di Amazon S3. Di lokasi Revisi, masukkan nama bucket S3 Anda.
4. Untuk jenis file Revisi, pilih .json atau.yaml, yang sesuai.
5. (Opsional) Dalam deskripsi Deployment, masukkan deskripsi untuk penerapan Anda.
6. Pilih Buat penerapan.
7. Dalam status Deployment, Anda dapat memantau penerapan. Setelah 100% lalu lintas produksi dialihkan ke set tugas pengganti dan sebelum waktu tunggu lima menit berakhir, Anda dapat memilih Hentikan set tugas asli untuk segera mengakhiri set tugas asli. Jika Anda tidak memilih Hentikan set tugas asli, set tugas asli akan berakhir setelah waktu tunggu lima menit yang Anda tentukan berakhir.



Langkah 4: Membersihkan

Tutorial berikutnya,Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi, dibangun di atas tutorial ini dan menggunakan CodeDeploy aplikasi dan kelompok penyebaran yang Anda buat. Jika Anda ingin mengikuti langkah-langkah dalam tutorial itu, lewati langkah ini dan jangan hapus sumber daya yang Anda buat.

> **Note**
>
> AWS Akun Anda tidak dikenakan biaya untuk CodeDeploy sumber daya yang Anda buat.

Nama sumber daya dalam langkah-langkah ini adalah nama yang disarankan dalam tutorial ini (**`ecs-demo-codedeploy-app`**misalnya, untuk nama CodeDeploy aplikasi Anda). Jika Anda menggunakan nama yang berbeda, maka pastikan untuk menggunakannya selama pembersihan Anda.

1. Gunakan perintah delete-deployment-group untuk menghapus grup penyebaran. CodeDeploy

   ```
   aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --
   deployment-group-name ecs-demo-dg --region aws-region-id
   ```

2. Gunakan perintah hapus-aplikasi untuk menghapus aplikasi. CodeDeploy

   ```
   aws deploy delete-application --application-name ecs-demo-codedeploy-app --
   region aws-region-id
   ```

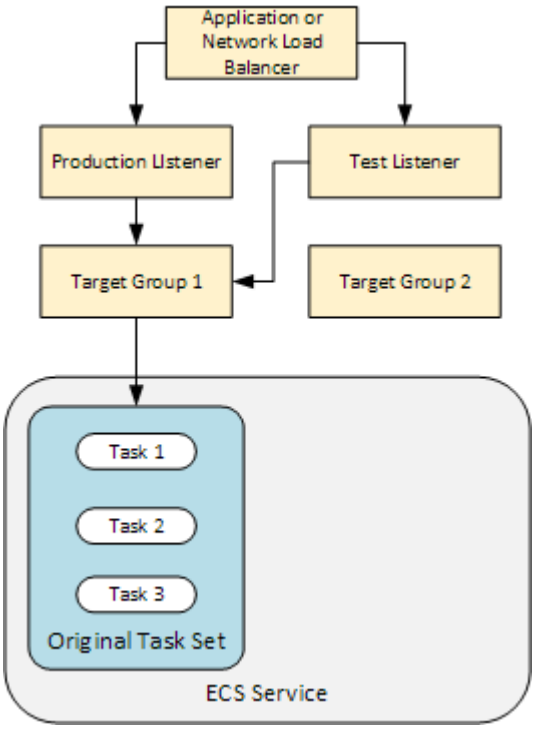
Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi

Dalam tutorial ini, Anda mempelajari cara menggunakan fungsi Lambda untuk memvalidasi bagian dari penerapan aplikasi Amazon ECS yang diperbarui. Tutorial ini menggunakan CodeDeploy aplikasi, grup CodeDeploy penyebaran, dan aplikasi Amazon ECS yang Anda gunakan. Tutorial: Menyebarkan aplikasi ke Amazon ECS Selesaikan tutorial itu sebelum memulai yang satu ini.

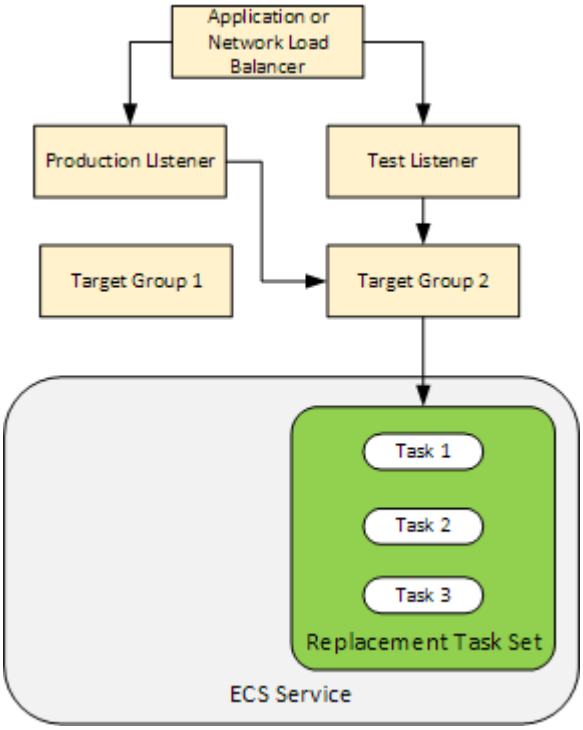
Untuk menambahkan uji validasi, pertama-tama Anda menerapkan pengujian dalam fungsi Lambda. Selanjutnya, dalam AppSpec file deployment Anda, Anda menentukan fungsi Lambda untuk hook siklus hidup yang ingin Anda uji. Jika tes validasi gagal, penerapan berhenti, diputar kembali, dan

ditandai gagal. Jika pengujian berhasil, penerapan akan berlanjut ke event atau hook siklus hidup penerapan berikutnya.

Selama penerapan Amazon ECS dengan pengujian validasi, CodeDeploy gunakan penyeimbang beban yang dikonfigurasi dengan dua grup target: satu pendengar lalu lintas produksi dan satu pendengar lalu lintas pengujian. Diagram berikut menunjukkan bagaimana penyeimbang beban, pendengar produksi dan pengujian, grup target, dan aplikasi Amazon ECS Anda terkait sebelum penerapan dimulai. Tutorial ini menggunakan Application Load Balancer. Anda juga dapat menggunakan Network Load Balancer.



Selama penerapan Amazon ECS, ada lima kait siklus hidup untuk pengujian. Tutorial ini mengimplementasikan satu pengujian selama hook penerapan siklus hidup ketiga,. `AfterAllowTestTraffic` Untuk informasi selengkapnya, lihat Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS. Setelah penerapan berhasil, pendengar lalu lintas produksi menyajikan lalu lintas ke set tugas penggantian baru Anda dan set tugas asli dihentikan. Diagram berikut menunjukkan bagaimana sumber daya Anda terkait setelah penerapan berhasil. Untuk informasi selengkapnya, lihat Apa yang terjadi selama penyebaran Amazon ECS.



> **Note**
>
> Menyelesaikan tutorial ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeDeploy, AWS Lambda, dan CloudWatch. Untuk informasi selengkapnya, lihat AWS CodeDeploy harga, AWS Lambda harga, dan CloudWatch harga Amazon.

Topik

- Prasyarat
- Langkah 1: Buat pendengar tes
- Langkah 2: Perbarui aplikasi Amazon ECS Anda
- Langkah 3: Buat fungsi Lambda kait siklus hidup
- Langkah 4: Perbarui AppSpec file Anda
- Langkah 5: Gunakan CodeDeploy konsol untuk menyebarkan layanan Amazon ECS Anda
- Langkah 6: Lihat output fungsi kait Lambda Anda di Log CloudWatch
- Langkah 7: Bersihkan

Prasyarat

Untuk berhasil menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Lengkapi prasyarat untuk. Prasyarat Tutorial: Menyebarkan aplikasi ke Amazon ECS
- Selesaikan langkah-langkah dalam Tutorial: Menyebarkan aplikasi ke Amazon ECS. Catat hal-hal berikut:
  - Nama penyeimbang beban Anda.
  - Nama-nama kelompok sasaran Anda.
  - Port yang digunakan oleh pendengar penyeimbang beban Anda.
  - ARN penyeimbang beban Anda. Anda menggunakan ini untuk membuat pendengar baru.
  - ARN dari salah satu kelompok sasaran Anda. Anda menggunakan ini untuk membuat pendengar baru.
  - Grup CodeDeploy aplikasi dan penyebaran yang Anda buat.
  - AppSpec File yang Anda buat yang digunakan oleh CodeDeploy penyebaran Anda. Anda mengedit file ini dalam tutorial ini.

Langkah 1: Buat pendengar tes

Penerapan Amazon ECS dengan pengujian validasi memerlukan pendengar kedua. Listener ini digunakan untuk menyajikan lalu lintas pengujian ke aplikasi Amazon ECS Anda yang diperbarui dalam set tugas pengganti. Tes validasi Anda dijalankan terhadap lalu lintas pengujian.

Listener untuk lalu lintas pengujian Anda dapat menggunakan salah satu grup target Anda. Gunakan AWS CLI perintah create-listener untuk membuat pendengar kedua dengan aturan default yang meneruskan lalu lintas pengujian ke port 8080. Gunakan ARN penyeimbang beban Anda dan ARN dari salah satu grup target Anda.

```
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
```

Langkah 2: Perbarui aplikasi Amazon ECS Anda

Di bagian ini, Anda memperbarui aplikasi Amazon ECS Anda untuk menggunakan revisi baru dari definisi tugasnya. Anda membuat revisi baru dan menambahkan pembaruan kecil ke dalamnya dengan menambahkan tag.

Untuk memperbarui definisi tugas Anda

1. Buka konsol klasik Amazon ECS di https://console.aws.amazon.com/ecs/.
2. Di panel navigasi, pilih Ketentuan Tugas.
3. Pilih kotak centang untuk definisi tugas yang digunakan oleh layanan Amazon ECS Anda.
4. Pilih Buat revisi baru.
5. Buat pembaruan kecil untuk definisi tugas dengan menambahkan tag. Di bagian bawah halaman, di Tag, buat tag baru dengan memasukkan pasangan kunci dan nilai baru.
6. Pilih Buat. Anda akan melihat bahwa nomor revisi definisi tugas Anda telah bertambah satu.
7. Pilih tab JSON. Catat nilainya`taskDefinitionArn`. Formatnya adalah `arn:aws:ecs:`*`aws-region`*`: `*`account-id`*`:task-definition/`*`task-definition-family`*`: `*`task-definition-revision`*. Ini adalah ARN dari definisi tugas Anda yang diperbarui.

Langkah 3: Buat fungsi Lambda kait siklus hidup

Di bagian ini, Anda menerapkan satu fungsi Lambda untuk hook penyebaran Amazon ECS Anda. `AfterAllowTestTraffic` Fungsi Lambda menjalankan uji validasi sebelum aplikasi Amazon ECS yang diperbarui diinstal. Untuk tutorial ini, fungsi Lambda kembali. `Succeeded` Selama penerapan dunia nyata, tes validasi kembali `Succeeded` atau`Failed`, tergantung pada hasil tes validasi. Juga selama penerapan dunia nyata, Anda dapat menerapkan fungsi pengujian Lambda untuk satu atau beberapa kait `BeforeInstall` peristiwa siklus hidup penerapan Amazon ECS lainnya (,,, dan). `AfterInstall` `BeforeAllowTraffic` `AfterAllowTraffic` Untuk informasi selengkapnya, lihat Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS.

Peran IAM diperlukan untuk membuat fungsi Lambda Anda. Peran tersebut memberikan izin fungsi Lambda untuk menulis CloudWatch ke Log dan menyetel status hook siklus hidup CodeDeploy .

Untuk membuat IAM role

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Dari panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Buat peran dengan properti berikut:

   - Entitas tepercaya: AWS Lambda.
   - Izin: AWSLambdaBasicExecutionRole. Ini memberikan izin fungsi Lambda Anda untuk menulis CloudWatch ke Log.
   - Nama peran: **`lambda-cli-hook-role`**.

   Untuk informasi selengkapnya, lihat Membuat peran AWS Lambda eksekusi.
4. Lampirkan izin `codedeploy:PutLifecycleEventHookExecutionStatus` ke peran yang Anda buat. Ini memberikan izin fungsi Lambda Anda untuk menyetel status hook siklus CodeDeploy hidup selama penerapan. Untuk informasi selengkapnya, lihat Menambahkan izin identitas IAM di Panduan AWS Identity and Access Management Pengguna dan PutLifecycleEventHookExecutionStatusdi Referensi CodeDeploy API.

Untuk membuat fungsi Lambda **`AfterAllowTestTraffic`**hook

1. Buat file bernama `AfterAllowTestTraffic.js` dengan isi berikut ini.

```
'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. Buat paket penyebaran Lambda.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Gunakan `create-function` perintah untuk membuat fungsi Lambda untuk hook Anda`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. Catat ARN fungsi Lambda Anda dalam tanggapannya. `create-function` Anda menggunakan ARN ini saat memperbarui AppSpec file CodeDeploy penerapan di langkah berikutnya.

Langkah 4: Perbarui AppSpec file Anda

Di bagian ini, Anda memperbarui AppSpec file Anda dengan `Hooks` bagian. Di `Hooks` bagian ini, Anda menentukan fungsi Lambda untuk kait siklus `AfterAllowTestTraffic` hidup.

Untuk memperbarui AppSpec file Anda

1. Buka AppSpec file file yang Anda buat Langkah 2: Buat AppSpec file dari fileTutorial: Menyebarkan aplikasi ke Amazon ECS.
2. Perbarui `TaskDefinition` properti dengan definisi tugas ARN yang Anda catat. Langkah 2: Perbarui aplikasi Amazon ECS Anda
3. Salin dan tempel `Hooks` bagian ke AppSpec file file Anda. Perbarui ARN setelahnya `AfterAllowTestTraffic` dengan ARN dari fungsi Lambda yang Anda catat. Langkah 3: Buat fungsi Lambda kait siklus hidup

   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. Simpan AppSpec file Anda dan unggah ke bucket S3-nya.

Langkah 5: Gunakan CodeDeploy konsol untuk menyebarkan layanan Amazon ECS Anda

Di bagian ini, Anda memperbarui grup penerapan dengan menentukan port untuk listener pengujian Anda. Ini adalah pendengar yang Anda buat. Langkah 1: Buat pendengar tes Selama penerapan, CodeDeploy jalankan pengujian validasi Anda selama hook siklus hidup `AfterAllowTestTraffic` penerapan menggunakan lalu lintas pengujian yang disajikan ke set tugas pengganti Anda menggunakan listener pengujian. Tes validasi Anda mengembalikan hasilnya`Succeeded`, sehingga penerapan dilanjutkan dengan peristiwa siklus hidup penerapan berikutnya. Dalam skenario dunia nyata, fungsi pengujian Anda kembali `Succeeded` atau`Failed`.

Untuk menambahkan listener pengujian ke grup penerapan

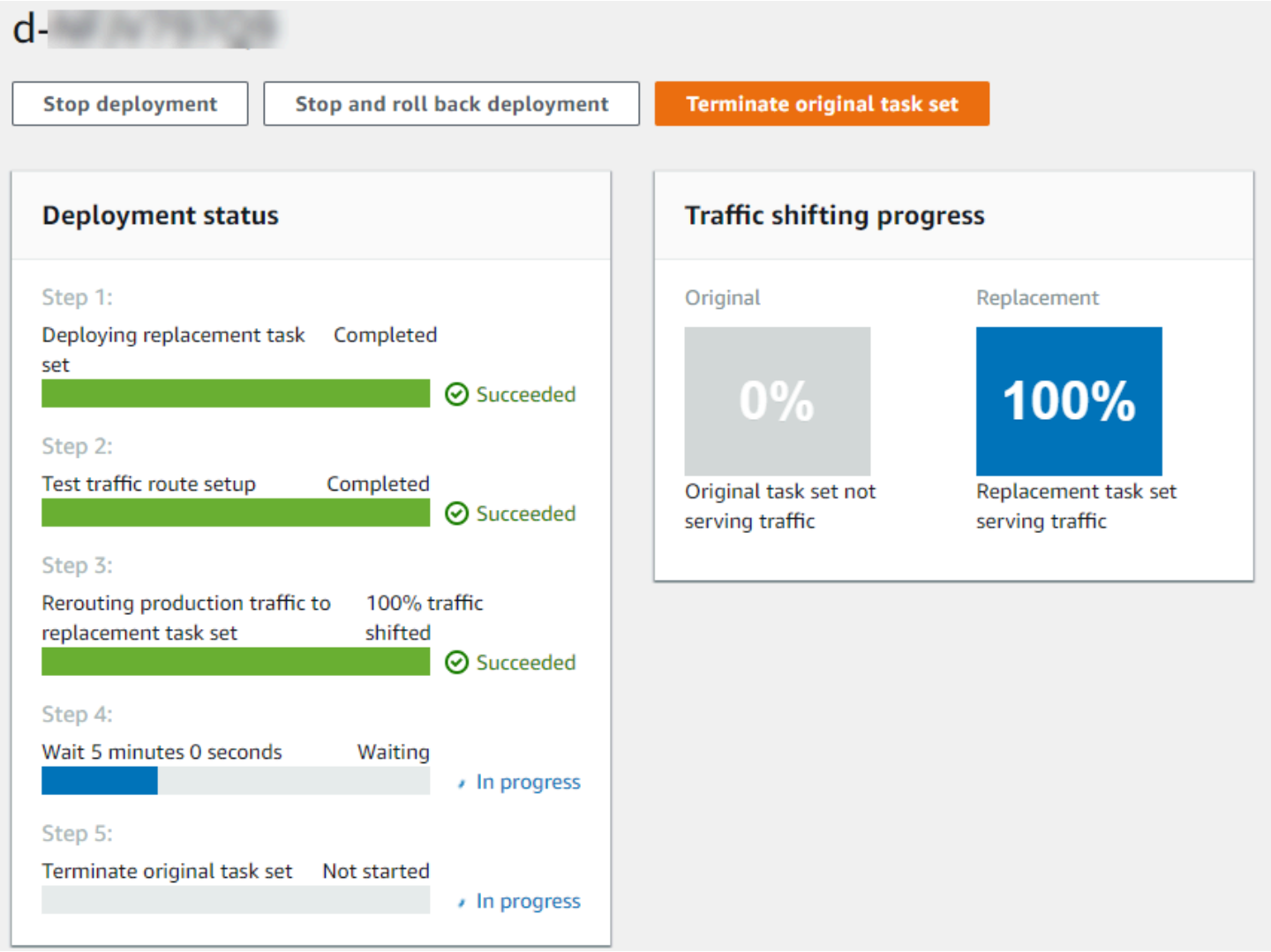
1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
2. Dari panel navigasi, pilih Aplikasi.
3. Pilih aplikasi yang Anda buatTutorial: Menyebarkan aplikasi ke Amazon ECS. Jika Anda menggunakan nama yang disarankan, itu adalah ecs-demo-codedeploy-app.
4. Di grup Deployment, pilih grup penyebaran yang Anda buat. Tutorial: Menyebarkan aplikasi ke Amazon ECS Jika Anda menggunakan nama yang disarankan, itu adalah ecs-demo-dg.

5. Pilih Edit.
6. Dari port Test listener, pilih port dan protokol untuk test listener yang Anda buat sebelumnya dalam tutorial ini. Ini harus HTTP: 8080.
7. Pilih Simpan perubahan.

Untuk menyebarkan aplikasi Amazon ECS Anda

1. Dari halaman konsol grup penerapan Anda, pilih Buat penerapan.
2. Untuk grup Deployment, pilih ecs-demo-dg.
3. Untuk jenis Revisi, pilih Aplikasi saya disimpan di Amazon S3. Di lokasi Revisi, masukkan nama bucket dan AppSpec file S3 Anda (misalnya,**`s3://my-s3-bucket/appspec.json`**).
4. Untuk jenis file Revisi, pilih .json atau.yaml yang sesuai.
5. (Opsional) Dalam deskripsi Deployment, masukkan deskripsi untuk penerapan Anda.
6. Pilih Buat penerapan.

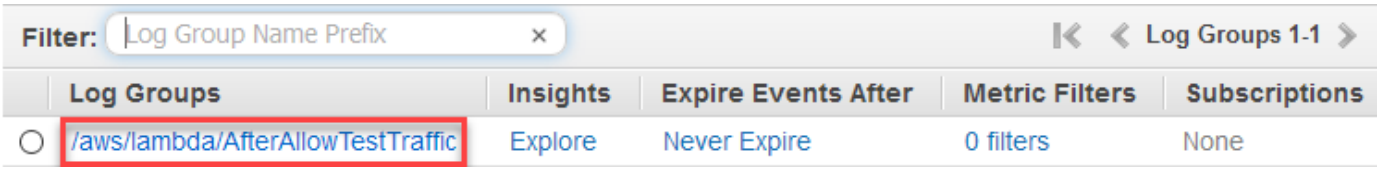
Anda dapat memantau penerapan dalam status Deployment. Setelah 100% lalu lintas produksi dirutekan ke set tugas pengganti, Anda dapat memilih Hentikan set tugas asli untuk segera menghentikan set tugas asli. Jika Anda tidak memilih Menghentikan set tugas asli, set tugas asli akan berakhir setelah durasi yang Anda tentukan saat membuat grup penerapan.



Langkah 6: Lihat output fungsi kait Lambda Anda di Log CloudWatch

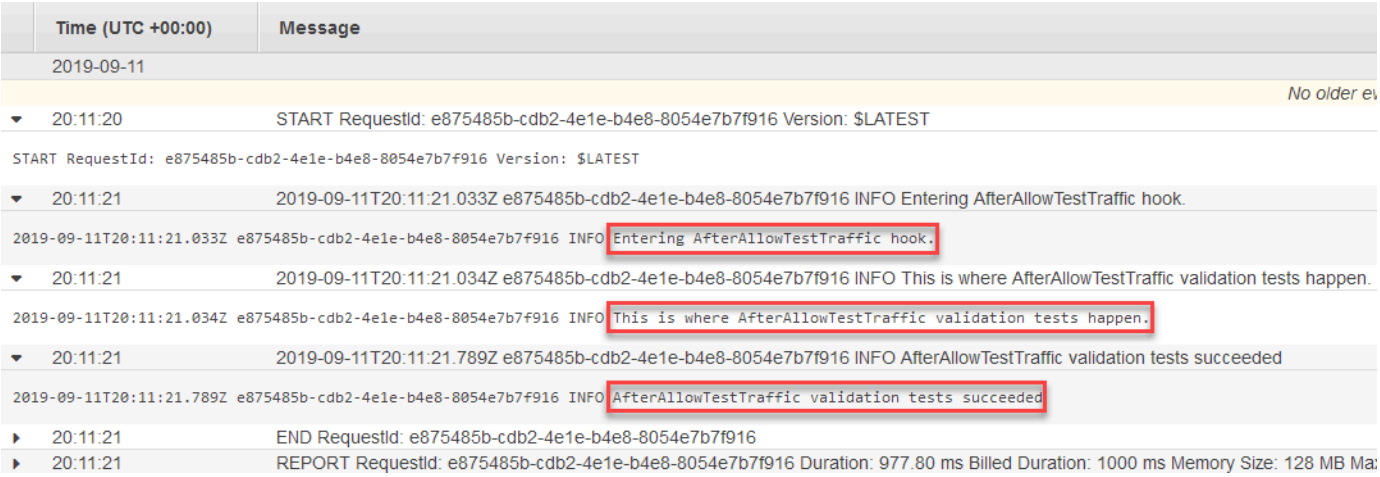
Jika CodeDeploy penerapan Anda berhasil, tes validasi di fungsi kait Lambda Anda juga berhasil. Anda dapat mengonfirmasi ini dengan melihat log untuk fungsi hook di CloudWatch Log.

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Dari panel navigasi, pilih Log. Anda akan melihat satu grup log baru untuk fungsi kait Lambda yang Anda tentukan dalam file Anda AppSpec .



3. Pilih grup log baru. Ini harus/aws/lambda/AfterAllowTestTrafficHook.

4. Pilih pengaliran log. Jika Anda melihat lebih dari satu aliran log, pilih salah satu dengan tanggal dan waktu terbaru di bawah Waktu Acara Terakhir.
5. Perluas peristiwa aliran log untuk mengonfirmasi fungsi kait Lambda Anda menulis pesan sukses ke log. Berikut ini menunjukkan fungsi kait `AfterAllowTraffic` Lambda berhasil.



Langkah 7: Bersihkan

Bila Anda telah menyelesaikan tutorial ini, bersihkan sumber daya yang terkait dengannya untuk menghindari biaya untuk sumber daya yang tidak Anda gunakan. Nama sumber daya dalam langkah ini adalah nama-nama yang disarankan dalam tutorial ini (**`ecs-demo-codedeploy-app`**misalnya, untuk nama CodeDeploy aplikasi Anda). Jika Anda menggunakan nama yang berbeda, maka pastikan untuk menggunakannya dalam pembersihan Anda.

Untuk membersihkan sumber daya tutorial

1. Gunakan perintah delete-deployment-group untuk menghapus grup penyebaran. CodeDeploy

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Gunakan perintah hapus-aplikasi untuk menghapus aplikasi. CodeDeploy

```
aws deploy delete-application --application-name ecs-demo-deployment-group --
region aws-region-id
```

3. Gunakan perintah delete-function untuk menghapus fungsi hook Lambda Anda.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

4. Gunakan perintah delete-log-group untuk menghapus grup log Anda. CloudWatch

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutorial: Menyebarkan fungsi CodeDeploy Lambda yang diperbarui dengan dan AWS Serverless Application Model

AWS SAM adalah kerangka kerja sumber terbuka untuk membangun aplikasi tanpa server. Ini mengubah dan memperluas sintaks YAMAL dalam AWS SAM template menjadi CloudFormation sintaks untuk membangun aplikasi tanpa server, seperti fungsi Lambda. Untuk informasi selengkapnya, lihat Apa itu AWS Model Aplikasi Tanpa Server?

Dalam tutorial ini, Anda menggunakan AWS SAM untuk membuat solusi yang melakukan hal berikut:

- Membuat fungsi Lambda Anda.
- Membuat grup CodeDeploy aplikasi dan penyebaran Anda.
- Membuat dua fungsi Lambda yang menjalankan tes validasi penerapan selama kait siklus hidup. CodeDeploy
- Mendeteksi kapan fungsi Lambda Anda diperbarui. Pembaruan fungsi Lambda memicu penerapan yang secara bertahap menggeser lalu lintas produksi dari versi asli fungsi Lambda Anda ke versi terbaru. CodeDeploy

> **Note**
>
> Tutorial ini mengharuskan Anda untuk membuat sumber daya yang mungkin menimbulkan biaya untuk akun AWS Anda. Ini termasuk kemungkinan biaya untuk CodeDeploy, Amazon CloudWatch, dan AWS Lambda. Untuk informasi selengkapnya, lihat CodeDeploy harga, CloudWatch harga Amazon, dan AWS Lambda harga.

Topik

- Prasyarat

- Langkah 1: Siapkan infrastruktur Anda
- Langkah 2: Perbarui fungsi Lambda
- Langkah 3: Menyebarkan fungsi Lambda yang diperbarui
- Langkah 4: Lihat hasil penerapan
- Langkah 5: Bersihkan

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Selesaikan langkah-langkah dalam Memulai dengan CodeDeploy.
- Instal AWS Serverless Application Model CLI. Untuk selengkapnya, lihat Menginstal AWS SAM CLI.
- Buat ember S3. AWS SAM mengunggah artefak yang direferensikan dalam template AWS SAM Anda ke dalam ember ini.

Langkah 1: Siapkan infrastruktur Anda

Topik ini menunjukkan cara menggunakan AWS SAM untuk membuat file untuk AWS SAM template dan fungsi Lambda Anda. Kemudian, Anda menggunakan `deploy` perintah AWS SAM `package` dan untuk menghasilkan komponen dalam infrastruktur Anda. Ketika infrastruktur sudah siap, Anda memiliki grup CodeDeploy aplikasi dan penyebaran, fungsi Lambda untuk memperbarui dan menyebarkan, dan dua fungsi Lambda yang berisi pengujian validasi yang berjalan saat Anda menerapkan fungsi Lambda. Setelah selesai, Anda dapat menggunakan CloudFormation untuk melihat komponen Anda di konsol Lambda atau AWS CLI untuk menguji fungsi Lambda Anda.

Topik

- Buat file Anda
- Package AWS Aplikasi SAM
- Menyebarkan AWS Aplikasi SAM
- (Opsional) memeriksa dan menguji infrastruktur Anda

Buat file Anda

Untuk membuat infrastruktur Anda, Anda harus membuat file berikut:

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Topik

- Buat Anda AWS Templat SAM
- Buat file untuk fungsi Lambda Anda
- Buat file untuk fungsi BeforeAllowTraffic Lambda Anda
- Buat file untuk fungsi AfterAllowTraffic Lambda Anda

Buat Anda AWS Templat SAM

Buat file template AWS SAM yang menentukan komponen dalam infrastruktur Anda.

Untuk membuat AWS Templat SAM

1. Membuat sebuah direktori bernama `SAM-Tutorial`.
2. Di `SAM-Tutorial` direktori Anda, buat file bernama`template.yml`.
3. Salin kode YAML berikut ke dalam`template.yml`. Ini adalah AWS SAM template Anda.

```
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
```

Template ini menentukan berikut ini. Untuk informasi selengkapnya, lihat konsep AWS SAM templat.

Fungsi Lambda yang disebut **`myDateTimeFunction`**

Ketika fungsi Lambda ini diterbitkan, `AutoPublishAlias` baris dalam template menautkannya ke alias bernama. `live` Kemudian dalam tutorial ini, pembaruan untuk fungsi ini memicu penerapan yang secara bertahap menggeser lalu lintas produksi dari versi asli ke versi yang diperbarui. AWS CodeDeploy

Dua fungsi validasi penerapan Lambda

Fungsi Lambda berikut dijalankan selama kait siklus CodeDeploy hidup. Fungsi berisi kode yang memvalidasi penyebaran yang diperbarui. `myDateTimeFunction` Hasil tes validasi diteruskan CodeDeploy menggunakan metode `PutLifecycleEventHookExecutionStatus` API-nya. Jika tes validasi gagal, penerapan gagal dan dibatalkan.

- `CodeDeployHook_beforeAllowTraffic`berjalan selama `BeforeAllowTraffic` pengait.
- `CodeDeployHook_afterAllowTraffic`berjalan selama `AfterAllowTraffic` pengait.

Nama kedua fungsi dimulai dengan`CodeDeployHook_`. `CodeDeployRoleForLambda`Peran memungkinkan panggilan ke `invoke` metode Lambda hanya dalam fungsi Lambda dengan nama yang dimulai dengan awalan ini. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk sebuah AWS Penyebaran Lambda dan PutLifecycleEventHookExecutionStatusdi Referensi CodeDeploy API.

Deteksi otomatis fungsi Lambda yang diperbarui

`AutoPublishAlias`Istilah ini memberi tahu kerangka kerja untuk mendeteksi kapan `myDateTimeFunction` fungsi berubah, dan kemudian menerapkannya menggunakan `live` alias.

Konfigurasi penerapan

Konfigurasi deployment menentukan tingkat di mana CodeDeploy aplikasi Anda mengalihkan lalu lintas dari versi asli fungsi Lambda ke versi baru. Template ini menentukan konfigurasi penerapan yang telah ditentukan. `Linear10PercentEvery1Minute`

> **Note**
>
> Anda tidak dapat menentukan konfigurasi penerapan kustom dalam template AWS SAM. Untuk informasi selengkapnya, lihat Create a Deployment Configuration.

Fungsi kait siklus hidup penerapan

`Hooks`Bagian ini menentukan fungsi yang akan dijalankan selama kait peristiwa siklus hidup. `PreTraffic`menentukan fungsi yang berjalan selama `BeforeAllowTraffic` hook. `PostTraffic`menentukan fungsi yang berjalan selama `AfterAllowTraffic` hook.

Izin untuk Lambda untuk memanggil fungsi Lambda lain

`lambda:InvokeFunction`Izin yang ditentukan memberikan peran yang digunakan oleh izin aplikasi AWS SAM untuk memanggil fungsi Lambda. Ini diperlukan saat `CodeDeployHook_afterAllowTraffic` fungsi `CodeDeployHook_beforeAllowTraffic` and memanggil fungsi Lambda yang diterapkan selama pengujian validasi.

Buat file untuk fungsi Lambda Anda

Buat file untuk fungsi yang Anda perbarui dan terapkan nanti dalam tutorial ini.

> **Note**
>
> Fungsi Lambda dapat menggunakan runtime apa pun yang didukung oleh. AWS Lambda Untuk informasi selengkapnya, lihat runtime AWS Lambda.

Untuk membuat fungsi Lambda Anda

1. Buat file teks dan simpan seperti `myDateTimeFunction.js` di `SAM-Tutorial` direktori.
2. Salin Node.js kode berikut ke dalam`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;
    }
  };
```

Fungsi Lambda mengembalikan hari, bulan, dan tahun untuk kemarin, hari ini, atau besok. Kemudian dalam tutorial ini, Anda menghapus kode komentar yang memperbarui fungsi untuk mengembalikan informasi tentang hari atau waktu yang Anda tentukan (misalnya, hari, bulan, dan tahun, atau jam,

menit, dan detik saat ini). Kerangka kerja yang dibuat oleh AWS SAM mendeteksi dan menyebarkan versi fungsi yang diperbarui.

> **Note**
>
> Fungsi Lambda ini juga digunakan dalam tutorial. AWS Cloud9 AWS Cloud9 adalah lingkungan pengembangan terintegrasi berbasis cloud. Untuk informasi tentang cara membuat, mengeksekusi, memperbarui, dan men-debug fungsi ini AWS Cloud9, lihat AWS Lambda tutorial untuk AWS Cloud9.

Buat file untuk fungsi BeforeAllowTraffic Lambda Anda

Buat file untuk fungsi Lambda `beforeAllowTraffic` hook Anda.

1. Buat file teks dan simpan seperti `beforeAllowTraffic.js` di `SAM-Tutorial` direktori.
2. Salin Node.js kode berikut ke dalam`beforeAllowTraffic.js`. Fungsi ini dijalankan selama hook penerapan Anda. `BeforeAllowTraffic`

```
'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");
        }
       });
      }
     });
    }
```

Buat file untuk fungsi AfterAllowTraffic Lambda Anda

Buat file untuk fungsi Lambda `afterAllowTraffic` hook Anda.

1. Buat file teks dan simpan seperti `afterAllowTraffic.js` di `SAM-Tutorial` direktori.
2. Salin Node.js kode berikut ke dalam`afterAllowTraffic.js`. Fungsi ini dijalankan selama hook penerapan Anda. `AfterAllowTraffic`

```
'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 AWS Aplikasi SAM

Anda sekarang harus memiliki empat file di `SAM-Tutorial` direktori Anda:

- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`
- `myDateTimeFunction.js`
- `template.yml`

Anda sekarang siap menggunakan sam package perintah AWS SAM untuk membuat dan mengemas artefak untuk fungsi CodeDeploy dan aplikasi Lambda Anda. Artefak diunggah ke ember S3. Output dari perintah adalah file baru bernama`package.yml`. File ini digunakan oleh sam deploy perintah AWS SAM pada langkah berikutnya.

> **Note**
>
> Untuk informasi selengkapnya tentang sam package perintah, lihat referensi perintah AWS SAM CLI di Panduan AWS Serverless Application Model Pengembang.

Di `SAM-Tutorial` direktori, jalankan yang berikut ini.

```
sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket amzn-s3-demo-bucket
```

Untuk `s3-bucket` parameternya, tentukan bucket Amazon S3 yang Anda buat sebagai prasyarat untuk tutorial ini. `output-template-file`Menentukan nama file baru yang digunakan oleh sam deploy perintah AWS SAM.

Menyebarkan AWS Aplikasi SAM

Gunakan sam deploy perintah AWS SAM dengan `package.yml` file untuk membuat fungsi Lambda Anda dan CodeDeploy aplikasi dan kelompok penyebaran menggunakan. CloudFormation

> **Note**
>
> Untuk informasi selengkapnya tentang sam deploy perintah, lihat referensi perintah AWS SAM CLI di Panduan AWS Serverless Application Model Pengembang.

Dalam `SAM-Tutorial` direktori, jalankan perintah berikut.

```
sam deploy \
  --template-file package.yml \
  --stack-name my-date-time-app \
  --capabilities CAPABILITY_IAM
```

`--capabilities CAPABILITY_IAM`Parameter ini diperlukan untuk mengotorisasi CloudFormation untuk membuat peran IAM.

(Opsional) memeriksa dan menguji infrastruktur Anda

Topik ini menunjukkan cara melihat komponen infrastruktur Anda dan menguji fungsi Lambda Anda.

Untuk melihat hasil tumpukan Anda setelah Anda menjalankan **`sam deploy`**

1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di panel navigasi, pilih Stacks (Tumpukan). `my-date-time-app`Tumpukan muncul di bagian atas.
3. Pilih tab Acara untuk melihat acara mana yang selesai. Anda dapat melihat peristiwa saat pembuatan tumpukan sedang berlangsung. Saat pembuatan tumpukan selesai, Anda dapat melihat semua peristiwa pembuatan tumpukan.
4. Dengan tumpukan Anda dipilih, pilih Sumber Daya. Di kolom Type, Anda dapat melihat fungsi Lambda Anda,, `myDateTimeFunction`, `CodeDeployHook_beforeAllowTraffic`, dan.

`CodeDeployHook_afterAllowTraffic` Kolom ID Fisik dari setiap fungsi Lambda Anda berisi tautan untuk melihat fungsi di konsol Lambda.

> **Note**
>
> Nama fungsi `myDateTimeFunction` Lambda ditambahkan dengan nama CloudFormation tumpukan dan memiliki pengenal yang ditambahkan ke dalamnya, sehingga terlihat seperti. `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. Buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
6. Di panel navigasi, perluas Deploy, lalu pilih Applications.
7. Anda akan melihat CodeDeploy aplikasi baru yang dibuat CloudFormation dengan nama yang dimulai dengan`my-date-time-app-ServerlessDeploymentApplication`. Pilih aplikasi ini.
8. Anda akan melihat grup penyebaran dengan nama yang dimulai dengan`my-date-time-app-myDateTimeFunctionDeploymentGroup`. Pilih grup penyebaran ini.

   Di bawah konfigurasi Deployment, Anda akan melihat CodeDeployDefault.LambdaLinear10PercentEvery1Minute.

(Opsional) untuk menguji fungsi Anda (konsol)

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Dari panel navigasi, pilih `my-date-time-app-myDateTimeFunction` fungsi Anda. Di konsol, namanya berisi pengidentifikasi, jadi sepertinya`my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Pilih Uji.
4. Dalam nama Acara, masukkan nama untuk acara pengujian Anda.
5. Masukkan yang berikut ini untuk acara pengujian Anda, lalu pilih Buat.

```
{
  "option": "date",
  "period": "today"
}
```

6. Pilih Uji. Anda hanya akan melihat acara pengujian Anda dalam daftar acara pengujian.

Untuk hasil Eksekusi, Anda akan melihat berhasil.

7. Di bawah Hasil eksekusi, perluas Detail untuk melihat hasilnya. Anda harus melihat bulan, hari, dan tahun saat ini.

(Opsional) untuk menguji fungsi Anda (AWS CLI)

1. Temukan ARN fungsi Lambda Anda. Itu muncul di bagian atas konsol Lambda saat Anda melihat fungsi Anda.
2. Jalankan perintah berikut. Ganti *your-function-arn* dengan fungsi ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload "{\"option\": \"date\", \"period\": \"today\"}" out.txt
```

3. Buka `out.txt` untuk mengonfirmasi hasilnya berisi bulan, hari, dan tahun saat ini.

Langkah 2: Perbarui fungsi Lambda

Dalam topik ini, Anda memperbarui `myDateTimeFunction.js` file Anda. Pada langkah berikutnya, Anda menggunakan file untuk menyebarkan fungsi yang diperbarui. Ini memicu CodeDeploy untuk menerapkannya dengan menggeser lalu lintas produksi dari versi fungsi Lambda saat ini ke versi yang diperbarui.

Untuk memperbarui fungsi Lambda Anda

1. Buka `myDateTimeFunction.js`.
2. Hapus dua penanda komentar (" `/*` "dan" `*/` ") dan teks penjelasan di awal dan akhir yang `case` disebutkan `time` di `switch` blok.

   Kode yang tidak dikomentari memungkinkan Anda meneruskan parameter baru,`time`, ke fungsi. Jika Anda meneruskan `time` ke fungsi yang diperbarui, ia mengembalikan arus`hour`,`minute`, dan`second`.
3. Simpan `myDateTimeFunction.js`. Seharusnya terlihat seperti berikut:

```
'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;
  }
};
```

Langkah 3: Menyebarkan fungsi Lambda yang diperbarui

Pada langkah ini, Anda menggunakan pembaruan `myDateTimeFunction.js` untuk memperbarui dan memulai penerapan fungsi Lambda Anda. Anda dapat memantau kemajuan penerapan di AWS Lambda konsol CodeDeploy atau.

`AutoPublishAlias: live`Baris di AWS SAM template Anda menyebabkan infrastruktur Anda mendeteksi pembaruan ke fungsi yang menggunakan `live` alias. Pembaruan pada fungsi Anda memicu penerapan yang menggeser lalu lintas produksi dari versi asli fungsi ke versi yang diperbarui. CodeDeploy

sam deployPerintah sam package dan digunakan untuk memperbarui dan memicu penyebaran fungsi Lambda Anda. Anda menjalankan perintah ini di Package AWS Aplikasi SAM dan Menyebarkan AWS Aplikasi SAM.

Untuk menerapkan fungsi Lambda Anda yang diperbarui

1. Dalam `SAM-Tutorial` direktori, jalankan perintah berikut.

   ```
   sam package \
     --template-file template.yml \
     --output-template-file package.yml  \
     --s3-bucket amzn-s3-demo-bucket
   ```

   Ini membuat kumpulan artefak baru yang mereferensikan fungsi Lambda Anda yang diperbarui di bucket S3 Anda.

2. Dalam `SAM-Tutorial` direktori, jalankan perintah berikut.

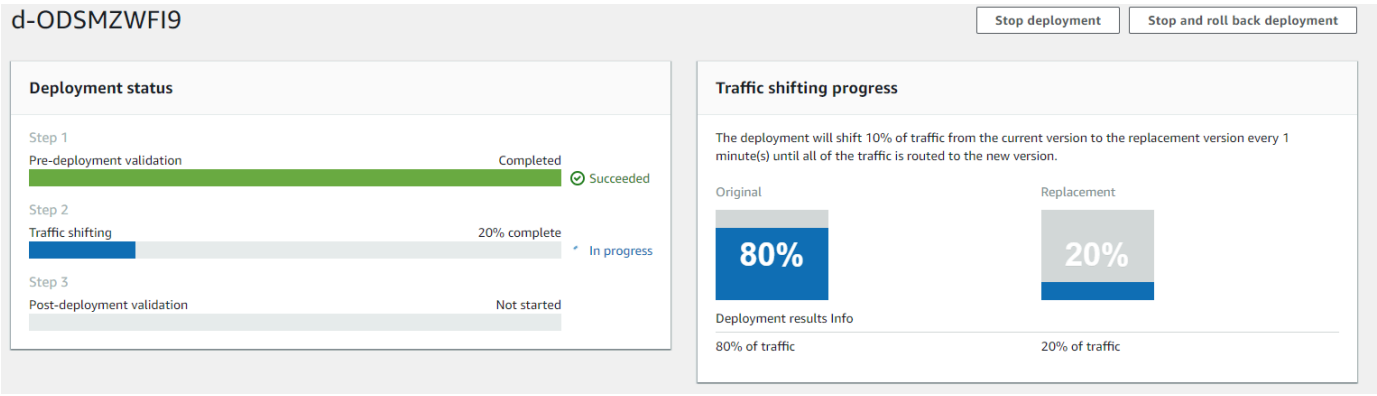
   ```
   sam deploy \
     --template-file package.yml \
     --stack-name my-date-time-app \
     --capabilities CAPABILITY_IAM
   ```

Karena nama tumpukan masih`my-date-time-app`, CloudFormation mengakui bahwa ini adalah pembaruan tumpukan. Untuk melihat tumpukan yang diperbarui, kembalikan CloudFormation konsol, dan dari panel navigasi, pilih Tumpukan.

(Opsional) untuk melihat lalu lintas selama penerapan (CodeDeploy konsol)

1. Buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
2. Di panel navigasi, perluas Aplikasi, lalu pilih aplikasi my-date-time-app - Anda. ServerlessDeploymentApplication
3. Di grup Deployment, pilih grup penerapan aplikasi Anda. Statusnya harus Sedang berlangsung.
4. Dalam riwayat grup Deployment, pilih penerapan yang sedang berlangsung.

   Bilah kemajuan pergeseran lalu lintas dan persentase di kotak Asli dan Penggantian di halaman ini menampilkan kemajuannya.

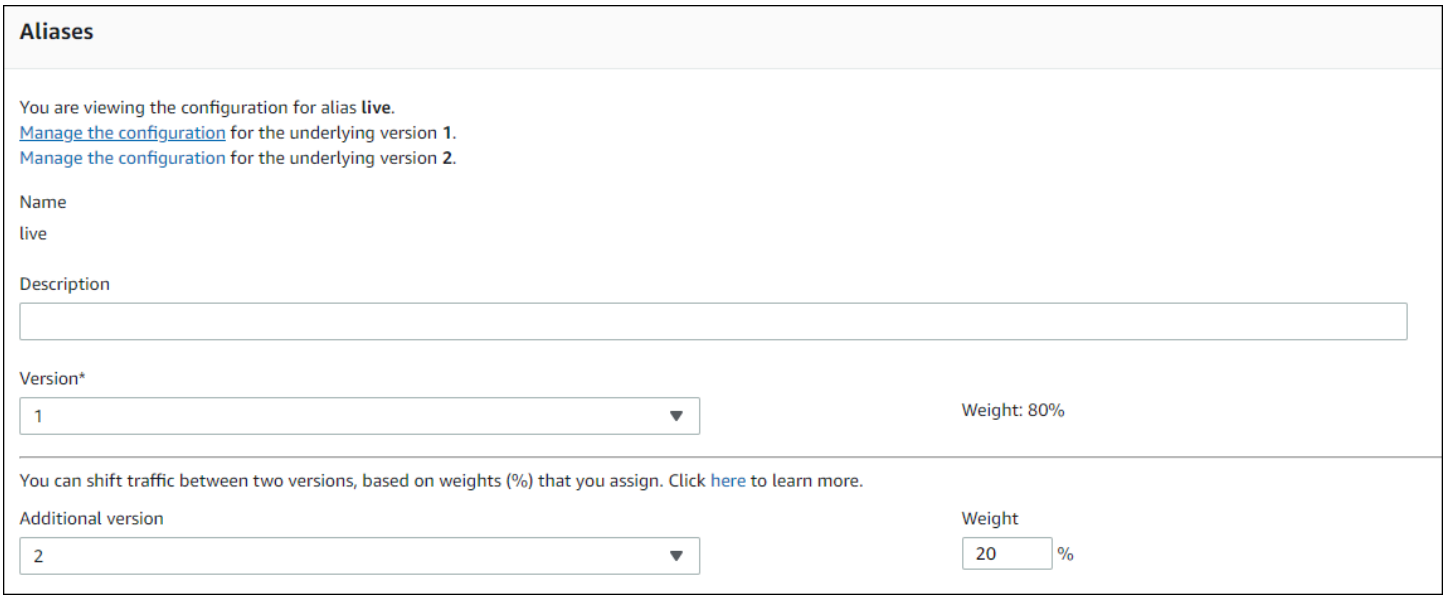


(Opsional) untuk melihat lalu lintas selama penerapan (konsol Lambda)

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Dari panel navigasi, pilih `my-date-time-app-myDateTimeFunction` fungsi Anda. Di konsol, namanya berisi pengidentifikasi, jadi sepertinya`my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Pilih Alias, lalu pilih live.

Bobot di sebelah versi fungsi asli Anda (versi 1) dan versi fungsi Anda yang diperbarui (versi 2) menunjukkan berapa banyak lalu lintas yang disajikan ke setiap versi pada saat halaman AWS Lambda konsol ini dimuat. Halaman tidak memperbarui bobot dari waktu ke waktu. Jika Anda me-

refresh halaman sekali dalam satu menit, berat untuk versi 1 berkurang 10 persen dan berat untuk versi 2 meningkat 10 persen hingga bobot untuk versi 2 adalah 100.



Langkah 4: Lihat hasil penerapan

Pada langkah ini, Anda melihat hasil penerapan Anda. Jika penerapan berhasil, Anda dapat mengonfirmasi bahwa fungsi Lambda yang diperbarui menerima lalu lintas produksi. Jika penerapan gagal, Anda dapat menggunakan CloudWatch Log untuk melihat output pengujian validasi dalam fungsi Lambda yang berjalan selama hoook siklus hidup penerapan Anda.

Topik

- Uji fungsi yang Anda gunakan
- Lihat peristiwa hook di CloudWatch Log

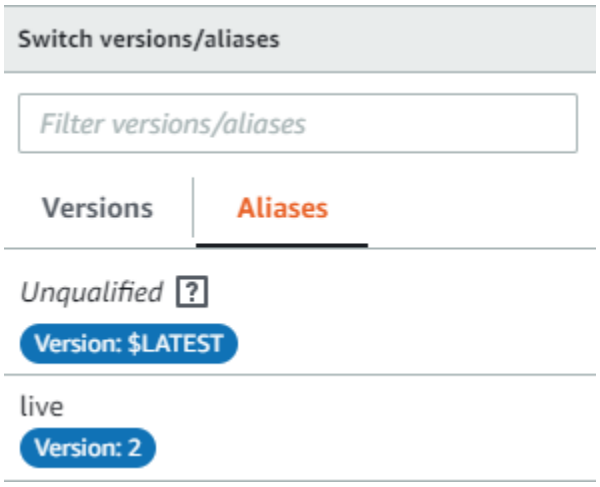
Uji fungsi yang Anda gunakan

sam deployPerintah memperbarui fungsi `my-date-time-app-myDateTimeFunction` Lambda. Versi fungsi diperbarui ke 2 dan ditambahkan ke `live` alias.

Untuk melihat pembaruan di konsol Lambda

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Dari panel navigasi, pilih `my-date-time-app-myDateTimeFunction` fungsinya. Di konsol, namanya berisi pengidentifikasi, jadi sepertinya`my-date-time-app-myDateTimeFunction-123456ABCDEF`.

3. Pilih Kualifikasi, lalu pilih Alias. Setelah penerapan selesai (sekitar 10 menit), untuk `live` alias, Anda akan melihat Versi: 2.



4. Dalam kode Fungsi, lihat kode sumber untuk fungsi Anda. Perubahan Anda akan muncul.
5. (Opsional) Anda dapat menggunakan instruksi pengujian Langkah 2: Perbarui fungsi Lambda untuk menguji fungsi Anda yang diperbarui. Buat acara pengujian baru dengan muatan berikut, lalu konfirmasikan hasilnya berisi jam, menit, dan detik saat ini.

```
{
    "option": "time"
  }
```

Untuk menggunakan fungsi AWS CLI to test yang diperbarui, jalankan perintah berikut, lalu buka `out.txt` untuk mengonfirmasi hasilnya berisi jam, menit, dan detik saat ini.

```
aws lambda invoke --function your-function-arn --payload "{\"option\": \"time\"}"
 out.txt
```

> **Note**
>
> Jika Anda menggunakan AWS CLI untuk menguji fungsi Anda sebelum penerapan selesai, Anda mungkin menerima hasil yang tidak terduga. Ini karena CodeDeploy secara bertahap menggeser 10 persen lalu lintas ke versi yang diperbarui setiap menit. Selama penyebaran, beberapa lalu lintas masih menunjuk ke versi aslinya, jadi `aws lambda invoke` mungkin menggunakan versi aslinya. Setelah 10 menit, penyebaran selesai dan semua lalu lintas menunjuk ke versi baru fungsi.

Lihat peristiwa hook di CloudWatch Log

Selama `BeforeAllowTraffic` hook, CodeDeploy jalankan fungsi `CodeDeployHook_beforeAllowTraffic` Lambda Anda. Selama `AfterAllowTraffic` hook, CodeDeploy jalankan fungsi `CodeDeployHook_afterAllowTraffic` Lambda Anda. Setiap fungsi menjalankan uji validasi yang memanggil versi terbaru dari fungsi Anda menggunakan parameter baru`time`. Jika pembaruan fungsi Lambda Anda berhasil, `time` opsi tidak menyebabkan kesalahan dan validasi berhasil. Jika fungsi tidak diperbarui, parameter yang tidak dikenal menyebabkan kesalahan dan validasi gagal. Tes validasi ini hanya untuk tujuan demonstrasi. Anda menulis pengujian Anda sendiri untuk memvalidasi penerapan Anda. Anda dapat menggunakan konsol CloudWatch Log untuk melihat pengujian validasi Anda.

Untuk melihat acara CodeDeploy hook Anda

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Dari panel navigasi, pilih Log.
3. Dari daftar grup log, pilih/aws/lambda/CodeDeployHook_before AllowTraffic atau/aws/lambda/CodeDeployHookAllowTraffic_after.
4. Pilih pengaliran log. Anda harus melihat hanya satu.
5. Perluas acara untuk melihat detailnya.



Langkah 5: Bersihkan

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat oleh AWS SAM template Anda dan CloudWatch log yang dibuat oleh fungsi validasi Lambda Anda.

Untuk menghapus CloudFormation tumpukan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di kolom Stacks, pilih `my-date-time-app` tumpukan Anda, lalu pilih Hapus.
3. Saat diminta, pilih Hapus tumpukan. Fungsi Lambda, grup CodeDeploy aplikasi dan penyebaran, dan peran IAM yang dibuat oleh dihapus. AWS SAM

Untuk menghapus log Anda di CloudWatch Log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Dari panel navigasi, pilih Log.
3. Dari daftar grup log, pilih tombol di sebelah/aws/lambda/CodeDeployHook_before AllowTraffic.
4. Dari Tindakan, pilih Hapus grup log, lalu pilih Ya, Hapus.
5. Dari daftar grup log, pilih tombol di sebelah/aws/lambda/CodeDeployHook_after AllowTraffic.
6. Dari Tindakan, pilih Hapus grup log, lalu pilih Ya, Hapus.

Bekerja dengan CodeDeploy agen

AWS CodeDeploy Agen adalah paket perangkat lunak yang, ketika diinstal dan dikonfigurasi pada sebuah instance, memungkinkan instance itu digunakan dalam CodeDeploy penerapan.

AWS secara aktif mendukung versi minor terbaru dari CodeDeploy agen. Untuk versi terbaru yang dirilis, lihatRiwayat versi CodeDeploy agen.

> **Note**
>
> CodeDeploy Agen hanya diperlukan jika Anda menyebarkan ke platform EC2/On-Premises komputasi. Agen tidak diperlukan untuk penerapan yang menggunakan platform komputasi Amazon ECS atau Lambda AWS .

File konfigurasi ditempatkan pada instance ketika agen diinstal. File ini digunakan untuk menentukan cara kerja agen. File konfigurasi ini menentukan jalur direktori dan pengaturan lain AWS CodeDeploy untuk digunakan saat berinteraksi dengan instance. Anda dapat mengubah beberapa opsi konfigurasi dalam file. Untuk informasi tentang bekerja dengan file konfigurasi CodeDeploy agen, lihatCodeDeploy referensi konfigurasi agen.

Untuk informasi selengkapnya tentang bekerja dengan CodeDeploy agen, seperti langkah-langkah untuk menginstal, memperbarui, dan memverifikasi versi, lihatMengelola operasi CodeDeploy agen.

Topik

- Sistem operasi yang didukung oleh CodeDeploy agen
- Protokol komunikasi dan port untuk CodeDeploy agen
- Riwayat versi CodeDeploy agen
- Mengelola CodeDeploy proses
- Revisi aplikasi dan pembersihan file log
- File yang diinstal oleh CodeDeploy agen
- Mengelola operasi CodeDeploy agen

Sistem operasi yang didukung oleh CodeDeploy agen

Sistem operasi Amazon EC2 AMI yang didukung

CodeDeploy Agen telah diuji pada sistem operasi Amazon EC2 AMI berikut:

- 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)
- Linux berbatu 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Server 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 pada instans EC2 Mac dengan silikon Apple (aarch64)

> **Note**
>
> Platform berikut telah diuji pada versi CodeDeploy agen 2.0.x dan yang lebih baru saja: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13, dan Ubuntu Server 24.04, 25.04, 26.04.
> macOS didukung pada CodeDeploy agen versi 2.1.2 dan yang lebih baru saja. Agen mendukung macOS 15, 26, 27 dengan silikon Apple.

CodeDeploy Agen tersedia sebagai open source bagi Anda untuk beradaptasi dengan kebutuhan Anda. Ini dapat digunakan dengan sistem operasi Amazon EC2 AMI lainnya. Untuk informasi lebih lanjut, buka repositori CodeDeploy agen di. GitHub

Sistem operasi lokal yang didukung

CodeDeploy Agen telah diuji pada sistem operasi lokal berikut:

- 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)
- Linux berbatu 9, 10 (x86_64, aarch64)
- CentOS Stream 9, 10 (x86_64, aarch64)
- SLES 15 (x86_64, aarch64)
- Debian 11, 12, 13 (x86_64, aarch64)
- Server 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 dengan silikon Apple (aarch64)

> **Note**
>
> Platform berikut telah diuji pada versi CodeDeploy agen 2.0.x dan yang lebih baru saja: Windows Server 2025, RHEL 10, Oracle Linux 8, 9, 10, Rocky Linux 9, 10, CentOS Stream 9, 10, SLES 15, Debian 11, 12, 13, dan Ubuntu Server 24.04, 25.04, 26.04.
> macOS didukung pada CodeDeploy agen versi 2.1.2 dan yang lebih baru saja. Agen mendukung macOS 15, 26, 27 dengan silikon Apple.

CodeDeploy Agen tersedia sebagai open source bagi Anda untuk beradaptasi dengan kebutuhan Anda. Hal ini dapat digunakan dengan sistem operasi instans lokal lainnya. Untuk informasi lebih lanjut, buka repositori CodeDeploy agen di. GitHub

Protokol komunikasi dan port untuk CodeDeploy agen

CodeDeploy Agen berkomunikasi keluar menggunakan HTTPS melalui port 443.

Ketika CodeDeploy agen berjalan pada instance EC2, ia akan menggunakan titik akhir metadata EC2 untuk mengambil informasi terkait instance. Cari tahu lebih lanjut tentang membatasi dan memberikan akses layanan metadata instans.

Riwayat versi CodeDeploy agen

Instans Anda harus menjalankan versi CodeDeploy agen yang didukung. Versi minimum yang didukung saat ini adalah 2.1.x.

> **Note**
>
> Kami merekomendasikan menggunakan versi terbaru dari CodeDeploy agen. Jika Anda mengalami masalah, perbarui ke versi terbaru sebelum menghubungi AWS Support. Untuk informasi peningkatan, lihatPerbarui CodeDeploy agen.

> **Status peluncuran versi 2.x**
>
> Versi 2.x CodeDeploy agen sedang diluncurkan Wilayah AWS dan mungkin belum tersedia di semua Wilayah. Versi 2.x adalah upgrade opt-in. Untuk meningkatkan ke versi 2.x dari revisi lama, lihat. Instal CodeDeploy agen Untuk mengetahui versi mana yang dapat Anda instal, lihatTentukan versi agen terbaru yang tersedia di Wilayah AWS.

Tabel berikut mencantumkan semua rilis CodeDeploy agen dan fitur serta penyempurnaan yang disertakan dengan setiap versi.

Versi	Tanggal rilis	Detail
2.1.2	Oktober 13, 2026	Ditambahkan: Support untuk macOS 15, 26, 27 dengan silikon Apple. Untuk petunjuk, lihat Instal CodeDeploy agen untuk macOS. Ditambahkan: Rentang unduhan paralel revisi aplikasi dari Amazon S3, dimatikan secara default. Untuk informasi selengkapnya, lihat CodeDeploy referensi konfigurasi agen. Tetap: Output skrip peristiwa Siklus hidup ditulis ke `codedeploy-agent-deployments.log` baris demi baris, diawali dengan ID penyebaran.

Versi	Tanggal rilis	Detail
2.1.1	Oktober 2, 2026	Tetap: Agen sekarang terhubung ke CodeDeploy titik akhir yang benar di AWS European Sovereign Cloud.
2.1.0	September 7, 2026	Ditambahkan: Dukungan asli untuk mode `RESTART` penerapan, yang memulai ulang aplikasi pada instance target menggunakan revisi dari penerapan terakhir grup penerapan yang berhasil. Agen menginstal dari salinan revisi yang sudah ada pada instance. Jika salinan itu hilang atau tidak lengkap, agen mengunduh revisi, sehingga penerapan masih selesai. Untuk mode penerapan itu sendiri, lihat DeploymentMode di Referensi API. AWS CodeDeploy Berubah (keamanan): Agen sekarang menolak penerapan yang jalur AppSpec filenya diselesaikan di luar direktori revisi aplikasi. Jalur yang bersarang di dalam revisi, seperti`configs/appspec.yml` , masih didukung.
2.0.1	Agustus 24, 2026	Tetap: Pada Windows, jalur AppSpec file yang dimulai dengan garis miring terbalik (\) atau dengan lebih dari satu pemisah (misalnya,`//my-folder`) sekarang diselesaikan di dalam revisi, bukan di root drive. Untuk contoh `source` nilai yang didukung, lihatAppSpec Bagian 'file' (EC2/On-Premises hanya penerapan).

Versi	Tanggal rilis	Detail
2.0.0	Juli 15, 2026	Berubah: CodeDeploy Agen sekarang Rust-based dan dikirimkan sebagai biner asli mandiri tunggal. Ketergantungan runtime Ruby telah dihapus. Ruby tidak lagi diperlukan untuk menginstal atau menjalankan agen. Agen kompatibel dengan perilaku dengan versi 1.8.x secara default; file konfigurasi yang ada, AppSpec penanganan, semantik kait siklus hidup, tata letak on-disk, dan hasil penerapan tidak berubah kecuali disebutkan di bawah ini. Ditambahkan: Dukungan sistem operasi yang diperluas , termasuk Windows Server 2025, RHEL 10, dan Ubuntu Server 26.04, serta distribusi Linux tambahan dan varian ARM (aarch64). Untuk daftar lengkap platform yang diuji dengan versi ini, lihatSistem operasi yang didukung oleh CodeDeploy agen. Ditambahkan: `codedeploy-agent update` Perintah untuk pembaruan mandiri agen on-demand. Ditambahkan: Sebuah port perintah lokal. Ini adalah antarmuka manajemen TCP opsional yang hanya mendengarkan pada alamat loopback (127.0.0.1) untuk menanyakan status agen dan menyuntikkan perintah secara lokal. Ini dinonaktifkan secara default dan diaktifkan dengan`enable_command_port` . Berubah: Perintah penyebaran lokal sekarang`codedeploy-agent deploy-local` . `codedeploy-local` Perintah tetap tersedia untuk kompatibilitas mundur. Semua opsi perintah dipertahankan, dan jalur lokal, Amazon S3 (`s3://`), dan sumber GitHub bundel masih didukung. Berubah: Di Linux, agen dikelola oleh unit systemd asli (`codedeploy-agent.service`); skrip init.d SysV tidak lagi diinstal. Gunakan `systemctl` untuk mengelola agen.

Versi	Tanggal rilis	Detail
		Berubah (keamanan): Verifikasi sertifikat TLS selalu diaktifkan, dan dump inti dinonaktifkan secara default. Ditambahkan: Pengaturan pengerasan opsional yang dapat Anda aktifkan untuk memenuhi persyaratan keamanan Anda sendiri. Setiap pengaturan dimatikan secara default, dan Anda mengaktifkan yang Anda inginkan satu per satu. Anda dapat menolak bundel yang berisi tautan simbolis, jalur traversal, atau izin tidak aman; membatasi lingkungan kait peristiwa siklus hidup; dan membatasi izin direktori agen dan file log. Untuk daftar lengkapnya, lihat CodeDeploy referensi konfigurasi agen. Berubah: Script instalasi baris perintah sekarang ditulis dalam Bash (sebelumnya Ruby). Itu membutuhkan salah satu `curl` atau`wget`. Berubah: Skrip penginstalan untuk versi 2.0.x diterbitkan di bawah `latestv2/` awalan di bucket regional. `aws-codedeploy-` *`region-identifier`* `latest/`Awalan terus melayani skrip instalasi versi 1.8.x. Berubah: AWS sekarang menerbitkan versi 2.0.x sebagai paket `AWSCodeDeployAgentV2` Systems Manager Distributor. `AWSCodeDeployAgent` Paket terus melayani versi 1.8.x. Dihapus: Pengaturan `ssl_verify_peer` konfigurasi dan tombol ProcessManager konfigurasi Ruby. Penting: Saat ini tidak ada jalur pembaruan otomatis dari versi 1.8.x ke 2.0.0. Untuk meningkatkan ke 2.0.0, instal dengan paket `AWSCodeDeployAgentV2` Systems Manager Distributor atau jalankan `latestv2/install` skrip regional secara manual. Untuk petunjuk, lihat Instal CodeDeploy agen.

Versi	Tanggal rilis	Detail
1.8.1	Februari 3, 2026	Tetap: S3 endpoint bugfix. Ditambahkan: Sertifikat Sectigo CA ke daftar Windows CA.
1.8.0	Juli 31, 2025	Berubah: Upgrade Ruby yang dibundel ke 3.2 di CodeDeploy agen untuk Windows.
1.7.1	November 14, 2024	Berubah: Dependensi yang diperbarui untuk patch keamanan.
1.7.0	Maret 6, 2024	Ditambahkan: Pengaturan `:disable_imds_v1:` konfigurasi ke file konfigurasi CodeDeploy agen. Gunakan pengaturan ini untuk menonaktifkan fallback ke IMDSv1 ketika kesalahan IMDSv2 terjadi. Default ke `false` (aktifkan fallback). Untuk informasi selengkapnya, lihat referensi konfigurasi CodeDeploy agen. Ditambahkan: Support untuk sistem operasi Red Hat Enterprise Linux 9 (RHEL 9). Ditambahkan: Support untuk Ruby versi 3.1 dan 3.2 pada Ubuntu Server. Tetap: CodeDeploy Agen sekarang menghasilkan kesalahan yang mudah digunakan jika file konfigurasi CodeDeploy agen gagal dimuat. Berubah: Upgrade Ruby ke 2.7.8-1 di agen untuk Windows. CodeDeploy

Versi	Tanggal rilis	Detail
1.6.0	30 Maret 2023	Ditambahkan: Support untuk Ruby 3.1, 3.2. Ditambahkan: Support untuk Amazon Linux 2023. Ditambahkan: Support untuk Windows Server 2022. Berubah: Pengaturan default `verbose` sekarang `false` untuk instance Windows Server. Untuk terus mencetak pesan debug dalam file log di Windows, Anda harus mengatur `verbose` ke`true`. Dihapus: Support untuk Windows Server 2016 dan Windows Server 2012 R2. Dihapus: Support untuk Amazon Linux 2018.03.x.
1.5.0	3 Maret 2023	Ditambahkan: Support untuk Ruby 3. Ditambahkan: Support untuk Ubuntu 22.04. Tetap: Masalah saat memulai ulang CodeDeploy agen segera setelah startup akan menyebabkan agen menggantung. Berubah: CodeDeploy Agen sekarang gagal dalam penerapan host pada startup agen jika layanan agen dimulai ulang secara tidak terduga saat menjalankan skrip hook. Perbaikan ini memungkinkan Anda menghindari menunggu periode batas waktu 70 menit sebelum mencoba kembali penerapan. Pemberitahuan penghentian: CodeDeploy agen 1.5.0 adalah rilis terakhir untuk mendukung Windows Server 2016 dan Windows Server 2012 R2. Dihapus: Support untuk CodeDeploy agen di Ubuntu 14.04 LTS, Windows Server 2008 R2, dan Windows Server 2008 R2 32-bit.

Versi	Tanggal rilis	Detail
1.4.1	6 Desember 2022	Tetap: Kerentanan keamanan terkait dengan logging. Peningkatan: Peningkatan logging saat polling untuk perintah host.

Versi	Tanggal rilis	Detail
1.4.0	31 Agustus 2022	Ditambahkan: Support untuk Red Hat Enterprise Linux 8. Ditambahkan: Support untuk jalur file panjang pada CodeDeploy agen untuk Windows. Untuk mengaktifkan jalur file yang panjang, Anda harus mengatur kunci registri Windows yang sesuai dan kemudian restart agen Anda. Untuk informasi selengkapnya, lihat Jalur file yang panjang menyebabkan kesalahan “Tidak ada file atau direktori seperti itu”. Tetap: Masalah dengan operasi unzip saat disk penuh. CodeDeploy Agen sekarang mendeteksi kode keluar unzip 50 yang menunjukkan disk penuh, menghapus sebagian file yang diekstraksi, dan memunculkan pengecualian untuk memposting kegagalan ke server. CodeDeploy Pesan kesalahan terlihat sebagai pesan kesalahan peristiwa siklus hidup, dan penerapan tingkat host akan berhenti tanpa macet atau kehabisan waktu. Tetap: Masalah yang akan menyebabkan agen gagal. Tetap: Masalah di mana kait akan habis selama kondisi balapan edge-case. Kait tanpa skrip sekarang akan berlanjut dan tidak lagi menyebabkan kegagalan atau batas waktu. Berubah: `update` Skrip dari `bin` direktori CodeDeploy agen telah dihapus karena tidak lagi digunakan. Berubah: CodeDeploy Agen untuk Windows Server sekarang bundel Ruby 2.7. Berubah: Variabel lingkungan baru ditambahkan, untuk digunakan oleh skrip hook tergantung pada sumber bundel penerapan (Amazon GitHub S3 atau). Untuk informasi selengkapnya, lihat Ketersediaan variabel lingkungan untuk kait.

Versi	Tanggal rilis	Detail
		Important Pemberitahuan penghentian: CodeDeploy agent 1.4.0 adalah rilis terakhir yang akan menyertakan installer untuk Windows Server 32-bit. Pemberitahuan penghentian: CodeDeploy agen 1.4.0 adalah rilis terakhir yang akan mendukung Windows Server 2008 R2. Dihapus: Support untuk CodeDeploy agen pada AMI Amazon EC2 berikut: Amazon Linux 2014.09, 2016.03, 2016.09, dan 2017.03.

Versi	Tanggal rilis	Detail
1.3.2	6 Mei 2021	**Important** CodeDeploy alamat agent 1.3.2 CVE-2018-1000201yang mempengaruhi host Windows yang menjalankan agen. CVE mengutip ruby-ffi, yang merupakan ketergantungan agen. CodeDeploy Jika agen Anda diinstal dengan Amazon EC2 Systems Manager (SSM) dan disetel untuk memperbarui secara otomatis, tindakan tidak diperlukan. Jika tidak, tindakan diperlukan untuk memperbarui agen secara manual. Untuk memutakhirkan agen, ikuti petunjuk di Perbarui CodeDeploy agen di Windows Server. Tetap: Masalah saat menginstal CodeDeploy agen di Ubuntu 20.04 dan yang lebih baru. Tetap: Masalah intermiten yang terjadi saat mengekstrak file terkompresi karena jalur relatif tidak ditangani dengan benar. Ditambahkan: Support untuk AWS PrivateLink dan titik akhir VPC untuk instance Windows. Ditambahkan: perbaikan AppSpec file, seperti dijelaskan di bawah ini. • Anda sekarang dapat menentukan nama file kustom untuk AppSpec file saat membuat penyebaran lokal. Untuk informasi selengkapnya, lihat Buat penerapan lokal. • AppSpec File sekarang dapat memiliki ekstensi `.yaml` file. • Anda sekarang dapat menimpa file yang digunakan menggunakan `file_exists_behavior` pengaturan opsional baru dalam file. AppSpec Untuk informasi selengkapnya, lihat AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan).

Versi	Tanggal rilis	Detail
		Upgrade: CodeDeploy sekarang menggunakan AWS SDK for Ruby 3.0.
1.3.1	22 Desember 2020	Diperbaiki: Masalah 1.3.0 yang mencegah instance lokal dimulai.
1.3.0	10 November 2020	**Important** Versi ini sudah usang. Tetap: Menghapus sertifikat kedaluwarsa yang tidak lagi digunakan. Tetap: Menghapus pesan prompt dari skrip uninstall agen yang digunakan oleh AWS Systems Manager, membuatnya lebih mudah untuk menurunkan versi host atau armada ke versi agen sebelumnya.
1.2.1	23 September 2020	Berubah: AWS SDK untuk Ruby Ketergantungan yang ditingkatkan dari v2 ke v3. Ditambahkan: Support untuk IMDSv2. Termasuk fallback senyap ke IMDSv1 jika permintaan http IMDSv2 gagal. Berubah: Diperbarui dependensi Rake dan Rubyzip untuk patch keamanan. Tetap: Pastikan bahwa file PID kosong akan mengembalikan status `No CodeDeploy Agent Running` dan membersihkan file PID pada agen mulai.

Versi	Tanggal rilis	Detail
1.1.2	4 Agustus 2020	Ditambahkan: Support untuk Ubuntu Server 19.10 dan 20.04. Catatan:: Versi 19.10 mencapai tanggal akhir masa pakainya dan tidak lagi didukung oleh Ubuntu atau. CodeDeploy Ditambahkan: Peningkatan efisiensi memori untuk Linux dan Ubuntu untuk merilis memori cadangan lebih tepat waktu. Ditambahkan: Kompatibilitas dengan Windows Server “silent-cleanup” yang menyebabkan agen menjadi tidak responsif dalam beberapa kasus. Ditambahkan: Abaikan direktori yang tidak kosong selama pembersihan untuk menghindari kegagalan pada penerapan. Ditambahkan: Support untuk Zona AWS Lokal di Los Angeles (LA). Ditambahkan: Ekstrak AZ dari metadata instance untuk menyediakan kompatibilitas untuk AWS Local Zones. Ditambahkan: Pengguna sekarang dapat menyediakan arsip mereka di subdirektori dan tidak diharuskan untuk menyimpan nya di direktori root. Ditambahkan: Mendeteksi masalah dengan Rubyzip yang dapat mengakibatkan kebocoran memori. Memperbarui perintah unzip untuk pertama kali mencoba menggunakan utilitas unzip yang diinstal sistem sebelum menggunakan Rubyzip. Ditambahkan: `:enable_auth_policy:` sebagai pengaturan konfigurasi agen. Berubah: Peringatan unzip sekarang diabaikan sehingga penerapan akan berlanjut.

Versi	Tanggal rilis	Detail
1.1.0	30 Juni 2020	Berubah: Versi CodeDeploy agen sekarang mengikuti konvensi versi standar Ruby. Ditambahkan: Parameter baru untuk menginstal dan memperbarui perintah untuk memungkinkan instalasi versi agen tertentu dari baris perintah. Dihapus: Menghapus CodeDeploy agen Auto Updater untuk Linux dan Ubuntu. Untuk mengonfigurasi pembaruan otomatis CodeDeploy agen, lihat Menginstal CodeDeploy agen menggunakan AWS Systems Manager.
1.0.1.1597	15 November 2018	Peningkatan: CodeDeploy mendukung Ubuntu 18.04. Peningkatan: CodeDeploy mendukung Ruby 2.5. Peningkatan: CodeDeploy mendukung titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS, lihat ikhtisar FIPS 140-2. Untuk titik akhir yang dapat digunakan CodeBuild , lihat CodeDeployWilayah dan titik akhir.
1.0.1.1518	Juni 12, 2018	Peningkatan: Memperbaiki masalah yang menyebabkan kesalahan saat CodeDeploy agen ditutup saat menerima permintaan jajak pendapat. Peningkatan: Menambahkan fitur pelacakan penerapan yang mencegah CodeDeploy agen ditutup saat penerapan sedang berlangsung. Peningkatan: Peningkatan kinerja saat menghapus file.

Versi	Tanggal rilis	Detail
1.0.1.1458	6 Maret 2018	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Peningkatan validasi sertifikat untuk mendukung otoritas yang lebih tepercaya. Peningkatan: Memperbaiki masalah yang menyebabkan CLI lokal gagal selama penerapan yang menyertakan BeforeInstall peristiwa siklus hidup. Peningkatan: Memperbaiki masalah yang mungkin menyebabkan penerapan aktif gagal saat CodeDeploy agen diperbarui.
1.0.1.1352	16 November 2017	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Fitur: Memperkenalkan fitur baru untuk menguji dan men-debug EC2/On-Premises penerapan pada mesin lokal atau instance tempat CodeDeploy agen diinstal.
1.0.1.1106	16 Mei 2017	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Fitur: Memperkenalkan dukungan baru untuk menangani konten di lokasi target yang bukan bagian dari revisi aplikasi dari penerapan sukses terbaru. Opsi penerapan untuk konten yang ada sekarang mencakup mempertahankan konten, menimpa konten, atau kegagalan penerapan. Peningkatan: Membuat CodeDeploy agen kompatibel dengan versi 2.9.2 dari AWS SDK untuk Ruby (aws-sdk-core 2.9.2).

Versi	Tanggal rilis	Detail
1.0.1.1095	Maret 29, 2017	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Memperkenalkan dukungan untuk CodeDeploy agen di Wilayah China (Beijing). Peningkatan: Boneka yang diaktifkan untuk berjalan di instance Windows Server saat dipanggil oleh hook peristiwa siklus hidup. Peningkatan: Meningkatkan penanganan `untar` operasi.
1.0.1.1067	Januari 6, 2017	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Merevisi banyak pesan kesalahan untuk menyertakan penyebab yang lebih spesifik untuk kegagalan penerapan. Peningkatan: Memperbaiki masalah yang mencegah CodeDeploy agen mengidentifikasi revisi aplikasi yang benar untuk diterapkan selama beberapa penerapan. Peningkatan: Mengembalikan penggunaan `pushd` dan `popd` sebelum dan sesudah operasi. `untar`
1.0.1.1045	21 November 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Membuat CodeDeploy agen kompatibel dengan versi 2.6.11 dari AWS SDK untuk Ruby (aws-sdk-core 2.6.11).

Versi	Tanggal rilis	Detail
1.0.1.1037	19 Oktober 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. CodeDeploy Agen untuk instans Amazon Linux, RHEL, dan Ubuntu Server telah diperbarui dengan perubahan berikut. Untuk instance Windows Server, versi terbaru tetap 1.0.1.998. Peningkatan: Agen sekarang dapat menentukan versi Ruby mana yang diinstal pada sebuah instance sehingga dapat memanggil `codedeploy-agent` skrip menggunakan versi itu.
1.0.1.1011.1	Agustus 17, 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Menghapus perubahan yang diperkenalkan oleh versi 1.0.1.1011 karena masalah dengan dukungan shell. Versi agen ini secara fungsional setara dengan versi 1.0.1.998 yang dirilis pada 11 Juli 2016.
1.0.1.1011	Agustus 15, 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. CodeDeploy Agen untuk instans Amazon Linux, RHEL, dan Ubuntu Server telah diperbarui dengan perubahan berikut. Untuk instance Windows Server, versi terbaru tetap 1.0.1.998. Fitur: Menambahkan dukungan untuk memanggil CodeDeploy agen menggunakan shell bash pada sistem operasi tempat sistem init systemd digunakan. Peningkatan: Dukungan yang diaktifkan untuk semua versi Ruby 2.x di CodeDeploy agen dan pembaru agen. CodeDeploy CodeDeploy Agen yang diperbarui tidak lagi bergantung pada Ruby 2.0 saja. (Ruby 2.0 masih diperlukan untuk versi deb dan rpm dari penginstal CodeDeploy agen.)

Versi	Tanggal rilis	Detail
1.0.1.998	Juli 11, 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Peningkatan: Dukungan tetap untuk menjalankan CodeDeploy agen dengan profil pengguna selain root. Variabel bernama `USER` diganti dengan `CODEDEPLOY_USER` untuk menghindari konflik dengan variabel lingkungan.
1.0.1.966	16 Juni 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Fitur: Memperkenalkan dukungan untuk menjalankan CodeDeploy agen dengan profil pengguna selain root. Peningkatan: Dukungan tetap untuk menentukan jumlah revisi aplikasi yang Anda ingin CodeDeploy agen arsipkan untuk grup penyebaran. Peningkatan: Membuat CodeDeploy agen kompatibel dengan versi 2.3 dari AWS SDK untuk Ruby (aws-sdk-core 2.3). Peningkatan: Memperbaiki masalah dengan UTF-8 pengkodean selama penerapan. Peningkatan: Peningkatan akurasi saat mengidentifikasi nama proses.
1.0.1.950	24 Maret 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunakan versi ini, penerapan Anda mungkin gagal. Fitur: Menambahkan dukungan proxy instalasi. Peningkatan: Memperbarui skrip instalasi untuk tidak mengunduh CodeDeploy agen jika versi terbaru sudah diinstal.

Versi	Tanggal rilis	Detail
1.0.1.934	11 Februari 2016	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunak an versi ini, penerapan Anda mungkin gagal. Fitur: Memperkenalkan dukungan untuk menentukan jumlah revisi aplikasi yang Anda ingin CodeDeploy agen arsipkan untuk grup penyebaran.
1.0.1.880	Januari 11, 2016	Catatan: Versi ini tidak lagi didukung dan dapat menyebabkan penerapan gagal. Peningkatan: Membuat CodeDeploy agen kompatibel dengan versi 2.2 dari AWS SDK untuk Ruby (aws-sdk-core 2.2). Versi 2.1.2 masih didukung.
1.0.1.854	17 November 2015	Catatan: Versi ini tidak lagi didukung. Jika Anda menggunak an versi ini, penerapan Anda mungkin gagal. Fitur: Memperkenalkan dukungan untuk algoritma SHA-256 hash. Fitur: Memperkenalkan dukungan pelacakan versi dalam `.version` file. Fitur: Membuat ID grup penyebaran tersedia melalui penggunaan variabel lingkungan. Peningkatan: Menambahkan dukungan untuk memantau log CodeDeploy agen menggunakan Amazon CloudWatch Logs.

Untuk informasi terkait, lihat yang berikut ini:

- Tentukan versi CodeDeploy agen
- Instal CodeDeploy agen

Untuk riwayat versi CodeDeploy agen, lihat repositori Rilis aktif. GitHub

Mengelola CodeDeploy proses

CodeDeploy Agen menggunakan systemd untuk mengelola proses agen pada semua distribusi Linux yang didukung.

Untuk versi 2.0.x dan yang lebih baru, agen hanya mengirimkan unit systemd asli ()`codedeploy-agent.service`. Tidak ada `/etc/init.d/codedeploy-agent` naskah. Gunakan `systemctl` untuk mengelola agen, atau memanggil biner secara langsung dengan`start`,, `stoprestart`, atau `status` subperintah.

Untuk versi 1.8.x dan sebelumnya, distribusi rpm dan deb dikirimkan dengan skrip startup di. `/etc/init.d/codedeploy-agent` Pada sistem yang mendukung systemd, menggunakan `service` perintah mungkin secara tidak sengaja menjalankan skrip init.d alih-alih mengizinkan systemd untuk mengelola proses. Gunakan `systemctl` untuk operasi agen untuk menghindari masalah ini.

Untuk mengelola CodeDeploy agen, gunakan `systemctl` utilitas:

```
sudo systemctl start codedeploy-agent
sudo systemctl stop codedeploy-agent
sudo systemctl restart codedeploy-agent
sudo systemctl status codedeploy-agent
```

Di macOS, agen berjalan sebagai daemon launchd`com.amazon.aws.codedeploy-agent`, yang ditentukan `/Library/LaunchDaemons/com.amazon.aws.codedeploy-agent.plist` dalam. launchd memulai agen saat boot dan memulai ulang jika macet. Untuk mengelola agen di macOS, gunakan utilitas`launchctl`:

```
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
```

Perintah ini mulai, berhenti, restart, dan menunjukkan status agen, masing-masing.

Revisi aplikasi dan pembersihan file log

CodeDeploy Agen mengarsipkan revisi dan file log pada instance. CodeDeploy Agen membersihkan artefak ini untuk menghemat ruang disk.

Log penerapan revisi aplikasi: Anda dapat menggunakan opsi:max_revisions: di file konfigurasi agen untuk menentukan jumlah revisi aplikasi yang akan diarsipkan dengan memasukkan bilangan bulat positif apa pun. CodeDeploy juga mengarsipkan file log untuk revisi tersebut. Semua yang lain dihapus, dengan pengecualian file log dari penerapan terakhir yang berhasil. File log itu selalu dipertahankan, bahkan jika jumlah penerapan yang gagal melebihi jumlah revisi yang dipertahankan. Jika tidak ada nilai yang ditentukan, CodeDeploy pertahankan lima revisi terbaru selain revisi yang saat ini diterapkan.

CodeDeploy log: Untuk Amazon Linux, Ubuntu Server, RHEL, dan macOS instance, agen memutar CodeDeploy file log di bawah folder. `/var/log/aws/codedeploy-agent` File log diputar pada 00:00:00 (waktu instance) setiap hari, atau ketika file mencapai 64 MB, mana yang lebih dulu. Jika rotasi kedua terjadi pada hari yang sama, agen menambahkan akhiran numerik (misalnya,). `codedeploy-agent.20260715.1.log` Agen menghapus file log setelah tujuh hari. Pola penamaan untuk file log yang diputar adalah`codedeploy-agent.`*`YYYYMMDD`*`.log`.

File yang diinstal oleh CodeDeploy agen

CodeDeploy Agen menyimpan revisi, riwayat penerapan, dan skrip penerapan di direktori root pada sebuah instance. Nama default dan lokasi direktori ini adalah:

`'/opt/codedeploy-agent/deployment-root'`untuk instans Amazon Linux, Ubuntu Server, RHEL, dan macOS.

`'C:\ProgramData\Amazon\CodeDeploy'`untuk instance Windows Server.

Anda dapat menggunakan pengaturan root_dir dalam file konfigurasi CodeDeploy agen untuk mengonfigurasi nama dan lokasi direktori. Untuk informasi selengkapnya, lihat CodeDeploy referensi konfigurasi agen.

Berikut ini adalah contoh struktur file dan direktori di bawah direktori root. Struktur mengasumsikan ada N jumlah grup penyebaran, dan setiap grup penyebaran berisi N jumlah penyebaran.

```
|--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
```

- Folder ID Grup Deployment mewakili setiap grup penyebaran Anda. Nama direktori grup penyebaran adalah ID-nya (misalnya,`acde1916-9099-7caf-fd21-012345abcdef`). Setiap direktori grup penyebaran berisi satu subdirektori untuk setiap upaya penerapan dalam grup penyebaran tersebut.

  Anda dapat menggunakan perintah batch-get-deployments untuk menemukan ID grup deployment.
- Folder ID Deployment mewakili setiap penerapan dalam grup penyebaran. Setiap nama direktori penyebaran adalah ID-nya. Setiap folder berisi:
  - bundle.tar, file terkompresi dengan isi revisi penerapan. Gunakan utilitas dekompresi zip jika Anda ingin melihat revisi.
  - deployment-archive, direktori yang berisi isi revisi penyebaran.
  - log, direktori yang berisi `scripts.log` file. File ini mencantumkan output dari semua skrip yang ditentukan dalam file penyebaran. AppSpec

  Jika Anda ingin menemukan folder untuk penerapan tetapi tidak mengetahui ID penyebaran atau ID grup penyebaran, Anda dapat menggunakan AWS CodeDeploy konsol atau AWS CLI untuk menemukannya. Untuk informasi selengkapnya, lihat Lihat CodeDeploy detail penerapan.

  Jumlah maksimum penerapan default yang dapat diarsipkan dalam grup penyebaran adalah lima. Ketika nomor itu tercapai, penerapan future diarsipkan dan arsip tertua dihapus. Anda dapat menggunakan pengaturan max_revisions dalam file konfigurasi CodeDeploy agen untuk mengubah default. Untuk informasi selengkapnya, lihat CodeDeploy referensi konfigurasi agen.

  > **Note**
  >
  > Jika Anda ingin memulihkan ruang hard disk yang digunakan oleh penerapan yang diarsipkan, perbarui pengaturan max_revisions ke angka rendah, seperti 1 atau 2. Penerapan berikutnya menghapus penerapan yang diarsipkan sehingga jumlahnya sama dengan nomor yang Anda tentukan.
- deployment-instructions berisi empat file teks untuk setiap grup penyebaran:
  - [ID Grup Deployment] -cleanup, file teks dengan versi undo dari setiap perintah yang dijalankan selama penerapan. Contoh nama file adalah`acde1916-9099-7caf-fd21-012345abcdef-cleanup`.
  - [ID Grup Penerapan] -install.json, file JSON yang dibuat selama penerapan terbaru. Ini berisi perintah yang dijalankan selama penerapan. Contoh nama file adalah`acde1916-9099-7caf-fd21-012345abcdef-install.json`.

  - [ID Grup Deployment] _last_successful_install, file teks yang mencantumkan direktori arsip dari penerapan terakhir yang berhasil. File ini dibuat ketika CodeDeploy agen telah menyalin semua file dalam aplikasi penyebaran ke instance. Ini digunakan oleh CodeDeploy agen selama penyebaran berikutnya untuk menentukan `BeforeInstall` skrip mana `ApplicationStop` dan yang akan dijalankan. Contoh nama file adalah`acde1916-9099-7caf-fd21-012345abcdef_last_successful_install`.
  - [ID Grup Deployment] _most_recent_install, file teks yang mencantumkan nama direktori arsip dari penerapan terbaru. File ini dibuat ketika file dalam penyebaran berhasil diunduh. File [deployment group ID] _last_successful_install dibuat setelah file ini, ketika file yang diunduh disalin ke tujuan akhir. Contoh nama file adalah`acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.
- deployment-logs berisi file log berikut:
  - codedeploy-agent.yyyymmdd.log file dibuat untuk setiap hari ada penyebaran. Setiap file log berisi informasi tentang penerapan hari itu. File log ini mungkin berguna untuk masalah debugging seperti masalah izin. File log awalnya bernama`codedeploy-agent.log`. Keesokan harinya, tanggal penerapannya dimasukkan ke dalam nama file. Misalnya, jika hari ini adalah 3 Januari 2018, Anda dapat melihat informasi tentang semua penerapan hari ini di. `codedeploy-agent.log` Besok, pada 4 Januari 2018, file log diganti namanya`codedeploy-agent.20180103.log`.
  - codedeploy-agent-deployments.log mengkompilasi isi `scripts.log` file untuk setiap penyebaran. `scripts.log`File-file tersebut terletak di `logs` subfolder di bawah setiap `Deployment ID` folder. Entri dalam file ini didahului oleh ID penyebaran. Misalnya, "`[d-ABCDEF123]LifecycleEvent - BeforeInstall`" mungkin ditulis selama penerapan dengan ID dari. `d-ABCDEF123` Ketika `codedeploy-agent-deployments.log` mencapai ukuran maksimumnya, CodeDeploy agen terus menulis padanya sambil menghapus konten lama.

Mengelola operasi CodeDeploy agen

Petunjuk di bagian ini menunjukkan kepada Anda cara menginstal, menghapus, menginstal ulang, atau memperbarui CodeDeploy agen dan cara memverifikasi CodeDeploy agen sedang berjalan.

> **Tip**
>
> Beberapa instruksi di halaman ini berbeda tergantung pada versi agen. Jika Anda tidak yakin versi mana yang diinstal, jalankan `/opt/codedeploy-agent/bin/codedeploy-`

`agent --version` (Linux atau macOS) & `'C:\ProgramData\Amazon\CodeDeploy \bin\codedeploy-agent.exe' --version` atau (Windows). Jika perintah itu tidak dikenali, Anda menjalankan versi 1.8.x atau yang lebih lama. Untuk informasi selengkapnya, lihat Tentukan versi CodeDeploy agen.

Topik

- Verifikasi CodeDeploy agen sedang berjalan
- Tentukan versi CodeDeploy agen
- Instal CodeDeploy agen
- Perbarui CodeDeploy agen
- Copot instalan agen CodeDeploy
- Kirim log CodeDeploy agen ke CloudWatch

Verifikasi CodeDeploy agen sedang berjalan

Bagian ini menjelaskan perintah untuk dijalankan jika Anda mencurigai CodeDeploy agen telah berhenti berjalan pada sebuah instance.

Topik

- Verifikasi CodeDeploy agen untuk Amazon Linux atau RHEL sedang berjalan
- Verifikasi CodeDeploy agen untuk Ubuntu Server sedang berjalan
- Verifikasi CodeDeploy agen untuk Windows Server sedang berjalan
- Verifikasi CodeDeploy agen untuk macOS sedang berjalan

Verifikasi CodeDeploy agen untuk Amazon Linux atau RHEL sedang berjalan

Untuk melihat apakah CodeDeploy agen diinstal dan dijalankan, masuk ke instance, dan jalankan perintah berikut:

```
systemctl status codedeploy-agent
```

Jika perintah mengembalikan kesalahan, CodeDeploy agen tidak diinstal. Instal seperti yang dijelaskan dalamInstal CodeDeploy agen untuk Amazon Linux atau RHEL.

Jika CodeDeploy agen diinstal dan berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.` (kode keluar 0).

Jika Anda melihat pesan seperti `The AWS CodeDeploy agent is not running.` (kode keluar 3), mulai layanan dan jalankan dua perintah berikut, satu per satu:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Untuk versi 2.0.x dan yang lebih baru, `codedeploy-agent status` perintah mengembalikan kode LSB-compatible keluar: `0` (berjalan), `3` (tidak berjalan), `4` (tidak diketahui). Saat menggunakan`systemctl status`, status unit ditampilkan (misalnya,`Active: active (running)`).

Verifikasi CodeDeploy agen untuk Ubuntu Server sedang berjalan

Untuk melihat apakah CodeDeploy agen diinstal dan dijalankan, masuk ke instance, dan jalankan perintah berikut:

```
systemctl status codedeploy-agent
```

Jika perintah mengembalikan kesalahan, CodeDeploy agen tidak diinstal. Instal seperti yang dijelaskan dalamInstal CodeDeploy agen untuk Server Ubuntu.

Jika CodeDeploy agen diinstal dan berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.` (kode keluar 0).

Jika Anda melihat pesan seperti `The AWS CodeDeploy agent is not running.` (kode keluar 3), mulai layanan dan jalankan dua perintah berikut, satu per satu:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Verifikasi CodeDeploy agen untuk Windows Server sedang berjalan

Untuk melihat apakah CodeDeploy agen diinstal dan dijalankan, masuk ke instance, dan jalankan perintah berikut:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Anda akan melihat output yang serupa dengan yang berikut:

```
Status   Name               DisplayName
------   ----               -----------
Running codedeployagent    CodeDeploy Host Agent Service
```

Jika perintah mengembalikan kesalahan, CodeDeploy agen tidak diinstal. Instal seperti yang dijelaskan dalamInstal CodeDeploy agen untuk Windows Server.

Jika `Status` menunjukkan sesuatu selain`Running`, mulai layanan dengan perintah berikut:

```
powershell.exe -Command Start-Service -Name codedeployagent
```

Anda dapat memulai ulang layanan dengan perintah berikut:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Anda dapat menghentikan layanan dengan perintah berikut:

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Verifikasi CodeDeploy agen untuk macOS sedang berjalan

Untuk melihat apakah CodeDeploy agen diinstal dan dijalankan, masuk ke instance, dan jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Jika perintah mengembalikan kesalahan karena file tidak ditemukan, CodeDeploy agen tidak diinstal. Instal seperti yang dijelaskan dalamInstal CodeDeploy agen untuk macOS.

Jika CodeDeploy agen diinstal dan berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.` (kode keluar 0).

Jika Anda melihat pesan seperti `The AWS CodeDeploy agent is not running.` (kode keluar 3), mulai agen dengan perintah berikut:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

Jika perintah itu melaporkan bahwa layanan tidak dapat ditemukan, daemon launchd tidak dimuat. Muat dan mulai dengan perintah berikut:

```
sudo launchctl bootstrap system /Library/LaunchDaemons/com.amazon.aws.codedeploy-
agent.plist
```

Untuk melihat status daemon launchd, jalankan:

```
sudo launchctl print system/com.amazon.aws.codedeploy-agent
```

Tentukan versi CodeDeploy agen

Anda dapat menentukan versi CodeDeploy agen yang berjalan pada instance Anda dengan dua cara.

Pertama, untuk versi 2.0.x dan yang lebih baru, gunakan `codedeploy-agent --version` perintah di semua platform.

Kedua, Anda dapat melihat versi dalam `.version` file pada instance. Di Amazon Linux, RHEL, Ubuntu Server, dan macOS file tersebut berada `/opt/codedeploy-agent/.version` di, dan di Windows Server di. `C:\ProgramData\Amazon\CodeDeploy\.version` Isi file berbeda menurut versi:

- Untuk versi 2.0.x dan yang lebih baru, file berisi nomor versi telanjang, dalam format. `agent_version: 2.0.0` Nilai ini cocok dengan versi yang dilaporkan agen ke CodeDeploy layanan.
- Untuk versi 1.8.x dan sebelumnya, string versi menggunakan format `OFFICIAL_1.0.1.854_rpm` (atau_deb,_msi).

Topik

- Tentukan versi di Amazon Linux atau RHEL
- Tentukan versi di Server Ubuntu
- Tentukan versi di Windows Server
- Tentukan versi di macOS

- Tentukan versi agen terbaru yang tersedia di Wilayah AWS

Tentukan versi di Amazon Linux atau RHEL

Masuk ke instance dan jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Untuk versi yang lebih awal dari 2.0.0, Anda juga dapat menggunakan:

```
sudo yum info codedeploy-agent
```

Tentukan versi di Server Ubuntu

Masuk ke instance dan jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Untuk versi yang lebih awal dari 2.0.0, Anda juga dapat menggunakan:

```
sudo dpkg -s codedeploy-agent
```

Tentukan versi di Windows Server

Masuk ke instance dan jalankan perintah berikut:

```
& 'C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe' --version
```

Tentukan versi di macOS

Masuk ke instance dan jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent --version
```

Anda juga dapat melihat versi paket yang diinstal:

```
pkgutil --pkg-info com.amazon.aws.codedeploy-agent
```

Tentukan versi agen terbaru yang tersedia di Wilayah AWS

Untuk mengetahui versi mana yang terbaru yang tersedia di tempat Wilayah AWS Anda menerapkan, kueri bucket kit CodeDeploy sumber daya untuk Wilayah tersebut. Setiap bucket regional menyimpan `latestv2/LATEST_VERSION` file yang memberi nama paket agen yang saat ini dirilis di Wilayah tersebut. File ini dapat dibaca publik, jadi Anda tidak memerlukan kredensil untuk mengambilnya.

```
curl https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/LATEST_VERSION
```

Ganti *`bucket-name`* dan *`region-identifier`* dengan bucket kit sumber daya dan pengenal Wilayah untuk Wilayah yang Anda gunakan. Untuk daftarnya, lihatNama bucket kit sumber daya menurut Wilayah.

Perintah tersebut mengembalikan output serupa dengan berikut ini:

```
{
    "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"
}
```

Setiap kunci menamai jenis paket, dan untuk RPM dan DEB arsitektur instans juga. Setiap nilai adalah kunci objek Amazon S3 dari paket itu, dan versi agen serta nomor build adalah bagian dari nama filenya, dalam format. *`agent-version-build-number`* Pada output sebelumnya, versi terbaru yang tersedia di Wilayah adalah 2.1.0.

Untuk versi 2.1.2 dan yang lebih baru, file tersebut juga memiliki `pkg_aarch64` kunci, yang menamai paket penginstal macOS.

Untuk menginstal atau memperbarui ke versi nama file ini, lihat Instal CodeDeploy agen danPerbarui CodeDeploy agen. Untuk fitur yang disertakan dalam setiap versi, lihatRiwayat versi CodeDeploy agen.

Instal CodeDeploy agen

Untuk digunakan CodeDeploy pada instans EC2 atau server lokal, CodeDeploy agen harus diinstal terlebih dahulu. Anda dapat menginstal CodeDeploy agen dengan AWS Systems Manager atau

dengan baris perintah. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu AWS Systems Manager.

Untuk versi yang disarankan untuk diinstal, lihatRiwayat versi CodeDeploy agen.

Topik

- Instal CodeDeploy agen menggunakan AWS Systems Manager
- Instal CodeDeploy agen menggunakan baris perintah

Instal CodeDeploy agen menggunakan AWS Systems Manager

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk menginstal CodeDeploy agen ke Amazon EC2 atau instans lokal dengan menggunakan. AWS Systems Manager Anda dapat memilih untuk menginstal versi tertentu atau memilih untuk selalu menginstal versi terbaru dari agen. Untuk informasi lebih lanjut tentang AWS Systems Manager, lihat Apa itu AWS Systems Manager.

Anda juga dapat menginstal CodeDeploy agen dari ember Amazon S3. Untuk informasi tentang penggunaan tautan unduhan Amazon S3, silakan lihat Instal CodeDeploy agen menggunakan baris perintah.

Topik

- Prasyarat
- Instal CodeDeploy agen

Prasyarat

Ikuti langkah-langkah Memulai dengan CodeDeploy untuk mengatur izin IAM dan. AWS CLI

Jika menginstal CodeDeploy agen di server lokal dengan Systems Manager, Anda harus mendaftarkan server lokal Anda dengan Amazon EC2 Systems Manager. Untuk informasi selengkapnya, lihat Menyiapkan Systems Manager di lingkungan hybrid di Panduan AWS Systems Manager Pengguna.

Instal CodeDeploy agen

Sebelum Anda dapat menggunakan Systems Manager untuk menginstal CodeDeploy agen, Anda harus memastikan bahwa instance dikonfigurasi dengan benar untuk Systems Manager.

Menginstal atau memperbarui agen SSM

Pada instans Amazon EC2, CodeDeploy agen mengharuskan instans menjalankan versi 2.3.274.0 atau yang lebih baru. Sebelum Anda menginstal CodeDeploy agen, perbarui atau instal agen SSM pada instance jika Anda belum melakukannya.

Agen SSM sudah diinstal sebelumnya pada beberapa AMI Amazon EC2 yang disediakan oleh. AWS Untuk informasi selengkapnya, lihat Amazon Machine Images (AMI) dengan agen SSM yang sudah diinstal sebelumnya.

> **Note**
>
> Pastikan bahwa sistem operasi instans juga didukung oleh CodeDeploy agen. Untuk informasi selengkapnya, lihat Sistem operasi yang didukung oleh CodeDeploy agen.

Untuk informasi tentang menginstal atau memperbarui agen SSM pada instans yang menjalankan Linux, lihat Menginstal dan mengonfigurasi agen SSM di instance Linux di Panduan Pengguna.AWS Systems Manager

Untuk informasi tentang menginstal atau memperbarui agen SSM pada instans yang menjalankan Windows Server, lihat Menginstal dan mengonfigurasi agen SSM pada instans Windows di Panduan Pengguna.AWS Systems Manager

Untuk informasi tentang menginstal atau memperbarui agen SSM pada instans EC2 Mac, lihat Menginstal dan menghapus instalan agen SSM secara manual pada instans EC2 untuk macOS di Panduan Pengguna.AWS Systems Manager

(Opsional) Verifikasikan Prasyarat Manajer Sistem

Sebelum Anda menggunakan Systems Manager Run Command untuk menginstal CodeDeploy agen, verifikasi bahwa instans Anda memenuhi persyaratan minimum Systems Manager. Untuk informasi selengkapnya, lihat Menyiapkan AWS Systems Manager di Panduan AWS Systems Manager Pengguna.

Instal CodeDeploy agen

Dengan SSM, Anda dapat menginstal CodeDeploy sekali atau mengatur jadwal untuk menginstal versi baru.

Untuk menginstal CodeDeploy agen, pilih paket Systems Manager Distributor untuk versi agen yang Anda inginkan, lalu ikuti langkah-langkah di Install atau update paket dengan AWS Systems Manager distributor. Paket berbeda menurut versi agen:

- Untuk versi 2.0.x dan yang lebih baru, pilih paket. `AWSCodeDeployAgentV2` Ini adalah satu-satunya paket yang mendukung macOS, dimulai dengan versi 2.1.2.
- Untuk versi 1.8.x dan sebelumnya, pilih paket. `AWSCodeDeployAgent`

Instal CodeDeploy agen menggunakan baris perintah

Gunakan topik berikut untuk menginstal dan menjalankan CodeDeploy agen menggunakan baris perintah.

Topik

- Instal CodeDeploy agen untuk Amazon Linux atau RHEL
- Instal CodeDeploy agen untuk Server Ubuntu
- Instal CodeDeploy agen untuk Windows Server
- Instal CodeDeploy agen untuk macOS

Instal CodeDeploy agen untuk Amazon Linux atau RHEL

Masuk ke instance, dan jalankan perintah berikut, satu per satu. Menjalankan perintah `sudo yum update` terlebih dahulu dianggap sebagai praktik terbaik saat menggunakan `yum` untuk menginstal paket, tetapi Anda dapat melewatinya jika Anda tidak ingin memperbarui semua paket Anda.

```
sudo yum update
```

Untuk versi 2.0.x dan yang lebih baru:

```
sudo yum install wget
```

Untuk versi 1.8.x dan sebelumnya:

```
sudo yum install ruby wget
```

(Opsional) Untuk membersihkan AMI dari informasi caching agen sebelumnya, jalankan skrip berikut:

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Ubah ke direktori home Anda:

```
cd /home/ec2-user
```

> **Note**
>
> Pada perintah sebelumnya, `/home/ec2-user` mewakili nama pengguna default untuk instans Amazon Linux atau RHEL Amazon EC2. Jika instans Anda dibuat menggunakan AMI kustom, pemilik AMI mungkin telah menetapkan nama pengguna default yang berbeda.

Unduh penginstal CodeDeploy agen. `latestv2/`Awalan melayani installer versi 2.0.x, dan `latest/` awalan melayani installer versi 1.8.x.

Untuk versi 2.0.x dan yang lebih baru:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Untuk versi 1.8.x dan sebelumnya:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*`bucket-name`*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda, dan *`region-identifier`* merupakan pengenal untuk wilayah Anda.

Misalnya, untuk versi 2.0.x dan yang lebih baru:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Untuk versi 1.8.x dan sebelumnya:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Untuk daftar nama bucket dan pengenal wilayah, lihatNama bucket kit sumber daya menurut Wilayah.

Mengatur izin eksekusi pada `install` file:

```
chmod +x ./install
```

Untuk menginstal versi terbaru dari CodeDeploy agen:

-
```
sudo ./install auto
```

Untuk menginstal versi CodeDeploy agen tertentu:

- Daftar versi yang tersedia di Wilayah Anda:

  ```
  aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
   | grep '\.rpm$'
  ```
- Instal salah satu versi:

  Untuk versi 2.0.x dan yang lebih baru, paket khusus arsitektur:

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.arch.rpm
  ```

  *arch*Dimana `x86_64` atau `aarch64` tergantung pada arsitektur instance Anda.

  Untuk versi 1.8.x dan sebelumnya, paket menggunakan: `noarch`

  ```
  sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
  ```

  Untuk versi terbaru yang dirilis, lihatRiwayat versi CodeDeploy agen.

Untuk memeriksa apakah layanan sedang berjalan, jalankan perintah berikut:

```
systemctl status codedeploy-agent
```

Untuk versi 2.0.x dan yang lebih baru, jika agen sedang berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.`

Untuk versi 1.8.x dan sebelumnya, jika agen sedang berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.`

Jika agen tidak berjalan, mulai layanan dan periksa statusnya:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Instal CodeDeploy agen untuk Server Ubuntu

Untuk menginstal CodeDeploy agen di Server Ubuntu

1. Masuk ke instance.
2. Masukkan perintah berikut, satu demi satu:

   ```
   sudo apt update
   ```

   Untuk versi 2.0.x dan yang lebih baru:

   ```
   sudo apt install wget
   ```

   Untuk versi 1.8.x dan sebelumnya:

   ```
   sudo apt install ruby-full wget
   ```
3. Masukkan perintah berikut:

   ```
   cd /home/ubuntu
   ```

   */home/ubuntu*mewakili nama pengguna default untuk instance Server Ubuntu. Jika instans Anda dibuat menggunakan AMI kustom, pemilik AMI mungkin telah menetapkan nama pengguna default yang berbeda.
4. Masukkan perintah berikut. `latestv2/`Awalan melayani installer versi 2.0.x, dan `latest/` awalan melayani installer versi 1.8.x.

   Untuk versi 2.0.x dan yang lebih baru:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/install
```

Untuk versi 1.8.x dan sebelumnya:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*bucket-name*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda, dan *region-identifier* merupakan pengenal untuk wilayah Anda.

Misalnya, untuk versi 2.0.x dan yang lebih baru:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/install`

Untuk versi 1.8.x dan sebelumnya:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`

Untuk daftar nama bucket dan pengenal wilayah, lihatNama bucket kit sumber daya menurut Wilayah.

5. Masukkan perintah berikut:

```
chmod +x ./install
```

6. Lakukan salah satu tindakan berikut:

   - Untuk menginstal versi terbaru dari CodeDeploy agen:

     ```
     sudo ./install auto
     ```

   - Untuk menginstal versi CodeDeploy agen tertentu:

     - Daftar versi yang tersedia di Wilayah Anda:

       ```
       aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.deb$'
       ```

     - Instal salah satu versi:

Untuk versi 2.0.x dan yang lebih baru, paket khusus arsitektur:

```
sudo ./install auto -v releases/codedeploy-agent_version_arch.deb
```

*arch*Dimana `amd64` atau `arm64` tergantung pada arsitektur instance Anda.

Untuk versi 1.8.x dan sebelumnya, paket menggunakan konvensi penamaan yang berbeda:

```
sudo ./install auto -v releases/codedeploy-agent_version_all.deb
```

Untuk versi terbaru yang dirilis, lihatRiwayat versi CodeDeploy agen.

Untuk memeriksa apakah layanan sedang berjalan

1. Masukkan perintah berikut:

   ```
   systemctl status codedeploy-agent
   ```

   Jika CodeDeploy agen diinstal dan dijalankan:

   Untuk versi 2.0.x dan yang lebih baru, Anda akan melihat: `The AWS CodeDeploy agent is running.`

   Untuk versi 1.8.x dan sebelumnya, Anda akan melihat: `The AWS CodeDeploy agent is running.`
2. Jika agen tidak berjalan, mulai layanan dan periksa statusnya:

   ```
   systemctl start codedeploy-agent
   ```

   ```
   systemctl status codedeploy-agent
   ```

Instal CodeDeploy agen untuk Windows Server

Pada instance Windows Server, Anda dapat menggunakan salah satu metode ini untuk mengunduh dan menginstal CodeDeploy agen:

- Gunakan AWS Systems Manager (disarankan)
- Jalankan serangkaian PowerShell perintah Windows.
- Pilih tautan unduhan langsung.
- Jalankan perintah salin Amazon S3.

> **Note**
>
> Folder tempat CodeDeploy agen diinstal adalah`C:\ProgramData\Amazon\CodeDeploy`. Jangan letakkan persimpangan direktori atau symlink di jalur ini. Untuk versi 2.0.x dan yang lebih baru, agen memberlakukan persyaratan ini dan menolak untuk memulai jika biner layanan tidak berada di bawah. `C:\ProgramData\Amazon\CodeDeploy\`

Topik

- Gunakan Systems Manager
- Gunakan Windows PowerShell
- Gunakan tautan langsung
- Gunakan perintah salin Amazon S3

Gunakan Systems Manager

Ikuti instruksi Instal CodeDeploy agen menggunakan AWS Systems Manager untuk menginstal CodeDeploy agen.

Gunakan Windows PowerShell

Masuk ke instance, dan jalankan perintah berikut di Windows PowerShell:

1. Mengharuskan semua skrip dan file konfigurasi yang diunduh dari Internet ditandatangani oleh penerbit tepercaya. Jika Anda diminta untuk mengubah kebijakan eksekusi, ketik "**Y**."

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

2. Memuat AWS Tools for Windows PowerShell.

   ```
   Import-Module AWSPowerShell
   ```

3. Buat direktori ke tempat file instalasi CodeDeploy agen diunduh.

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

4. Konfigurasikan AWS kredensil menggunakan perintah `Set-AWSCredential` and`Initialize-AWSDefaultConfiguration`. Untuk informasi selengkapnya, lihat Menggunakan AWS kredensil di AWS alat untuk PowerShell Panduan Pengguna.
5. Unduh file instalasi CodeDeploy agen.

> **Note**
>
> Untuk versi terbaru yang dirilis, lihatRiwayat versi CodeDeploy agen.

Untuk menginstal versi terbaru dari CodeDeploy agen. `latestv2/`Awalan melayani installer versi 2.x (2.0.x dan yang lebih baru), dan `latest/` awalan melayani installer versi 1.8.x.

- Untuk versi 2.0.x dan yang lebih baru:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latestv2/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Untuk versi 1.8.x dan sebelumnya:

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/
codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Untuk menginstal versi CodeDeploy agen tertentu:

-
```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/
codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
```

*bucket-name*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda. Misalnya, untuk Wilayah Timur AS (Ohio), ganti *bucket-name* denganaws-codedeploy-us-east-2. Untuk daftar nama bucket, lihatNama bucket kit sumber daya menurut Wilayah.

6. Jalankan file instalasi CodeDeploy agen.

```
c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
```

Untuk memeriksa apakah layanan sedang berjalan, jalankan perintah berikut:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Jika CodeDeploy agen baru saja diinstal dan belum dimulai, maka setelah menjalankan Get-Service perintah, di bawah Status, Anda akan melihat**`Start...`**:

```
Status     Name               DisplayName
------     ----               -----------
Start...   codedeployagent    CodeDeploy Host Agent Service
```

Jika CodeDeploy agen sudah berjalan, setelah menjalankan Get-Service perintah, di bawah Status, Anda akan melihat**`Running`**:

```
Status     Name               DisplayName
------     ----               -----------
Running    codedeployagent    CodeDeploy Host Agent Service
```

Gunakan tautan langsung

Jika pengaturan keamanan browser pada instance Windows Server memberikan izin (misalnya, keh`ttps://s3.*.amazonaws.com`), Anda dapat menggunakan tautan langsung untuk Wilayah Anda untuk mengunduh CodeDeploy agen dan kemudian menjalankan penginstal secara manual.

`latestv2/`Awalan melayani installer versi 2.x (2.0.x dan yang lebih baru), dan `latest/` awalan melayani installer versi 1.8.x.

Untuk versi 2.0.x dan yang lebih baru, tautannya adalah:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latestv2/codedeploy-agent.msi
```

Untuk versi 1.8.x dan sebelumnya:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... di *region* mana AWS Wilayah tempat Anda menerapkan aplikasi Anda.

Misalnya, untuk versi 2.0.x dan yang lebih baru:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latestv2/codedeploy-
agent.msi
```

> **Important**
>
> Dapatkan `.msi` file dari Wilayah yang sama dengan CodeDeploy aplikasi Anda. Memilih Wilayah yang berbeda dapat menyebabkan `inconsistent region` kegagalan dalam `codedeploy-agent-log` file saat Anda menjalankan `.msi` file.

Gunakan perintah salin Amazon S3

Jika diinstal pada instance, Anda dapat menggunakan perintah Amazon S3 cp untuk mengunduh CodeDeploy agen dan kemudian menjalankan penginstal secara manual. AWS CLI Untuk selengkapnya, lihat Menginstal AWS Command Line Interface di Microsoft Windows.

`latestv2/`Awalan melayani installer versi 2.x (2.0.x dan yang lebih baru), dan `latest/` awalan melayani installer versi 1.8.x.

Untuk versi 2.0.x dan yang lebih baru, perintah Amazon S3 adalah:

```
aws s3 cp s3://aws-codedeploy-region/latestv2/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

Untuk versi 1.8.x dan sebelumnya:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
 --region region
```

... di *region* mana AWS Wilayah tempat Anda menerapkan aplikasi Anda.

Misalnya, untuk versi 2.0.x dan yang lebih baru:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latestv2/codedeploy-agent.msi codedeploy-
agent.msi --region af-south-1
```

Instal CodeDeploy agen untuk macOS

Masuk ke instance, dan jalankan perintah berikut, satu per satu.

Ubah ke direktori home Anda:

```
cd ~
```

Unduh paket penginstal CodeDeploy agen:

```
curl -O https://bucket-name.s3.region-identifier.amazonaws.com/latestv2/codedeploy-
agent.aarch64.pkg
```

*bucket-name*adalah nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit untuk wilayah Anda, dan *region-identifier* merupakan pengenal untuk wilayah Anda.

Contoh:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latestv2/codedeploy-agent.aarch64.pkg`

Untuk daftar nama bucket dan pengenal wilayah, lihatNama bucket kit sumber daya menurut Wilayah.

Untuk menginstal versi terbaru dari CodeDeploy agen:

-
```
sudo installer -pkg codedeploy-agent.aarch64.pkg -target /
```

Untuk menginstal versi CodeDeploy agen tertentu:

- Daftar versi yang tersedia di Wilayah Anda:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier
 | grep '\.pkg$'
```

- Unduh dan instal salah satu versi:

```
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 /
```

Untuk versi terbaru yang dirilis, lihatRiwayat versi CodeDeploy agen.

Untuk memeriksa apakah layanan sedang berjalan, jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Jika agen berjalan, Anda akan melihat pesan seperti `The AWS CodeDeploy agent is running.`

Jika agen tidak berjalan, mulai layanan dan periksa statusnya:

```
sudo launchctl kickstart system/com.amazon.aws.codedeploy-agent
```

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent status
```

Perbarui CodeDeploy agen

Anda dapat mengonfigurasi pembaruan CodeDeploy agen otomatis dan terjadwal pada semua sistem operasi yang didukung menggunakan AWS Systems Manager. Anda juga dapat memaksa pembaruan pada semua sistem operasi yang didukung dengan menjalankan perintah pada sebuah instance.

Topik

- Perbarui CodeDeploy agen di Amazon Linux atau RHEL
- Perbarui CodeDeploy agen di Server Ubuntu
- Perbarui CodeDeploy agen di Windows Server
- Perbarui CodeDeploy agen di macOS

Perbarui CodeDeploy agen di Amazon Linux atau RHEL

Untuk mengonfigurasi pembaruan otomatis dan terjadwal dari CodeDeploy agen yang menggunakan AWS Systems Manager, ikuti langkah-langkah di Instal CodeDeploy agen dengan AWS Systems Manager.

Untuk versi 2.0.x dan yang lebih baru, untuk memaksa pembaruan, jalankan:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Atau, unduh ulang dan jalankan `latestv2/install` skrip regional. Untuk instruksi instalasi, lihat Instal CodeDeploy agen untuk Amazon Linux atau RHEL.

Untuk versi 1.8.x dan sebelumnya, untuk memaksa pembaruan, jalankan:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Saat ini tidak ada jalur pembaruan otomatis dari versi 1.8.x ke 2.0.0. Untuk meningkatkan ke 2.0.0, instal dengan paket `AWSCodeDeployAgentV2` Systems Manager Distributor atau jalankan `latestv2/install` skrip regional secara manual. Untuk petunjuk, lihat Instal CodeDeploy agen.

Perbarui CodeDeploy agen di Server Ubuntu

Untuk mengonfigurasi pembaruan otomatis dan terjadwal dari CodeDeploy agen yang menggunakan AWS Systems Manager, ikuti langkah-langkah di Instal CodeDeploy agen dengan AWS Systems Manager.

Untuk versi 2.0.x dan yang lebih baru, untuk memaksa pembaruan, jalankan:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Atau, unduh ulang dan jalankan `latestv2/install` skrip regional. Untuk instruksi instalasi, lihat Instal CodeDeploy agen untuk Server Ubuntu.

Untuk versi 1.8.x dan sebelumnya, untuk memaksa pembaruan, jalankan:

```
sudo /opt/codedeploy-agent/bin/install auto
```

> **Important**
>
> Saat ini tidak ada jalur pembaruan otomatis dari versi 1.8.x ke 2.0.0. Untuk meningkatkan ke 2.0.0, instal dengan paket `AWSCodeDeployAgentV2` Systems Manager Distributor atau jalankan `latestv2/install` skrip regional secara manual. Untuk petunjuk, lihat Instal CodeDeploy agen.

Perbarui CodeDeploy agen di Windows Server

Anda dapat mengaktifkan pembaruan otomatis CodeDeploy agen dengan AWS Systems Manager. Dengan Systems Manager, Anda dapat mengonfigurasi jadwal pembaruan untuk Amazon EC2 atau instans lokal dengan membuat asosiasi dengan Manajer Negara Systems Manager. Anda juga dapat memperbarui CodeDeploy agen secara manual dengan menghapus versi saat ini dan menginstal yang lebih baru.

Topik

- Siapkan pembaruan CodeDeploy agen otomatis dengan AWS Systems Manager
- Perbarui CodeDeploy agen secara manual

Siapkan pembaruan CodeDeploy agen otomatis dengan AWS Systems Manager

Untuk mengkonfigurasi Systems Manager dan mengaktifkan pembaruan otomatis CodeDeploy agen, ikuti petunjuk di Instal CodeDeploy agen menggunakan AWS Systems Manager.

Perbarui CodeDeploy agen secara manual

Untuk memperbarui CodeDeploy agen secara manual, Anda dapat menginstal versi terbaru dari CLI atau menggunakan Systems Manager. Ikuti instruksi di Instal CodeDeploy agen. Disarankan agar Anda menghapus versi CodeDeploy agen yang lebih lama dengan mengikuti petunjuk di Copot pemasangan agen. CodeDeploy

Perbarui CodeDeploy agen di macOS

Untuk mengonfigurasi pembaruan otomatis dan terjadwal dari CodeDeploy agen yang menggunakan AWS Systems Manager, ikuti langkah-langkah Instal CodeDeploy agen menggunakan AWS Systems Manager dan pilih `AWSCodeDeployAgentV2` paket.

Untuk memaksa pembaruan, jalankan:

```
sudo /opt/codedeploy-agent/bin/codedeploy-agent update
```

Atau, unduh `.pkg` file terbaru dan instal melalui agen yang ada. Untuk instruksi instalasi, lihat Instal CodeDeploy agen untuk macOS.

Copot instalan agen CodeDeploy

Anda dapat menghapus CodeDeploy agen dari contoh ketika tidak lagi diperlukan atau ketika Anda ingin melakukan instalasi baru.

Copot pemasangan CodeDeploy agen dari Amazon Linux atau RHEL

Untuk menghapus instalan CodeDeploy agen, masuk ke instance dan jalankan perintah berikut:

```
sudo yum erase codedeploy-agent
```

Copot instalan CodeDeploy agen dari Server Ubuntu

Untuk menghapus instalan CodeDeploy agen, masuk ke instance dan jalankan perintah berikut:

```
sudo dpkg --purge codedeploy-agent
```

Copot pemasangan CodeDeploy agen dari Windows Server

Untuk menghapus instalan CodeDeploy agen, masuk ke instance dan jalankan tiga perintah berikut, satu per satu:

```
wmic
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
exit
```

> **Note**
>
> Pada Windows Server 2025 dan yang lebih baru, `wmic` tidak digunakan lagi. Gunakan PowerShell perintah berikut sebagai gantinya:
>
> ```
> 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 }
> ```

Anda juga dapat masuk ke instance, dan di Control Panel, buka Program dan Fitur, pilih Agen CodeDeploy Host, lalu pilih Uninstall.

Copot pemasangan CodeDeploy agen dari macOS

Untuk menghapus instalan CodeDeploy agen, masuk ke instance dan jalankan perintah berikut:

```
sudo /opt/codedeploy-agent/bin/uninstall.sh
```

Kirim log CodeDeploy agen ke CloudWatch

Anda dapat mengirim metrik CodeDeploy agen dan data log untuk CloudWatch menggunakan CloudWatch agen terpadu, atau lebih sederhananya, CloudWatch agen.

Gunakan instruksi berikut untuk menginstal CloudWatch agen dan mengkonfigurasinya untuk digunakan dengan CodeDeploy agen.

Prasyarat

Sebelum memulai, selesaikan tugas berikut:

- Instal CodeDeploy agen dan pastikan itu berjalan. Untuk informasi selengkapnya, lihat Instal CodeDeploy agen dan Verifikasi CodeDeploy agen sedang berjalan.
- Instal CloudWatch agen. Untuk informasi selengkapnya, lihat Menginstal CloudWatch agen.
- Tambahkan izin berikut ke profil instans CodeDeploy IAM:
  - CloudWatchLogsFullAccess

- CloudWatchAgentServerPolicy

Untuk informasi lebih lanjut tentang profil CodeDeploy instance, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 AndaMemulai dengan CodeDeploy.

Konfigurasikan CloudWatch agen untuk mengumpulkan CodeDeploy log

Anda dapat mengonfigurasi CloudWatch agen dengan melangkah melalui wizard atau dengan membuat atau mengedit file konfigurasi secara manual.

Untuk mengkonfigurasi CloudWatch agen menggunakan wizard (Linux)

1. Jalankan wizard, seperti yang dijelaskan dalam Jalankan wizard konfigurasi CloudWatch agen.
2. Di wizard, ketika ditanya `Do you want to monitor any log files?` masuk**1**.
3. Tentukan file log CodeDeploy agen, sebagai berikut:
    1. Untuk `Log file path` masukkan path untuk file CodeDeploy log, misalnya:**`/var/log/aws/codedeploy-agent/codedeploy-agent.log`**.
    2. Untuk `Log group name` memasukkan nama grup log, misalnya:**`codedeploy-agent-log`**.
    3. Untuk `Log stream name` memasukkan nama aliran log, misalnya:**`{instance_id}-codedeploy-agent-log`**.
4. Ketika ditanya`Do you want to specify any additional log files?`, masuk**1**.
5. Tentukan log penyebaran CodeDeploy agen, sebagai berikut:
    1. Untuk `Log file path` masukkan path untuk file log CodeDeploy penyebaran, misalnya:**`/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`**.
    2. Untuk `Log group name` memasukkan nama grup log, misalnya:**`codedeploy-agent-deployment-log`**.
    3. Untuk `Log stream name` memasukkan nama aliran log, misalnya:**`{instance_id}-codedeploy-agent-deployment-log`**.
6. Ketika ditanya`Do you want to specify any additional log files?`, masuk**1**.
7. Tentukan log CodeDeploy agen updater, sebagai berikut:
    1. Untuk`Log file path`, masukkan jalur untuk file log CodeDeploy updater. Path tergantung pada versi agen Anda:

- Untuk versi 2.0.x dan yang lebih baru, gunakan. **/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log**
- Untuk versi 1.8.x dan sebelumnya, gunakan. **/tmp/codedeploy-agent.update.log**

2. Untuk `Log group name` memasukkan nama grup log, misalnya:**codedeploy-agent-updater-log**.
3. Untuk `Log stream name` memasukkan nama aliran log, misalnya:**{instance_id}-codedeploy-agent-updater-log**.

Untuk mengkonfigurasi CloudWatch agen menggunakan wizard (Windows)

1. Jalankan wizard, seperti yang dijelaskan dalam Jalankan wizard konfigurasi CloudWatch agen.
2. Di wizard, ketika ditanya `Do you want to monitor any customized log files?` masuk**1**.
3. Tentukan file CodeDeploy log, sebagai berikut:

   1. Untuk`Log file path`, masukkan jalur untuk file log CodeDeploy agen. Path tergantung pada versi agen Anda:
      - Untuk versi 2.0.x dan yang lebih baru, gunakan. **C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log**
      - Untuk versi 1.8.x dan sebelumnya, gunakan. **C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt**
   2. Untuk `Log group name` memasukkan nama grup log, misalnya:**codedeploy-agent-log**.
   3. Untuk `Log stream name` memasukkan nama aliran log, misalnya:**{instance_id}-codedeploy-agent-log**.
4. Ketika ditanya`Do you want to specify any additional log files?`, masuk**1**.
5. Tentukan log penyebaran CodeDeploy agen, sebagai berikut:

   1. Untuk`Log file path`, masukkan jalur untuk file log CodeDeploy penerapan:**C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log**.
   2. Untuk `Log group name` memasukkan nama grup log, misalnya:**codedeploy-agent-deployment-log**.
   3. Untuk `Log stream name` memasukkan nama aliran log, misalnya:**{instance_id}-codedeploy-agent-deployment-log**.

Untuk mengkonfigurasi CloudWatch agen dengan membuat atau mengedit file konfigurasi secara manual (Linux)

1. Buat atau edit file konfigurasi CloudWatch agen seperti yang dijelaskan dalam Buat atau edit file konfigurasi CloudWatch agen secara manual.
2. Pastikan file tersebut dipanggil `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` dan berisi kode berikut:

   Jalur log updater berbeda menurut versi. Untuk versi 2.0.x dan yang lebih baru, gunakan. `/var/log/aws/codedeploy-agent/codedeploy-agent-updater.log` Untuk versi 1.8.x dan sebelumnya, gunakan. `/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"
                }
            ]
        }
    }
}
...
```

Untuk mengkonfigurasi CloudWatch agen dengan membuat atau mengedit file konfigurasi secara manual (Windows)

1. Buat atau edit file konfigurasi CloudWatch agen seperti yang dijelaskan dalam Buat atau edit file konfigurasi CloudWatch agen secara manual.
2. Pastikan file tersebut dipanggil `C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` dan berisi kode berikut:

   Untuk versi 2.0.x dan yang lebih baru, nama file log agen adalah. `codedeploy-agent.log` Untuk versi 1.8.x dan sebelumnya, nama file log agen adalah. `codedeploy-agent-log.txt` Jalur log penerapan sama untuk kedua versi.

```
...
"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"
                    }
                ]
            },
            ...
        }
    },
...
```

Mulai ulang CloudWatch agen

Setelah melakukan perubahan, restart CloudWatch agen seperti yang dijelaskan dalam Mulai CloudWatch agen.

Bekerja dengan instance untuk CodeDeploy

CodeDeploy mendukung penerapan ke instance yang menjalankan Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL), dan Windows Server.

Anda dapat menggunakannya CodeDeploy untuk menerapkan instans Amazon EC2 dan instans lokal. Instans lokal adalah perangkat fisik apa pun yang bukan instans Amazon EC2 yang dapat menjalankan CodeDeploy agen dan terhubung ke titik akhir layanan AWS publik. Anda dapat menggunakan CodeDeploy untuk secara bersamaan menyebarkan aplikasi ke instans Amazon EC2 di cloud dan ke PC desktop di kantor atau server Anda di pusat data Anda sendiri.

Membandingkan instans Amazon EC2 dengan instans lokal

Tabel berikut membandingkan instans Amazon EC2 dan instans lokal:

Subjek	Instans Amazon EC2	On-premises contoh
Mengharuskan Anda untuk menginstal dan menjalankan versi CodeDeploy agen yang kompatibel dengan sistem operasi yang berjalan pada instance.	Ya	Ya
Membutuhkan instance untuk dapat terhubung ke CodeDeploy.	Ya	Ya
Memerlukan profil instans IAM untuk dilampirkan ke instance. Profil instans IAM harus memiliki izin untuk berpartis ipasi dalam CodeDeploy penerapan. Untuk informasi , lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.	Ya	Tidak

Subjek	Instans Amazon EC2	On-premises contoh
Mengharuskan Anda melakukan salah satu hal berikut untuk mengautentikasi dan mendaftarkan instance: • Buat peran IAM yang dapat diasumsikan oleh pengguna IAM pada setiap instance untuk mengambil kredenal sementara yang diperbaru i secara berkala yang dihasilkan melalui. AWS Security Token Service • Buat pengguna IAM untuk setiap instance dan simpan kredenal akun pengguna IAM dalam teks biasa pada instance.	Tidak	Ya
Mengharuskan Anda untuk mendaftarkan setiap instance CodeDeploy sebelum Anda dapat menerapkannya.	Tidak	Ya
Mengharuskan Anda untuk menandai setiap instance sebelum CodeDeploy dapat menyebarkan ke sana.	Ya	Ya
Dapat berpartisipasi dalam skenario Amazon EC2 Auto Scaling dan Elastic Load Balancing sebagai bagian dari penerapan. CodeDeploy	Ya	Tidak

Subjek	Instans Amazon EC2	On-premises contoh
Dapat digunakan dari bucket GitHub dan repositori Amazon S3.	Ya	Ya
Dapat mendukung pemicu yang meminta pengiriman pemberitahuan SMS atau email ketika peristiwa tertentu terjadi dalam penerapan atau instance.	Ya	Ya
Dapat ditagih untuk penerapan terkait.	Tidak	Ya

Tugas instans untuk CodeDeploy

Untuk meluncurkan atau mengonfigurasi instance untuk digunakan dalam penerapan, pilih dari petunjuk berikut:

Saya ingin meluncurkan instans Amazon Linux atau Windows Server Amazon EC2 baru.	Untuk meluncurkan instans Amazon EC2 dengan upaya paling sedikit, lihat. Buat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat) Untuk meluncurkan instans Amazon EC2 sebagian besar sendiri, lihat. Buat instans Amazon EC2 untuk (CodeDeploy AWS CLI atau konsol Amazon EC2)
Saya ingin meluncurkan Server Ubuntu baru atau instance Amazon EC2 RHEL.	Lihat Buat instans Amazon EC2 untuk (CodeDeploy AWS CLI atau konsol Amazon EC2).

Saya ingin mengonfigurasi instance Amazon Linux, Windows Server, Ubuntu Server, atau RHEL Amazon EC2.	Lihat Konfigurasikan instans Amazon EC2 agar berfungsi CodeDeploy.
Saya ingin mengonfigurasi instance lokal Windows Server, Ubuntu Server, atau RHEL (perangkat fisik yang bukan instance Amazon EC2).	Lihat Working with On-Premises Instances.
Saya CodeDeploy ingin menyediakan armada instance pengganti selama blue/green penerapan.	Lihat Bekerja dengan penerapan di CodeDeploy.

Untuk menyiapkan instans Amazon EC2 di grup Penskalaan Otomatis Amazon EC2, Anda harus mengikuti beberapa langkah tambahan. Untuk informasi selengkapnya, lihat Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling.

Topik

- Tagging Instances for Deployments
- Working with Amazon EC2 Instances
- Working with On-Premises Instances
- View Instance Details
- Instance Health

Menandai instance untuk grup penerapan di CodeDeploy

Untuk membantu mengelola instans Amazon EC2 dan instans lokal, Anda dapat menggunakan tag untuk menetapkan metadata Anda sendiri ke setiap sumber daya. Tag memungkinkan Anda untuk mengkategorikan instans Anda dengan cara yang berbeda (misalnya, berdasarkan tujuan, pemilik, atau lingkungan). Ini berguna ketika Anda memiliki banyak contoh. Anda dapat dengan cepat mengidentifikasi instance atau grup instance berdasarkan tag yang telah Anda tetapkan padanya. Setiap tag terdiri dari satu kunci dan satu nilai opsional, yang keduanya Anda tentukan sendiri. Untuk informasi selengkapnya, lihat Menandai sumber daya Amazon EC2 Anda.

Untuk menentukan instance mana yang disertakan dalam grup CodeDeploy penyebaran, Anda menentukan tag dalam satu atau beberapa grup tag. Instans yang memenuhi kriteria tag Anda adalah instans tempat revisi aplikasi terbaru diinstal saat penerapan ke grup penerapan tersebut dibuat.

> **Note**
>
> Anda juga dapat menyertakan grup Amazon EC2 Auto Scaling dalam grup penerapan, tetapi grup tersebut diidentifikasi berdasarkan namanya, bukan oleh tag yang diterapkan pada instance. Untuk informasi, lihat Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling.

Kriteria untuk instance dalam grup penyebaran dapat sesederhana satu tag dalam satu grup tag. Ini bisa serumit 10 tag masing-masing dalam maksimal tiga grup tag.

Jika Anda menggunakan satu grup tag, instance apa pun yang diidentifikasi oleh setidaknya satu tag dalam grup akan disertakan dalam grup penyebaran. Jika Anda menggunakan beberapa grup tag, hanya instance yang diidentifikasi oleh setidaknya satu tag di setiap grup tag yang disertakan.

Contoh berikut menggambarkan bagaimana tag dan grup tag dapat digunakan untuk memilih instance untuk grup penyebaran.

Topik

- Contoh 1: Grup tag tunggal, tag tunggal
- Contoh 2: Grup tag tunggal, beberapa tag
- Contoh 3: Beberapa grup tag, tag tunggal
- Contoh 4: Beberapa grup tag, beberapa tag

Contoh 1: Grup tag tunggal, tag tunggal

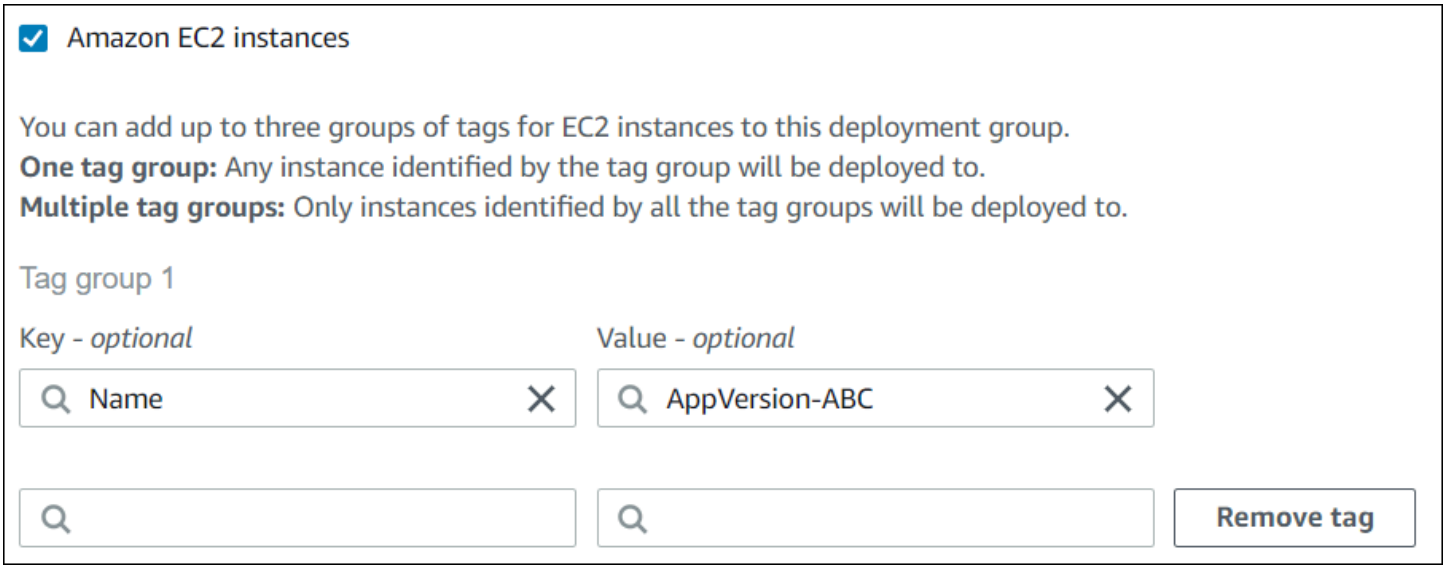
Anda dapat menentukan satu tag dalam satu grup tag:

Tag grup 1

Key	Nilai
Nama	AppVersion-ABC

Setiap instance yang diberi tag `Name=AppVersion-ABC` adalah bagian dari grup penyebaran, meskipun tag lain diterapkan.

CodeDeploy tampilan pengaturan konsol:



Struktur JSON:

```
"ec2TagSet": {
         "ec2TagSetList": [
            [
               {
                  "Type": "KEY_AND_VALUE",
                  "Key": "Name",
                  "Value": "AppVersion-ABC"
               }
            ]
         ]
      },
```

Contoh 2: Grup tag tunggal, beberapa tag

Anda juga dapat menentukan beberapa tag dalam satu grup tag:

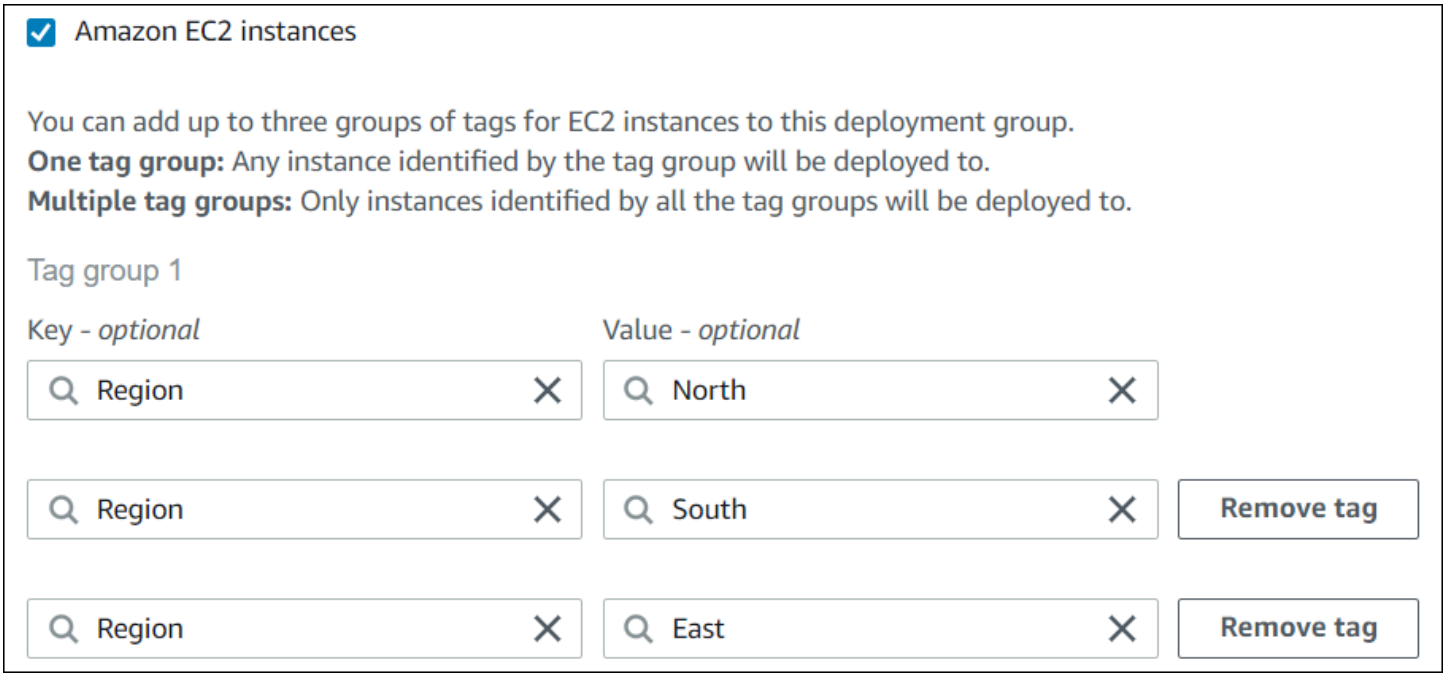
Tag grup 1

Key	Nilai
Region	Utara

Key	Nilai
Region	Selatan
Region	Timur

Instance yang ditandai dengan salah satu dari ketiga tag ini adalah bagian dari grup penyebaran, bahkan jika tag lain diterapkan. Jika, misalnya, Anda memiliki instance lain yang ditandai`Region=West`, instance tersebut tidak akan disertakan dalam grup penerapan.

CodeDeploy tampilan pengaturan konsol:



Struktur 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"
            }
        ]
    ]
},
```

Contoh 3: Beberapa grup tag, tag tunggal

Anda juga dapat menggunakan beberapa set grup tag dengan satu pasangan kunci-nilai di masing-masing untuk menentukan kriteria instance dalam grup penerapan. Bila Anda menggunakan beberapa grup tag dalam grup penyebaran, hanya instance yang diidentifikasi oleh semua grup tag yang disertakan dalam grup penyebaran.

Tag grup 1

Key	Nilai
Nama	AppVersion-ABC

Tag grup 2

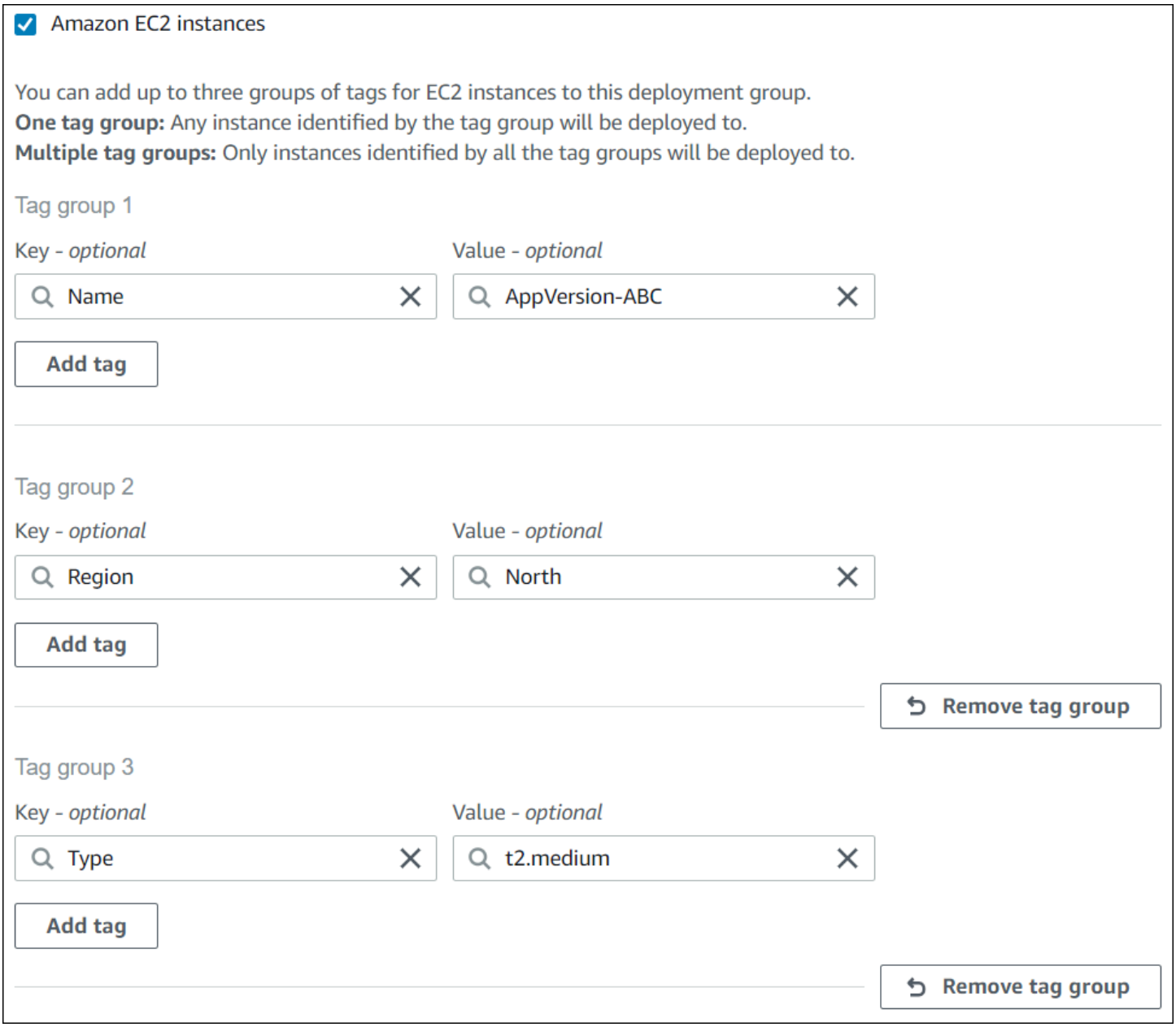
Key	Nilai
Region	Utara

Tag grup 3

Key	Nilai
Tipe	t2.medium

Anda mungkin memiliki instance di banyak Wilayah dan berbagai jenis instance yang ditandai. `Name=AppVersion-ABC` Dalam contoh ini, hanya instance yang juga ditandai dengan `Region=North` dan `Type=t2.medium` merupakan bagian dari grup penerapan.

CodeDeploy tampilan pengaturan konsol:



Struktur 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"
            }
        ],
    ]
},
```

Contoh 4: Beberapa grup tag, beberapa tag

Bila Anda menggunakan beberapa grup tag dengan beberapa tag dalam satu atau beberapa grup, instance harus cocok dengan setidaknya satu tag di setiap grup.

Tag grup 1

Key	Nilai
Lingkungan	Beta
Lingkungan	Pementasan

Tag grup 2

Key	Nilai
Region	Utara
Region	Selatan

Key	Nilai
Region	Timur

Tag grup 3

Key	Nilai
Tipe	t2.medium
Tipe	t2.large

Dalam contoh ini, untuk dimasukkan dalam grup penyebaran, sebuah instance harus ditandai dengan (1) `Environment=Beta` atau`Environment=Staging`, dengan (2),, atau `Region=North` `Region=South``Region=East`, dan dengan (3) `Type=t2.medium` atau. `Type=t2.large`

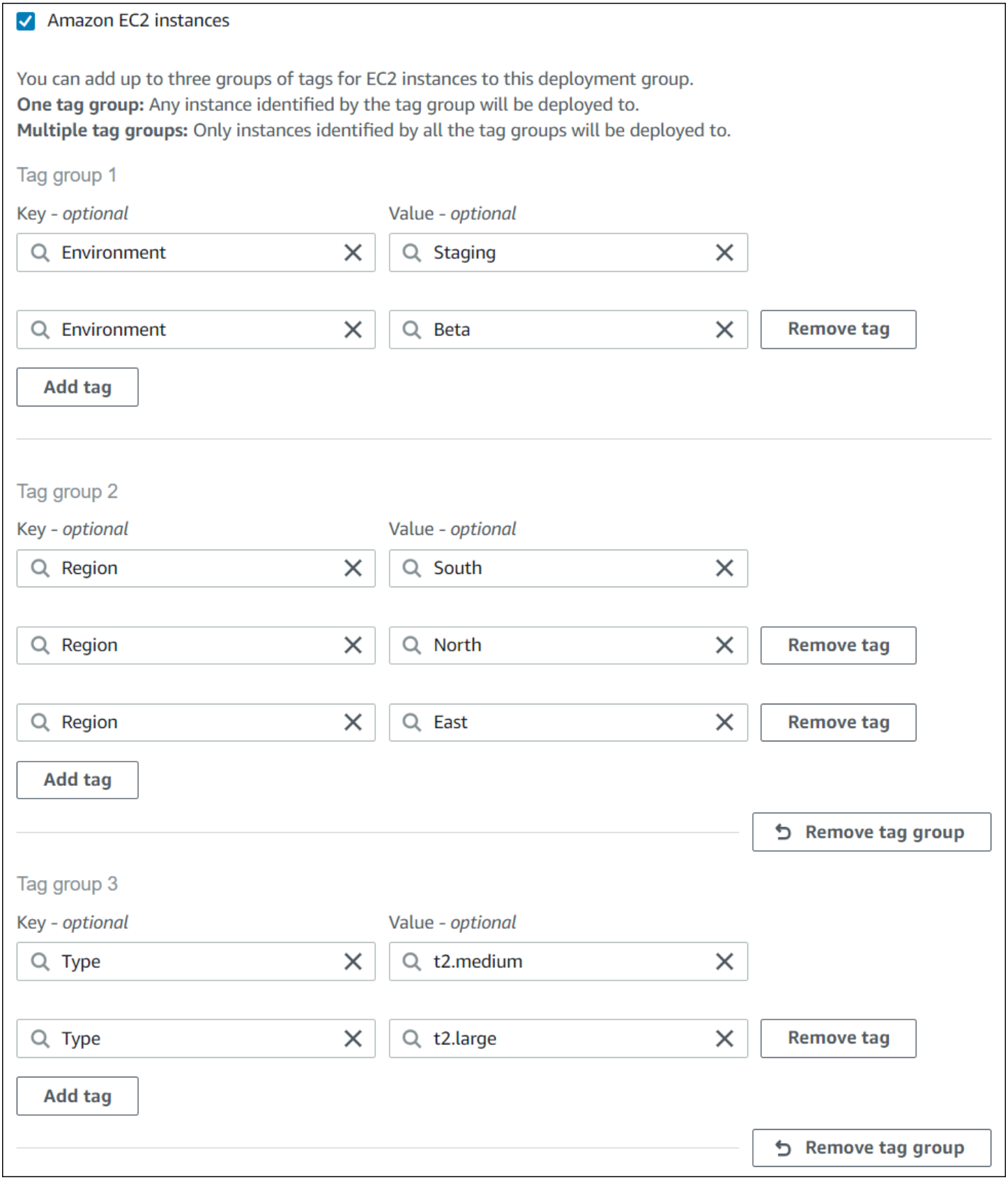
Sebagai ilustrasi, instance dengan grup tag berikut akan termasuk di antara yang termasuk dalam grup penerapan:

- `Environment=Beta`, `Region=North`,`Type=t2.medium`
- `Environment=Staging`,`Region=East`,`Type=t2.large`
- `Environment=Staging`,`Region=South`,`Type=t2.large`

Contoh dengan grup tag berikut tidak akan disertakan dalam grup penyebaran. Nilai kunci yang disorot menyebabkan instance dikecualikan:

- `Environment=Beta`, Wilayah = Barat, `Type=t2.medium`
- `Environment=Staging`,**`Region=East`**, Jenis = t2.micro
- Lingkungan = Produksi,, `Region=South` `Type=t2.large`

CodeDeploy tampilan pengaturan konsol:


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

Struktur 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"
               }
```

```
            ],
         ]
      },
```

Bekerja dengan instans Amazon EC2 untuk CodeDeploy

Instans Amazon EC2 adalah lingkungan komputasi virtual yang Anda buat dan konfigurasikan menggunakan Amazon Elastic Compute Cloud. Amazon EC2 menyediakan kapasitas komputasi yang dapat diskalakan di Cloud. AWS Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan untuk penerapan Anda CodeDeploy .

Untuk informasi selengkapnya tentang Amazon EC2, lihat Panduan Memulai Amazon EC2.

Petunjuk di bagian ini menunjukkan cara membuat dan mengonfigurasi instans Amazon EC2 untuk digunakan dalam penerapan Anda. CodeDeploy

Topik

- Buat instans Amazon EC2 untuk (CodeDeploy AWS CLI atau konsol Amazon EC2)
- Buat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat)
- Konfigurasikan instans Amazon EC2 agar berfungsi CodeDeploy

Buat instans Amazon EC2 untuk (CodeDeploy AWS CLI atau konsol Amazon EC2)

Petunjuk ini menunjukkan cara meluncurkan instans Amazon EC2 baru yang dikonfigurasi untuk digunakan dalam CodeDeploy penerapan.

Anda dapat menggunakan CloudFormation template kami untuk meluncurkan instans Amazon EC2 yang menjalankan Amazon Linux atau Windows Server yang sudah dikonfigurasi untuk digunakan dalam CodeDeploy penerapan. Kami tidak menyediakan CloudFormation template untuk instans Amazon EC2 yang menjalankan Ubuntu Server atau Red Hat Enterprise Linux (RHEL). Untuk alternatif penggunaan template, lihatBekerja dengan instance untuk CodeDeploy.

Anda dapat menggunakan konsol Amazon EC2 AWS CLI, atau Amazon EC2 API untuk meluncurkan instans Amazon EC2.

Luncurkan instans Amazon EC2 (konsol)

Prasyarat

Jika Anda belum melakukannya, ikuti petunjuk Memulai dengan CodeDeploy untuk mengatur dan mengkonfigurasi AWS CLI dan membuat profil instans IAM.

Luncurkan instans Amazon EC2

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi, pilih Instans, lalu pilih Launch Instance.
3. Pada Langkah 1: Pilih halaman Amazon Machine Image (AMI), dari tab Mulai Cepat, cari sistem operasi dan versi yang ingin Anda gunakan, lalu pilih Pilih. Anda harus memilih sistem operasi Amazon EC2 AMI yang didukung oleh. CodeDeploy Untuk informasi selengkapnya, lihat Sistem operasi yang didukung oleh CodeDeploy agen.
4. Pada Langkah 2: Pilih halaman Jenis Instance, pilih jenis instans Amazon EC2 yang tersedia, lalu pilih Berikutnya: Konfigurasi Detail Instance.
5. Pada halaman Langkah 3: Konfigurasikan Detail Instance, dalam daftar peran IAM, pilih peran instans IAM yang Anda buat. Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda Jika Anda menggunakan nama peran yang disarankan, pilih**`CodeDeployDemo-EC2-Instance-Profile`**. Jika Anda membuat nama peran Anda sendiri, pilih itu.

   > **Note**
   >
   > Jika virtual private cloud (VPC) default tidak ditampilkan dalam daftar Jaringan, Anda harus memilih atau membuat Amazon VPC dan subnet. Pilih Buat VPC baru atau Buat subnet baru atau keduanya. Untuk informasi selengkapnya, lihat VPC dan subnet Anda.

6. Pilih Berikutnya: Tambahkan Penyimpanan.
7. Biarkan Langkah 4: Tambahkan halaman Penyimpanan tidak berubah, dan pilih Berikutnya: Tambahkan Tag.
8. Pada halaman Langkah 5: Menambahkan dengan menandai, memilih Menambahkan dengan menandai.
9. Di kotak Kunci, ketik**`Name`**. Dalam tipe kotak Nilai**`CodeDeployDemo`**.

> **Important**
>
> Isi kotak Kunci dan Nilai peka huruf besar/kecil.

10. Pilih Berikutnya: Konfigurasi Grup Keamanan..
11. Pada halaman Langkah 6: Konfigurasikan Grup Keamanan, biarkan opsi Buat grup keamanan baru dipilih.

    Peran SSH default dikonfigurasi untuk instans Amazon EC2 yang menjalankan Amazon Linux, Ubuntu Server, atau RHEL. Peran RDP default dikonfigurasi untuk instans Amazon EC2 yang menjalankan Windows Server.
12. Jika Anda ingin membuka port HTTP, pilih tombol Add Rule, dan dari daftar drop-down Type, pilih**HTTP**. Terima nilai Sumber default Custom 0.0.0. 0/0, lalu pilih Review and Launch.

    > **Note**
    >
    > Dalam lingkungan produksi, kami merekomendasikan untuk membatasi akses ke port SSH, RDP, dan HTTP, alih-alih menentukan Anywhere 0.0.0. 0/0. CodeDeploy tidak memerlukan akses port yang tidak dibatasi dan tidak memerlukan akses HTTP. Untuk informasi selengkapnya, lihat Tips untuk mengamankan instans Amazon EC2 Anda.

    Jika kotak dialog Boot from General Purpose (SSD) muncul, ikuti petunjuknya, lalu pilih Berikutnya.
13. Biarkan halaman Langkah 7: Tinjau Peluncuran Instans tidak berubah, dan pilih Luncurkan.
14. Dalam kotak dialog Select an existing key pair atau create a new key pair, pilih Pilih key pair yang ada atau Create a new key pair. Jika Anda sudah mengonfigurasi pasangan kunci instance Amazon EC2, Anda dapat memilihnya di sini.

    Jika Anda belum memiliki pasangan kunci instance Amazon EC2, pilih Buat pasangan kunci baru dan berikan nama yang dapat dikenali. Pilih Download Key Pair untuk mengunduh key pair instans Amazon EC2 ke komputer Anda.

> **Important**
>
> Anda harus memiliki key pair jika ingin mengakses instans Amazon EC2 Anda dengan SSH atau RDP.

15. Pilih Luncurkan Instans.
16. Pilih ID untuk instans Amazon EC2 Anda. Jangan lanjutkan sampai instance diluncurkan dan lulus semua pemeriksaan.

Instal CodeDeploy agen

CodeDeploy Agen harus diinstal pada instans Amazon EC2 Anda sebelum menggunakannya dalam CodeDeploy penerapan. Untuk informasi selengkapnya, lihat Instal CodeDeploy agen.

> **Note**
>
> Anda dapat mengonfigurasi instalasi otomatis dan pembaruan CodeDeploy agen saat membuat grup penyebaran di konsol.

Luncurkan instans Amazon EC2 (CLI)

Prasyarat

Jika Anda belum melakukannya, ikuti petunjuk Memulai dengan CodeDeploy untuk mengatur dan mengkonfigurasi AWS CLI dan membuat profil instans IAM.

Luncurkan instans Amazon EC2

1. Untuk Windows Server saja Jika Anda membuat instans Amazon EC2 yang menjalankan Windows Server, panggil create-security-group dan authorize-security-group-ingress perintah untuk membuat grup keamanan yang memungkinkan akses RDP (yang tidak diizinkan secara default) dan, sebagai alternatif, akses HTTP. Misalnya, untuk membuat grup keamanan bernama CodeDeployDemo-Windows-Security-Group, jalankan perintah berikut, satu per satu:

```
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**
>
> Untuk tujuan demonstrasi, perintah ini membuat grup keamanan yang memungkinkan akses tidak terbatas untuk RDP melalui port 3389 dan, sebagai alternatif, HTTP melalui port 80. Sebagai praktik terbaik, kami sarankan untuk membatasi akses ke port RDP dan HTTP. CodeDeploy tidak memerlukan akses port yang tidak dibatasi dan tidak memerlukan akses HTTP. Untuk informasi selengkapnya, lihat Tips untuk mengamankan instans Amazon EC2 Anda.

2. Panggil run-instances perintah untuk membuat dan meluncurkan instans Amazon EC2.

   Sebelum Anda memanggil perintah ini, Anda perlu mengumpulkan yang berikut:

   - ID Amazon Machine Image (AMI) (*`ami-id`*) yang Anda gunakan untuk instance. Untuk mendapatkan ID, lihat Menemukan AMI yang cocok.
   - Nama jenis instans Amazon EC2 (*`instance-type`*) yang Anda buat, seperti. `t1.micro` Untuk daftar, lihat Jenis instans Amazon EC2.
   - Nama profil instans IAM dengan izin untuk mengakses bucket Amazon S3 tempat CodeDeploy file instalasi agen untuk Wilayah Anda disimpan.

     Untuk informasi tentang membuat profil instans IAM, lihatLangkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.
   - Nama key pair (*`key-name`*) instans Amazon EC2 untuk mengaktifkan akses SSH ke instans Amazon EC2 yang menjalankan Amazon Linux, Ubuntu Server, atau akses RHEL atau RDP ke instans Amazon EC2 yang menjalankan Windows Server.

> **Important**
>
> Ketik nama key pair saja, bukan ekstensi file key pair. Misalnya, my-keypair, bukan my-keypair.pem.

Untuk menemukan nama key pair, buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2 Di panel navigasi, di bawah Network & Security, pilih Key Pairs, dan catat nama key pair dalam daftar.

Untuk membuat key pair, lihat Membuat key pair menggunakan Amazon EC2. Pastikan Anda membuat key pair di salah satu Region yang terdaftar di Region dan endpoint di Referensi Umum AWS. Jika tidak, Anda tidak akan dapat menggunakan key pair instans Amazon EC2 dengan. CodeDeploy

Untuk Amazon Linux, RHEL, dan Server Ubuntu

Untuk memanggil run-instances perintah untuk meluncurkan instans Amazon EC2 yang menjalankan Amazon Linux, Ubuntu Server, atau RHEL dan lampirkan profil instans IAM yang Anda buat. Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda Contoh:

```
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**
>
> Perintah ini membuat grup keamanan default untuk instans Amazon EC2 yang memungkinkan akses ke beberapa port, termasuk akses tidak terbatas untuk SSH melalui port 22 dan, sebagai alternatif, HTTP melalui port 80. Sebagai praktik terbaik, kami sarankan membatasi akses ke port SSH dan HTTP saja. CodeDeploy tidak memerlukan akses port yang tidak terbatas dan tidak memerlukan akses port HTTP.

> Untuk informasi selengkapnya, lihat Tips untuk mengamankan instans Amazon EC2 Anda.

Untuk Windows Server

Untuk memanggil run-instances perintah untuk meluncurkan instans Amazon EC2 yang menjalankan Windows Server dan melampirkan profil instans IAM yang Anda buatLangkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda, dan tentukan nama grup keamanan yang Anda buat di Langkah 1. Contoh:

```
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
```

Perintah ini meluncurkan instans Amazon EC2 tunggal dengan tipe AMI, key pair, dan instance yang ditentukan, dengan profil instans IAM yang ditentukan, dan menjalankan skrip yang ditentukan selama peluncuran.

3. Perhatikan nilai `InstanceID` dalam output. Jika Anda lupa nilai ini, Anda bisa mendapatkannya nanti dengan memanggil describe-instances perintah terhadap key pair instans Amazon EC2.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query
 "Reservations[*].Instances[*].[InstanceId]" --output text
```

Gunakan ID instans untuk memanggil create-tags perintah, yang menandai instans Amazon EC2 sehingga CodeDeploy dapat menemukannya nanti selama penerapan. Dalam contoh berikut, tag diberi nama**`CodeDeployDemo`**, tetapi Anda dapat menentukan tag instans Amazon EC2 yang Anda inginkan.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

Anda dapat menerapkan beberapa tag ke sebuah instance secara bersamaan. Contoh:

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance
 Key=Region,Value=West Key=Environment,Value=Beta
```

Untuk memverifikasi instans Amazon EC2 telah diluncurkan dan melewati semua pemeriksaan, gunakan ID instans untuk memanggil perintah. describe-instance-status

```
aws ec2 describe-instance-status --instance-ids instance-id --query
 "InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Jika instance telah diluncurkan dan melewati semua pemeriksaan, ok muncul di output.

Instal CodeDeploy agen

CodeDeploy Agen harus diinstal pada instans Amazon EC2 Anda sebelum menggunakannya dalam CodeDeploy penerapan. Untuk informasi selengkapnya, lihat Instal CodeDeploy agen.

> **Note**
>
> Anda dapat mengonfigurasi instalasi otomatis dan pembaruan CodeDeploy agen saat membuat grup penyebaran di konsol.

Buat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat)

Anda dapat menggunakan CloudFormation template kami untuk meluncurkan instans Amazon EC2 dengan cepat yang menjalankan Amazon Linux atau Windows Server. Anda dapat menggunakan AWS CLI, CodeDeploy konsol, atau AWS API untuk meluncurkan instance dengan template. Selain meluncurkan instance, template melakukan hal berikut:

- Menginstruksikan CloudFormation untuk memberikan izin instans untuk berpartisipasi dalam CodeDeploy penerapan.
- Menandai instance sehingga CodeDeploy dapat menemukannya selama penerapan.
- Menginstal dan menjalankan CodeDeploy agen pada instance.

Anda tidak harus menggunakan kami CloudFormation untuk menyiapkan instans Amazon EC2. Untuk alternatif, lihatBekerja dengan instance untuk CodeDeploy.

Kami tidak menyediakan CloudFormation template untuk instans Amazon EC2 yang menjalankan Ubuntu Server atau Red Hat Enterprise Linux (RHEL).

Topik

- Sebelum Anda mulai
- Luncurkan instans Amazon EC2 dengan CloudFormation template (konsol)
- Luncurkan instans Amazon EC2 dengan CloudFormation templat (AWS CLI)

Sebelum Anda mulai

Sebelum Anda dapat menggunakan CloudFormation template untuk meluncurkan instans Amazon EC2, pastikan Anda menyelesaikan langkah-langkah berikut.

1. Pastikan Anda telah membuat pengguna administratif, seperti yang dijelaskan dalamLangkah 1: Menyiapkan. Double-check bahwa pengguna memiliki izin minimum berikut dan menambahkan yang tidak ada:

   - pembentukan awan: *
   - penyebaran kode: *
   - EC2: *
   - saya: AddRoleToInstanceProfile
   - saya: CreateInstanceProfile
   - saya: CreateRole
   - saya: DeleteInstanceProfile
   - saya: DeleteRole
   - saya: DeleteRolePolicy
   - saya: GetRole
   - saya: DeleteRolePolicy
   - saya: PutRolePolicy
   - saya: RemoveRoleFromInstanceProfile
2. Pastikan Anda memiliki instance key pair untuk mengaktifkan akses SSH ke instans Amazon EC2 yang menjalankan Amazon Linux atau akses RDP ke instance yang menjalankan Windows Server.

   Untuk menemukan nama key pair, buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2 Di panel navigasi, di bawah Network & Security, pilih Key Pairs, dan catat nama key pair dalam daftar.

Untuk membuat key pair baru, lihat Membuat key pair menggunakan Amazon EC2. Pastikan key pair dibuat di salah satu Wilayah yang terdaftar di Region dan endpoint di Referensi Umum AWS. Jika tidak, Anda tidak dapat menggunakan instance key pair with CodeDeploy.

Luncurkan instans Amazon EC2 dengan CloudFormation template (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.

   > **Important**
   >
   > Masuk ke akun Konsol Manajemen AWS dengan akun yang sama dengan yang Anda gunakanMemulai dengan CodeDeploy. Pada bilah navigasi, di pemilih Wilayah, pilih salah satu Wilayah yang terdaftar di Wilayah dan titik akhir di. Referensi Umum AWS CodeDeploy hanya mendukung Wilayah ini.

2. Pilih Buat tumpukan.
3. Di Pilih templat, pilih Tentukan URL template Amazon S3. Di kotak, ketik lokasi CloudFormation templat untuk Wilayah Anda, lalu pilih Berikutnya.

Region	Lokasi CloudFormation template
Wilayah AS Timur (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah AS Timur (Virginia Utara)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah AS Barat (California Utara)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah AS Barat (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Kanada (Pusat)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Irlandia)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Hong Kong)	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pacific (Tokyo)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pacific (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah Asia Pasifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Amerika Selatan (Sao Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

4. Di kotak nama Stack, ketikkan nama untuk tumpukan (misalnya,**`CodeDeployDemoStack`**).
5. Di Parameter, ketik berikut ini, lalu pilih Berikutnya.

   - Untuk InstanceCount, ketik jumlah instance yang ingin Anda luncurkan. (Kami sarankan Anda meninggalkan default 1.)
   - Untuk InstanceType, ketik jenis instance yang ingin Anda luncurkan (atau biarkan default t1.micro).
   - Untuk KeyPairName, ketik nama instance key pair. Ketik nama key pair saja, bukan ekstensi file key pair.
   - Untuk OperatingSystemkotak, ketik **`Windows`** untuk meluncurkan instance yang menjalankan Windows Server (atau biarkan default Linux).
   - Untuk SSHLocation, ketik rentang alamat IP yang akan digunakan untuk menghubungkan ke instance dengan SSH atau RDP (atau biarkan default 0.0.0. 0/0).

   > **Important**
   >
   > Default **`0.0.0.0/0`** disediakan untuk tujuan demonstrasi saja. CodeDeploy tidak mengharuskan instans Amazon EC2 memiliki akses tidak terbatas ke port. Sebagai

praktik terbaik, kami sarankan untuk membatasi akses ke port SSH (dan HTTP). Untuk informasi selengkapnya, lihat Tips untuk mengamankan instans Amazon EC2 Anda.

- Untuk TagKey, ketik kunci tag instance yang CodeDeploy akan digunakan untuk mengidentifikasi instance selama penerapan (atau biarkan default Nama).
- Untuk TagValue, ketik nilai tag instance yang CodeDeploy akan digunakan untuk mengidentifikasi instance selama penerapan (atau biarkan default). CodeDeployDemo

6. Pada halaman Opsi, biarkan kotak opsi kosong, dan pilih Berikutnya.

> **Important**
>
> CloudFormation tag berbeda dari CodeDeploy tag. CloudFormation menggunakan tag untuk menyederhanakan administrasi infrastruktur Anda. CodeDeploy menggunakan tag untuk mengidentifikasi instans Amazon EC2. Anda menentukan CodeDeploy tag pada halaman Tentukan Parameter.

7. Pada halaman Tinjauan, di Kemampuan, pilih kotak Saya mengakui yang CloudFormation mungkin membuat sumber daya IAM, lalu pilih Buat.

   Setelah CloudFormation membuat tumpukan dan meluncurkan instans Amazon EC2, di CloudFormation konsol, CREATE_COMPLETE akan ditampilkan di kolom Status. Proses ini dapat memakan waktu beberapa menit.

Untuk memverifikasi CodeDeploy agen berjalan di instans Amazon EC2, lihatMengelola operasi CodeDeploy agen, lalu lanjutkan ke. Buat aplikasi dengan CodeDeploy

Luncurkan instans Amazon EC2 dengan CloudFormation templat (AWS CLI)

1. Gunakan CloudFormation template kami dalam panggilan ke create-stack perintah. Tumpukan ini akan meluncurkan instans Amazon EC2 baru dengan CodeDeploy agen diinstal.

   Untuk meluncurkan instans Amazon EC2 yang menjalankan 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
```

Untuk meluncurkan instans Amazon EC2 yang menjalankan 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*adalah nama instance key pair. Ketik nama key pair saja, bukan ekstensi file key pair.

*template-url*adalah lokasi CloudFormation template untuk Wilayah Anda:

Region	Lokasi CloudFormation template
Wilayah AS Timur (Ohio)	`http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah AS Timur (Virginia Utara)	`http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah AS Barat (California Utara)	`http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah AS Barat (Oregon)	`http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Kanada (Pusat)	`http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Irlandia)	`http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (London)	`http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Paris)	`http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Eropa (Frankfurt)	`http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah Israel (Tel Aviv)	`http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Hong Kong)	`http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pacific (Tokyo)	`http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Seoul)	`http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Singapura)	`http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pacific (Sydney)	`http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json`

Region	Lokasi CloudFormation template
Wilayah Asia Pasifik (Melbourne)	`https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Asia Pasifik (Mumbai)	`http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json`
Wilayah Amerika Selatan (Sao Paulo)	`aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json`

Perintah ini membuat CloudFormation tumpukan bernama**`CodeDeployDemoStack`**, menggunakan CloudFormation template di bucket Amazon S3 yang ditentukan. Instans Amazon EC2 didasarkan pada jenis instans t1.micro, tetapi Anda dapat menggunakan jenis apa pun. Ini ditandai dengan nilainya**`CodeDeployDemo`**, tetapi Anda dapat menandainya dengan nilai apa pun. Ini memiliki instance key pair yang ditentukan diterapkan.

2. Panggil describe-stacks perintah untuk memverifikasi CloudFormation tumpukan **`CodeDeployDemoStack`** bernama berhasil dibuat:

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query
 "Stacks[0].StackStatus" --output text
```

Jangan melanjutkan sampai nilai `CREATE_COMPLETE` dikembalikan.

Untuk memverifikasi CodeDeploy agen berjalan di instans Amazon EC2, lihatMengelola operasi CodeDeploy agen, lalu lanjutkan ke. Buat aplikasi dengan CodeDeploy

Konfigurasikan instans Amazon EC2 agar berfungsi CodeDeploy

Petunjuk ini menunjukkan cara mengonfigurasi instans Amazon EC2 yang menjalankan Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL), atau Windows Server untuk digunakan dalam penerapan. CodeDeploy

> **Note**
>
> Jika Anda tidak memiliki instans Amazon EC2, Anda dapat menggunakan CloudFormation template untuk meluncurkan satu yang menjalankan Amazon Linux atau Windows Server. Kami tidak menyediakan template untuk Ubuntu Server atau RHEL.

Langkah 1: Verifikasi bahwa profil instans IAM dilampirkan ke instans Amazon EC2

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi, di Instans, pilih Instans.
3. Jelajahi dan pilih instans Amazon EC2 Anda dalam daftar.
4. Di panel detail, pada tab Deskripsi, catat nilai di bidang peran IAM, lalu lanjutkan ke bagian berikutnya.

   Jika bidang kosong, Anda dapat melampirkan profil instans IAM ke instance. Untuk selengkapnya, lihat Melampirkan peran IAM ke instance.

Langkah 2: Verifikasi profil instans IAM terlampir memiliki izin akses yang benar

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Jelajahi dan pilih nama peran IAM yang Anda catat di langkah 4 dari bagian sebelumnya.

> **Note**
>
> Jika Anda ingin menggunakan peran layanan yang dihasilkan oleh CloudFormation template, bukan yang Anda buat dengan mengikuti petunjukLangkah 2: Buat peran layanan untuk CodeDeploy, perhatikan hal berikut:

> Di beberapa versi CloudFormation template kami, nama tampilan profil instans IAM yang dihasilkan dan dilampirkan ke instans Amazon EC2 tidak sama dengan nama tampilan di konsol IAM. Misalnya, profil instans IAM mungkin memiliki nama tampilan`CodeDeploySampleStack-expnyi6-InstanceRoleInstanceProfile-IK8J8A9123EX`, sedangkan profil instans IAM di konsol IAM mungkin memiliki nama tampilan. `CodeDeploySampleStack-expnyi6-InstanceRole-C5P33V1L64EX` Untuk membantu Anda mengidentifikasi profil instance di konsol IAM, Anda akan melihat awalan dari `CodeDeploySampleStack-expnyi6-InstanceRole` is sama untuk keduanya. Untuk informasi tentang mengapa nama tampilan ini mungkin berbeda, lihat Profil instans.

4. Pilih tab Trust Relationships (Hubungan Kepercayaan). Jika tidak ada entri di Entitas Tepercaya yang membaca Penyedia identitas ec2.amazonaws.com, Anda tidak dapat menggunakan instans Amazon EC2 ini. Hentikan dan buat instans Amazon EC2 menggunakan informasi di. Bekerja dengan instance untuk CodeDeploy

   Jika ada entri yang bertuliskan Penyedia identitas ec2.amazonaws.com, dan Anda hanya menyimpan aplikasi Anda di GitHub repositori, lalu lanjutkan ke. Langkah 3: Tandai instans Amazon EC2

   Jika ada entri yang bertuliskan Penyedia identitas ec2.amazonaws.com, dan Anda menyimpan aplikasi Anda di bucket Amazon S3, pilih tab Izin.

5. Jika ada kebijakan di area Kebijakan izin, perluas kebijakan, lalu pilih Edit kebijakan.

6. Pilih tab JSON. Jika Anda menyimpan aplikasi Anda di bucket Amazon S3, pastikan `"s3:Get*"` dan `"s3:List*"` berada dalam daftar tindakan yang ditentukan.

   Mungkin terlihat seperti ini:

```
{"Statement":[{"Resource":"*","Action":[
  ... Some actions may already be listed here ...
  "s3:Get*","s3:List*"
  ... Some more actions may already be listed here ...
  ],"Effect":"Allow"}]}
```

   Atau mungkin terlihat seperti ini:

JSON

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

Jika `"s3:Get*"` dan tidak `"s3:List*"` ada dalam daftar tindakan yang ditentukan, pilih Edit untuk menambahkannya, lalu pilih Simpan. (Jika tidak ada `"s3:Get*"` atau `"s3:List*"` merupakan tindakan terakhir dalam daftar, pastikan untuk menambahkan koma setelah tindakan, sehingga dokumen kebijakan memvalidasi.)

> **Note**
>
> Kami menyarankan Anda membatasi kebijakan ini hanya untuk bucket Amazon S3 yang harus diakses instans Amazon EC2 Anda. Pastikan untuk memberikan akses ke bucket Amazon S3 yang berisi agen. CodeDeploy Jika tidak, kesalahan dapat terjadi ketika CodeDeploy agen diinstal atau diperbarui pada instance. Untuk memberikan akses profil instans IAM hanya ke beberapa bucket kit CodeDeploy sumber daya di Amazon S3, gunakan kebijakan berikut, tetapi hapus baris untuk bucket yang ingin Anda cegah aksesnya:
>
> ```
> {
>   "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/*"
      ]
    }
  ]
}
```

Langkah 3: Tandai instans Amazon EC2

Untuk petunjuk tentang cara menandai instans Amazon EC2 sehingga CodeDeploy dapat menemukannya selama penerapan, lihat Bekerja dengan tag di konsol, lalu kembali ke halaman ini.

> **Note**
>
> Anda dapat menandai instans Amazon EC2 dengan kunci dan nilai apa pun yang Anda suka. Pastikan untuk menentukan kunci dan nilai ini saat Anda menerapkannya.

Langkah 4: Instal AWS CodeDeploy agen pada instans Amazon EC2

Untuk petunjuk tentang cara menginstal CodeDeploy agen pada instans Amazon EC2 dan memverifikasinya sedang berjalan, lihatMengelola operasi CodeDeploy agen, lalu lanjutkan ke. Buat aplikasi dengan CodeDeploy

Bekerja dengan instans lokal untuk CodeDeploy

Instans lokal adalah perangkat fisik apa pun yang bukan instans Amazon EC2 yang dapat menjalankan CodeDeploy agen dan terhubung ke titik akhir layanan AWS publik.

Menerapkan revisi CodeDeploy aplikasi ke instans lokal melibatkan dua langkah utama:

- Langkah 1 — Konfigurasikan setiap instans lokal, daftarkan CodeDeploy, lalu beri tag.
- Langkah 2 — Menyebarkan revisi aplikasi ke instans lokal.

> **Note**
>
> Untuk bereksperimen dengan membuat dan menerapkan contoh revisi aplikasi ke instans lokal yang dikonfigurasi dan terdaftar dengan benar, lihat. Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux) Untuk informasi tentang instans lokal dan cara kerjanya CodeDeploy, lihat. Working with On-Premises Instances

Jika tidak ingin instance lokal digunakan dalam penerapan lagi, Anda dapat menghapus tag instans lokal dari grup penerapan. Untuk pendekatan yang lebih kuat, hapus tag instance lokal dari instance. Anda juga dapat membatalkan pendaftaran instans lokal secara eksplisit sehingga tidak dapat digunakan lagi dalam penerapan apa pun. Untuk informasi selengkapnya, lihat Mengelola operasi instans lokal di CodeDeploy.

Petunjuk di bagian ini menunjukkan cara mengonfigurasi instans lokal, lalu mendaftar dan menandainya CodeDeploy agar dapat digunakan dalam penerapan. Bagian ini juga menjelaskan cara menggunakan CodeDeploy untuk mendapatkan informasi tentang instans lokal dan membatalkan pendaftaran instans lokal setelah Anda tidak lagi berencana untuk menerapkannya.

Topik

- Prasyarat untuk mengonfigurasi instans lokal
- Daftarkan instans lokal dengan CodeDeploy
- Mengelola operasi instans lokal di CodeDeploy

Prasyarat untuk mengonfigurasi instans lokal

Prasyarat berikut harus dipenuhi sebelum Anda dapat mendaftarkan instans lokal.

> **Important**
>
> Jika Anda menggunakan perintah register-on-premises-instance dan kredensi sementara yang diperbarui secara berkala yang dihasilkan dengan (), ada prasyarat lain. AWS Security Token Service AWS STS Untuk informasi, lihat Prasyarat pendaftaran ARN sesi IAM.

Persyaratan perangkat

Perangkat yang ingin Anda siapkan, daftarkan, dan beri tag sebagai instans lokal CodeDeploy harus menjalankan sistem operasi yang didukung. Untuk daftarnya, lihat Sistem operasi yang didukung oleh CodeDeploy agen.

Jika sistem operasi Anda tidak didukung, CodeDeploy agen tersedia sebagai open source bagi Anda untuk beradaptasi dengan kebutuhan Anda. Untuk informasi lebih lanjut, lihat repositori CodeDeploy agen di. GitHub

Komunikasi keluar

Instans lokal harus dapat terhubung ke titik akhir AWS layanan publik untuk berkomunikasi. CodeDeploy

CodeDeploy Agen berkomunikasi keluar menggunakan HTTPS melalui port 443.

Kontrol administratif

Akun lokal atau jaringan yang digunakan pada instans lokal untuk mengonfigurasi instans lokal harus dapat dijalankan sebagai `sudo` atau `root` (untuk Server Ubuntu) atau sebagai administrator (untuk Windows Server).

Izin IAM

Identitas IAM yang Anda gunakan untuk mendaftarkan instans lokal harus diberikan izin untuk menyelesaikan pendaftaran (dan untuk membatalkan pendaftaran instans lokal, sesuai kebutuhan).

Selain kebijakan yang dijelaskan dalamLangkah 3: Batasi izin CodeDeploy pengguna, pastikan identitas IAM panggilan memiliki kebijakan tambahan berikut yang dilampirkan.

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": "*"
    }
  ]
}
```

Untuk informasi tentang cara melampirkan kebijakan IAM, lihat Mengelola kebijakan IAM.

Daftarkan instans lokal dengan CodeDeploy

Untuk mendaftarkan instans lokal, Anda harus menggunakan identitas IAM untuk mengautentikasi permintaan Anda. Anda dapat memilih dari opsi berikut untuk identitas IAM dan metode pendaftaran yang Anda gunakan:

- Gunakan peran IAM ARN untuk mengautentikasi permintaan.
  - Gunakan perintah register-on-premises-instance dan kredensi sementara yang diperbarui secara berkala yang dihasilkan dengan () untuk mengonfigurasi sebagian besar opsi pendaftaran secara manual. AWS Security Token Service AWS STS Opsi ini menawarkan tingkat keamanan tertinggi, karena otentikasi terjadi menggunakan token sementara yang habis waktu dan harus disegarkan secara berkala. Opsi ini direkomendasikan untuk penyebaran produksi dalam berbagai ukuran. Untuk informasi, lihat Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal.
- (Tidak disarankan) Gunakan ARN pengguna IAM untuk mengautentikasi permintaan.
  - Gunakan perintah register untuk proses pendaftaran paling otomatis. Opsi ini hanya boleh digunakan untuk penerapan non-produksi di mana keamanan kurang menjadi perhatian. Opsi ini kurang aman karena menggunakan kredensi statis (permanen) untuk otentikasi. Opsi ini berfungsi dengan baik untuk mendaftarkan satu instance lokal. Untuk informasi, lihat Gunakan perintah register (pengguna IAM ARN) untuk mendaftarkan instance lokal.
  - Gunakan perintah register-on-premises-instance untuk mengonfigurasi sebagian besar opsi pendaftaran secara manual. Cocokkan untuk mendaftarkan sejumlah kecil instans lokal. Untuk informasi, lihat Gunakan perintah register-on-premises-instance (ARN pengguna IAM) untuk mendaftarkan instance lokal.

Topik

- Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal
- Gunakan perintah register (pengguna IAM ARN) untuk mendaftarkan instance lokal
- Gunakan perintah register-on-premises-instance (ARN pengguna IAM) untuk mendaftarkan instance lokal

Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal

Untuk kontrol maksimum atas autentikasi dan pendaftaran instans lokal, Anda dapat menggunakan perintah register-on-premises-instance dan kredensyal sementara yang diperbarui secara berkala yang dihasilkan dengan (). AWS Security Token Service AWS STS Peran IAM statis untuk instance mengasumsikan peran AWS STS kredensyal yang disegarkan ini untuk melakukan operasi penerapan. CodeDeploy

Metode ini paling berguna ketika Anda perlu mendaftarkan sejumlah besar instance. Ini memungkinkan Anda untuk mengotomatiskan proses pendaftaran dengan CodeDeploy. Anda dapat menggunakan sistem identitas dan autentikasi Anda sendiri untuk mengautentikasi instans lokal dan mendistribusikan kredensyal sesi IAM dari layanan ke instans untuk digunakan. CodeDeploy

> **Note**
>
> Atau, Anda dapat menggunakan pengguna IAM bersama yang didistribusikan ke semua instans lokal untuk memanggil AWS STS AssumeRoleAPI guna mengambil kredensyal sesi untuk instance lokal. Metode ini kurang aman dan tidak direkomendasikan untuk lingkungan produksi atau misi kritis.

Gunakan informasi dalam topik berikut untuk mengonfigurasi instans lokal menggunakan kredensyal keamanan sementara yang dihasilkan. AWS STS

Topik

- Prasyarat pendaftaran ARN sesi IAM
- Langkah 1: Buat peran IAM yang akan diasumsikan oleh instans lokal
- Langkah 2: Hasilkan kredensyal sementara untuk instance individual menggunakan AWS STS
- Langkah 3: Tambahkan file konfigurasi ke instans lokal
- Langkah 4: Siapkan instans lokal untuk penerapan CodeDeploy
- Langkah 5: Daftarkan instans lokal dengan CodeDeploy
- Langkah 6: Tandai instans lokal
- Langkah 7: Menerapkan revisi aplikasi ke instans lokal
- Langkah 8: Lacak penerapan ke instans lokal

Prasyarat pendaftaran ARN sesi IAM

Selain prasyarat yang tercantum dalamPrasyarat untuk mengonfigurasi instans lokal, persyaratan tambahan berikut harus dipenuhi:

Izin IAM

Identitas IAM yang Anda gunakan untuk mendaftarkan instans lokal harus diberikan izin untuk melakukan operasi. CodeDeploy Pastikan kebijakan AWSCodeDeployFullAccessterkelola dilampirkan pada identitas IAM. Untuk selengkapnya, lihat kebijakan AWS terkelola di Panduan Pengguna IAM.

Sistem untuk menyegarkan kredensyal sementara

Jika Anda menggunakan ARN sesi IAM untuk mendaftarkan instans lokal, Anda harus memiliki sistem untuk menyegarkan kredensyal sementara secara berkala. Kredensyal sementara kedaluwarsa setelah satu jam atau lebih cepat jika periode yang lebih pendek ditentukan saat kredensyal dihasilkan. Ada dua metode untuk menyegarkan kredensyal:

- Metode 1: Gunakan sistem identitas dan otentikasi yang ada di jaringan perusahaan Anda dengan skrip CRON yang secara berkala melakukan polling identitas dan sistem otentikasi dan menyalin kredensyal sesi terbaru ke instans. Ini memungkinkan Anda untuk mengintegrasikan otentikasi dan struktur identitas Anda AWS tanpa perlu membuat perubahan pada CodeDeploy agen atau layanan untuk mendukung jenis otentikasi yang Anda gunakan di organisasi Anda.
- Metode 2: Jalankan tugas CRON secara berkala pada instance untuk memanggil AWS STS AssumeRoletindakan dan menulis kredensyal sesi ke file yang dapat diakses CodeDeploy agen. Metode ini masih memerlukan penggunaan pengguna IAM dan menyalin kredensyal ke instans lokal, tetapi Anda dapat menggunakan kembali pengguna dan kredensyal IAM yang sama di seluruh armada instans lokal.

Untuk informasi tentang membuat dan bekerja dengan AWS STS kredensyal, lihat Referensi AWS Security Token Service API dan Menggunakan kredensyal keamanan sementara untuk meminta akses ke sumber daya. AWS

Langkah 1: Buat peran IAM yang akan diasumsikan oleh instans lokal

Anda dapat menggunakan konsol IAM AWS CLI atau IAM untuk membuat peran IAM yang akan digunakan oleh instans lokal untuk mengautentikasi dan berinteraksi. CodeDeploy

Anda hanya perlu membuat peran IAM tunggal. Setiap instans lokal Anda dapat mengambil peran ini untuk mengambil kredensyal keamanan sementara yang memberikan izin yang diberikan untuk peran ini.

Peran yang Anda buat akan memerlukan izin berikut untuk mengakses file yang diperlukan untuk menginstal CodeDeploy agen:

JSON

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

Sebaiknya batasi kebijakan ini hanya pada bucket Amazon S3 yang perlu diakses instans lokal. Jika Anda membatasi kebijakan ini, pastikan untuk memberikan akses ke bucket Amazon S3 yang berisi agen. CodeDeploy Jika tidak, kesalahan dapat terjadi setiap kali CodeDeploy agen diinstal atau diperbarui pada instans lokal. Untuk informasi tentang mengontrol akses ke bucket Amazon S3, lihat Mengelola izin akses ke sumber daya Amazon S3 Anda.

Untuk membuat peran IAM

1. Panggil perintah create-role menggunakan `--role-name` opsi untuk menentukan nama untuk peran IAM (misalnya,`CodeDeployInstanceRole`) dan `--assume-role-policy-document` opsi untuk memberikan izin.

   Saat Anda membuat peran IAM untuk instance ini, Anda mungkin memberinya nama `CodeDeployInstanceRole` dan menyertakan izin yang diperlukan dalam file bernama: `CodeDeployRolePolicy.json`

```
aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-
document file://CodeDeployRolePolicy.json
```

2. Dalam output panggilan ke create-role perintah, perhatikan nilai bidang ARN. Contoh:

```
arn:aws:iam::123456789012:role/CodeDeployInstanceRole
```

Anda akan memerlukan peran ARN saat menggunakan AWS STS AssumeRoleAPI untuk menghasilkan kredensyal jangka pendek untuk setiap instance.

Untuk informasi selengkapnya tentang membuat peran IAM, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM.

Untuk informasi tentang menetapkan izin ke peran yang ada, lihat put-role-policy di Referensi Perintah.AWS CLI

Langkah 2: Hasilkan kredensyal sementara untuk instance individual menggunakan AWS STS

Sebelum Anda membuat kredensyal sementara yang akan digunakan untuk mendaftarkan instans lokal, Anda harus membuat atau memilih identitas IAM (pengguna atau peran) yang akan Anda buat kredensyal sementara. `sts:AssumeRole`lzin harus disertakan dalam pengaturan kebijakan untuk identitas IAM ini.

Untuk informasi tentang pemberian `sts:AssumeRole` izin ke identitas IAM, lihat Membuat peran untuk mendelegasikan izin ke layanan dan. AWS AssumeRole

Ada dua cara untuk menghasilkan kredensyal sementara:

- Gunakan perintah assume-role dengan. AWS CLI Contoh:

```
aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-
session-name session-name
```

Di mana:

- *12345ACCOUNT*adalah nomor akun 12 digit untuk organisasi Anda.
- *role-arn*adalah ARN dari peran yang akan diasumsikan, yang Anda hasilkan. Langkah 1: Buat peran IAM yang akan diasumsikan oleh instans lokal

- *session-name*adalah nama yang ingin Anda berikan untuk sesi peran yang Anda buat sekarang.

> **Note**
>
> Jika Anda menggunakan skrip CRON yang secara berkala melakukan polling identitas dan sistem otentikasi dan menyalin kredensyal sesi terbaru ke instance (metode 1 untuk menyegarkan kredensyal sementara yang dijelaskan dalamPrasyarat pendaftaran ARN sesi IAM), Anda dapat menggunakan SDK apa pun yang didukung untuk memanggil. AWS AssumeRole

- Gunakan alat yang disediakan oleh AWS.

  Alat aws-codedeploy-session-helper menghasilkan AWS STS kredensyal dan menuliskannya ke file yang Anda tempatkan pada instance. Alat ini paling cocok untuk metode 2 untuk menyegarkan kredensyal sementara yang dijelaskan dalam. Prasyarat pendaftaran ARN sesi IAM Dalam metode ini, alat aws-codedeploy-session-helper ditempatkan pada setiap instance dan menjalankan perintah menggunakan izin pengguna IAM. Setiap instance menggunakan kredensyal pengguna IAM yang sama dalam hubungannya dengan alat ini.

  Untuk informasi selengkapnya, lihat repositori GitHub aws-codedeploy-session-helper.

> **Note**
>
> Setelah Anda membuat kredensyal sesi IAM, letakkan kredensyal sesi IAM di lokasi mana pun di instans lokal. Pada langkah berikutnya, Anda akan mengonfigurasi CodeDeploy agen untuk mengakses kredensyal di lokasi ini.

Sebelum melanjutkan, pastikan sistem yang akan Anda gunakan untuk menyegarkan kredensyal sementara secara berkala sudah ada. Jika kredensyal sementara tidak di-refresh, penerapan ke instans lokal akan gagal. Untuk informasi selengkapnya, lihat “Sistem untuk menyegarkan kredensyal sementara” di. Prasyarat pendaftaran ARN sesi IAM

Langkah 3: Tambahkan file konfigurasi ke instans lokal

Tambahkan file konfigurasi ke instans lokal, menggunakan izin root atau administrator. File konfigurasi ini digunakan untuk mendeklarasikan kredensyal IAM dan AWS Wilayah target yang akan digunakan. CodeDeploy File harus ditambahkan ke lokasi tertentu pada instance lokal. File

harus menyertakan ARN sesi sementara IAM, ID kunci rahasia dan kunci akses rahasia, dan Wilayah target. AWS

Untuk menambahkan file konfigurasi

1. Membuat file bernama `codedeploy.onpremises.yml` (untuk Server Ubuntu atau instans lokal RHEL) atau `conf.onpremises.yml` (untuk instance lokal Windows Server) di lokasi berikut pada instans lokal:

   - Untuk Server Ubuntu: `/etc/codedeploy-agent/conf`
   - Untuk Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Gunakan editor teks untuk menambahkan informasi berikut ke `codedeploy.onpremises.yml` file yang baru dibuat (Linux) atau `conf.onpremises.yml` file (Windows):

   ```
   ---
   iam_session_arn: iam-session-arn
   aws_credentials_file: credentials-file
   region: supported-region
   ```

   Di mana:

   - *`iam-session-arn`*adalah ARN sesi IAM yang Anda catat. Langkah 2: Hasilkan kredensyal sementara untuk instance individual menggunakan AWS STS
   - *`credentials-file`*adalah lokasi file kredensyal untuk sesi sementara ARN, sebagaimana disebutkan dalam. Langkah 2: Hasilkan kredensyal sementara untuk instance individual menggunakan AWS STS
   - *`supported-region`*adalah salah satu Wilayah yang CodeDeploy mendukung, sebagaimana tercantum di Wilayah dan titik akhir di Referensi Umum AWS.

Langkah 4: Siapkan instans lokal untuk penerapan CodeDeploy

Instal dan konfigurasikan AWS CLI

Instal dan konfigurasikan instans lokal AWS CLI di tempat. (Ini AWS CLI akan digunakan untuk mengunduh dan menginstal CodeDeploy agen pada instance lokal.)

1. Untuk menginstal instans lokal, ikuti petunjuk di Menyiapkan instans AWS CLI di Panduan AWS Command Line Interface Pengguna. AWS CLI

> **Note**
>
> CodeDeploy perintah untuk bekerja dengan instance lokal menjadi tersedia di versi 1.7.19 dari. AWS CLI Jika Anda memiliki versi yang AWS CLI sudah diinstal, Anda dapat memeriksa versinya dengan meneleponaws --version.

2. Untuk mengonfigurasi instans lokal, ikuti petunjuk dalam Mengonfigurasi AWS CLI di AWS Command Line Interface Panduan Pengguna. AWS CLI

> **Important**
>
> Saat Anda mengonfigurasi AWS CLI (misalnya, dengan memanggil aws configure perintah), pastikan untuk menentukan ID kunci rahasia dan kunci akses rahasia dari pengguna IAM yang memiliki, setidaknya, izin yang dijelaskan dalam. Prasyarat pendaftaran ARN sesi IAM

Mengatur Variabel Lingkungan AWS_REGION (Server Ubuntu dan RHEL Saja)

Jika Anda tidak menjalankan Ubuntu Server atau RHEL pada instans lokal Anda, lewati langkah ini dan langsung ke “Instal CodeDeploy agen.”

Instal CodeDeploy agen di Server Ubuntu atau instans lokal RHEL dan aktifkan instance untuk memperbarui CodeDeploy agen setiap kali versi baru tersedia. Anda melakukan ini dengan menyetel variabel `AWS_REGION` lingkungan pada instance ke pengidentifikasi salah satu Wilayah yang didukung oleh CodeDeploy. Kami menyarankan Anda menetapkan nilai ke Wilayah tempat CodeDeploy aplikasi, grup penyebaran, dan revisi aplikasi Anda berada (misalnya,`us-west-2`). Untuk daftar Wilayah, lihat Wilayah dan titik akhir di. Referensi Umum AWS

Untuk mengatur variabel lingkungan, panggil yang berikut ini dari terminal:

```
export AWS_REGION=supported-region
```

Di *`supported-region`* mana pengenal Wilayah (misalnya,`us-west-2`).

Instal CodeDeploy agen

- Untuk instance lokal Server Ubuntu, ikuti petunjuknyaInstal CodeDeploy agen untuk Server Ubuntu, lalu kembali ke halaman ini.

- Untuk instance lokal RHEL, ikuti petunjuknyaInstal CodeDeploy agen untuk Amazon Linux atau RHEL, lalu kembali ke halaman ini.
- Untuk instance lokal Windows Server, ikuti petunjuknyaInstal CodeDeploy agen untuk Windows Server, lalu kembali ke halaman ini.

Langkah 5: Daftarkan instans lokal dengan CodeDeploy

Petunjuk dalam langkah ini mengasumsikan Anda mendaftarkan instans lokal dari instans lokal itu sendiri. Anda dapat mendaftarkan instans lokal dari perangkat atau instans terpisah yang telah AWS CLI diinstal dan dikonfigurasi.

Gunakan AWS CLI untuk mendaftarkan instance lokal CodeDeploy sehingga dapat digunakan dalam penerapan.

Sebelum Anda dapat menggunakan AWS CLI, Anda akan memerlukan ARN dari kredensyal sesi sementara yang Anda buat. Langkah 3: Tambahkan file konfigurasi ke instans lokal Misalnya, untuk contoh yang Anda identifikasi sebagai`AssetTag12010298EX`:

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Panggil perintah register-on-premises-instance, dengan menentukan:

- Nama yang secara unik mengidentifikasi instance lokal (dengan opsi). `--instance-name`

> **Important**
>
> Untuk membantu mengidentifikasi instans lokal, terutama untuk tujuan debugging, kami sangat menyarankan Anda menentukan nama yang memetakan ke beberapa karakteristik unik instans lokal (misalnya, nama sesi kredenal STS dan nomor seri atau pengenal aset internal, jika berlaku). Jika Anda menentukan alamat MAC sebagai nama, ketahuilah bahwa alamat MAC berisi karakter yang CodeDeploy tidak mengizinkan, seperti titik dua (:). Untuk daftar karakter yang diizinkan, lihatCodeDeploy kuota.

- ARN sesi IAM yang Anda atur untuk mengautentikasi beberapa instans lokal. Langkah 1: Buat peran IAM yang akan diasumsikan oleh instans lokal

Contoh:

```
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
```

Di mana:

- *name-of-instance*adalah nama yang Anda gunakan untuk mengidentifikasi instance lokal, sepertiAssetTag12010298EX.
- *account-id*adalah ID akun 12 digit untuk organisasi Anda, seperti111222333444.
- *role-to-assume*adalah nama peran IAM yang Anda buat untuk instance, sepertiCodeDeployInstanceRole.
- *session-name*adalah nama peran sesi yang Anda tentukanLangkah 2: Hasilkan kredensyal sementara untuk instance individual menggunakan AWS STS.

Langkah 6: Tandai instans lokal

Anda dapat menggunakan konsol AWS CLI atau CodeDeploy konsol untuk menandai instans lokal. (CodeDeploymenggunakan tag instans lokal untuk mengidentifikasi target penerapan selama penerapan.)

Untuk menandai instance lokal (CLI)

- Panggil perintah add-tags-to-on-premises-instance, dengan menentukan:
  - Nama yang secara unik mengidentifikasi instance lokal (dengan opsi). `--instance-names`
  - Nama kunci tag instans lokal dan nilai tag yang ingin Anda gunakan (dengan `--tags` opsi). Anda harus menentukan nama dan nilai. CodeDeploy tidak mengizinkan tag instance lokal yang hanya memiliki nilai.

    Contoh:

    ```
    aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
     --tags Key=Name,Value=CodeDeployDemo-OnPrem
    ```

Untuk menandai instance lokal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih On-premises instance.
3. Dalam daftar instance lokal, pilih nama instans lokal yang ingin Anda tag.
4. Dalam daftar tag, pilih atau masukkan kunci tag dan nilai tag yang diinginkan. Setelah Anda memasukkan kunci tag dan nilai tag, baris lain muncul. Anda dapat mengulangi ini hingga 10 tag. Untuk menghapus sebuah tag, pilih Hapus.
5. Setelah Anda menambahkan tag, pilih Perbarui Tag.

Langkah 7: Menerapkan revisi aplikasi ke instans lokal

Anda sekarang siap untuk menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag.

Anda menerapkan revisi aplikasi ke instans lokal dengan cara yang mirip dengan menerapkan revisi aplikasi ke instans Amazon EC2. Untuk petunjuk, lihat Buat penerapan dengan CodeDeploy. Instruksi ini mencakup tautan ke prasyarat, termasuk membuat aplikasi, membuat grup penyebaran, dan menyiapkan revisi aplikasi. Jika Anda memerlukan contoh revisi aplikasi sederhana untuk diterapkan, Anda dapat membuat yang dijelaskan Langkah 2: Buat contoh revisi aplikasi dalam. Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)

> **Important**
>
> Jika Anda menggunakan kembali peran CodeDeploy layanan sebagai bagian dari pembuatan grup penerapan yang menargetkan instance lokal, Anda `Tag:get*` harus menyertakan `Action` bagian pernyataan kebijakan peran layanan. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Langkah 8: Lacak penerapan ke instans lokal

Setelah menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag, Anda dapat melacak kemajuan penerapan.

Anda melacak penerapan ke instans lokal dengan cara yang mirip dengan melacak penerapan ke instans Amazon EC2. Untuk petunjuk, lihat Lihat CodeDeploy detail penerapan.

Gunakan perintah register (pengguna IAM ARN) untuk mendaftarkan instance lokal

> **Important**
>
> Mendaftarkan instance menggunakan pengguna IAM tidak disarankan karena menggunakan kredenSIAL statis (permanen) untuk otentikasi. Untuk keamanan yang lebih baik, sebaiknya daftarkan instance menggunakan kredentif sementara untuk otentikasi. Untuk informasi selengkapnya, lihat Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal.

> **Important**
>
> Pastikan Anda memiliki rencana untuk memutar kunci akses pengguna IAM (kredenSIAL permanen). Untuk informasi selengkapnya, lihat Memutar tombol akses.

Bagian ini menjelaskan cara mengonfigurasi instans lokal dan mendaftarkan serta men CodeDeploy andainya dengan sedikit usaha. Per register intah ini paling berguna saat Anda bekerja dengan armada instans lokal tunggal atau kecil. Anda dapat menggunakan register perintah hanya ketika Anda menggunakan ARN pengguna IAM untuk mengotentikasi sebuah instance. Anda tidak dapat menggunakan register perintah dengan sesi IAM ARN untuk otentikasi.

Ketika Anda menggunakan register perintah, Anda dapat CodeDeploy melakukan hal berikut:

- Buat pengguna IAM AWS Identity and Access Management untuk instance lokal, jika Anda tidak menentukannya dengan perintah.
- Simpan kredentif pengguna IAM ke file konfigurasi instans lokal.
- Daftarkan instance lokal dengan CodeDeploy.
- Tambahkan tag ke instance lokal, jika Anda menentukannya sebagai bagian dari perintah.

> **Note**
>
> Per intah register-on-premises-instance adalah alternatif dari perintah register. https://docs.aws.amazon.com/cli/latest/reference/deploy/register.html Anda menggunakan register-on-premises-instance perintah jika ingin mengonfigurasi instans lokal dan mendaftarkan serta menandainya dengan CodeDeploy sebagian besar sendiri. Per register-on-premises-instance intah ini juga memberi Anda pilihan untuk menggunakan ARN sesi IAM untuk mendaftarkan instance alih-alih ARN pengguna IAM. Pendekatan ini memberikan keuntungan besar jika Anda memiliki armada instans lokal yang besar. Secara khusus, Anda dapat menggunakan ARN sesi IAM tunggal untuk mengotentikasi beberapa instance alih-alih harus membuat pengguna IAM untuk setiap instans lokal satu per satu. Untuk informasi selengkapnya, lihat Gunakan perintah register-on-premises-instance (ARN pengguna IAM) untuk mendaftarkan instance lokal dan Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal.

Topik

- Langkah 1: Instal dan konfigurasikan AWS CLI pada instans lokal
- Langkah 2: Panggil perintah register
- Langkah 3: Panggil perintah install
- Langkah 4: Menyebarkan revisi aplikasi ke instans lokal
- Langkah 5: Melacak penerapan ke instans lokal

Langkah 1: Instal dan konfigurasikan AWS CLI pada instans lokal

1. Instal instans AWS CLI di lokasi. Ikuti petunjuk dalam Menyiapkan dengan AWS CLI di Panduan AWS Command Line Interface Pengguna.

   > **Note**
   >
   > CodeDeploy perintah untuk bekerja dengan instans lokal tersedia di AWS CLI versi 1.7.19 dan yang lebih baru. Jika Anda AWS CLI sudah menginstal, hubungi aws --version untuk memeriksa versinya.

2. Konfigurasikan instans AWS CLI di lokasi. Ikuti petunjuk dalam Meng onfigurasi AWS CLI dalam Panduan AWS Command Line Interface Pengguna.

Important

Saat Anda mengonfigurasi AWS CLI (misalnya, dengan memanggil aws configure perintah), pastikan untuk menentukan ID kunci rahasia dan kunci akses rahasia pengguna IAM yang memiliki, minimal, izin AWS akses berikut selain izin yang ditentukan dalamPrasyarat untuk mengonfigurasi instans lokal. Hal ini memungkinkan untuk mengunduh dan menginstal CodeDeploy agen pada instans lokal. Izin akses mungkin terlihat mirip dengan ini:

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**
>
> Jika Anda melihat kesalahan akses ditolak saat mencoba mengakses salah satu bucket Amazon S3 yang ditampilkan sebelumnya, coba hilangkan `/*` bagian ARN sumber daya bucket, misalnya,`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/*"
      ]
    }
  ]
}
```

Langkah 2: Panggil perintah register

Untuk langkah ini, kami menganggap Anda mendaftarkan instance lokal dari instans lokal itu sendiri. Anda juga dapat mendaftarkan instance lokal dari perangkat atau instans terpisah yang telah AWS CLI diinstal dan dikonfigurasi seperti yang dijelaskan pada langkah sebelumnya.

Gunakan AWS CLI untuk memanggil perintah register, menentukan:

- Nama yang secara unik mengidentifikasi instance lokal ke CodeDeploy (dengan `--instance-name` opsi).

  > **Important**
  >
  > Untuk membantu mengidentifikasi instance lokal nanti, terutama untuk tujuan debugging, kami sangat menyarankan Anda menggunakan nama yang dipetakan ke beberapa karakteristik unik dari instance lokal (misalnya, nomor seri atau beberapa pengenal aset internal yang unik, jika berlaku). Jika Anda menentukan alamat MAC untuk nama, ketahuilah bahwa alamat MAC berisi karakter yang CodeDeploy tidak memungkinkan, seperti titik dua (`:`). Untuk daftar karakter yang diizinkan, lihatCodeDeploy kuota.

- Secara opsional, ARN dari pengguna IAM yang ada yang ingin Anda kaitkan dengan instance lokal ini (dengan `--iam-user-arn` opsi). Untuk mendapatkan ARN pengguna IAM, panggil perintah get-user, atau pilih nama pengguna IAM di bagian U sers pada konsol IAM dan kemudian temukan nilai ARN Pengguna di bagian Ringkasan. Jika opsi ini tidak ditentukan, CodeDeploy akan membuat pengguna IAM atas nama Anda di AWS akun Anda dan menghubungkannya dengan instance lokal.

> **Important**
>
> Jika Anda menentukan `--iam-user-arn` opsi, Anda juga harus membuat file konfigurasi instans lokal secara manual, seperti yang dijelaskan dalamLangkah 4: Menambahkan file konfigurasi ke instance lokal.
> Anda hanya dapat mengaitkan satu pengguna IAM dengan satu instans lokal saja. Mencoba mengaitkan satu pengguna IAM dengan beberapa instans lokal dapat mengakibatkan kesalahan, penerapan yang gagal ke instans lokal tersebut, atau penerapan ke instans lokal yang macet dalam status tertunda abadi.

- Secara opsional, kumpulan tag instans lokal (dengan `--tags` opsi) yang CodeDeploy akan digunakan untuk mengidentifikasi kumpulan instans Amazon EC2 yang akan digunakan untuk menerapkan. Tentukan setiap tag dengan `Key=`*`tag-key`*`,Value=`*`tag-value`* (misalnya,`Key=Name,Value=Beta Key=Name,Value=WestRegion`). Jika opsi ini tidak ditentukan, tidak ada tag yang akan terdaftar. Untuk mendaftarkan tag nanti, panggil perintah add-tags- to-on-premises-instance.
- Secara opsional, Wil AWS ayah tempat instance lokal akan didaftarkan CodeDeploy (dengan `--region` opsi). Ini harus menjadi salah satu Wilayah yang didukung yang terdaftar di Wilayah dan titik akhir di Referensi Umum AWS (misalnya,`us-west-2`). Jika opsi ini tidak ditentukan, Wil AWS ayah default yang terkait dengan pengguna IAM yang memanggil akan digunakan.

Contoh:

```
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
```

Per register intah melakukan hal berikut:

1. Jika tidak ada pengguna IAM yang ada yang ditentukan, membuat pengguna IAM, melampirkan izin yang diperlukan untuk itu, dan menghasilkan kunci rahasia dan kunci akses rahasia yang sesuai. Instans lokal akan menggunakan pengguna IAM ini beserta izin serta kredensialnya untuk mengotentikasi dan berinteraksi dengannya. CodeDeploy
2. Mendaftarkan instance lokal dengan CodeDeploy.
3. Jika ditentukan, ka CodeDeploy itkan dalam tag yang ditentukan dengan `--tags` opsi dengan nama instans lokal terdaftar.

4. Jika pengguna IAM dibuat, juga membuat file konfigurasi yang diperlukan di direktori yang sama dari mana register perintah dipanggil.

Jika perintah ini menemukan kesalahan, pesan kesalahan muncul, menjelaskan bagaimana Anda dapat menyelesaikan langkah-langkah yang tersisa secara manual. Jika tidak, pesan sukses muncul, menjelaskan cara memanggil install perintah seperti yang tercantum pada langkah berikutnya.

Langkah 3: Panggil perintah install

Dari instance lokal, gunakan AWS CLI untuk memanggil perintah install, tentukan:

- Jalur ke file konfigurasi (dengan `--config-file` opsi).
- Secara opsional, apakah akan mengganti file konfigurasi yang sudah ada di instance lokal (dengan `--override-config` opsi). Jika tidak ditentukan, file konfigurasi yang ada tidak akan diganti.
- Secara opsional, Wil AWS ayah tempat instance lokal akan didaftarkan CodeDeploy (dengan `--region` opsi). Ini harus menjadi salah satu Wilayah yang didukung yang terdaftar di Wilayah dan titik akhir di Referensi Umum AWS (misalnya,`us-west-2`). Jika opsi ini tidak ditentukan, Wil AWS ayah default yang terkait dengan pengguna IAM yang memanggil akan digunakan.
- Secara opsional, lokasi khusus untuk menginstal CodeDeploy agen (dengan `--agent-installer` opsi). Opsi ini berguna untuk menginstal versi kustom CodeDeploy agen yang CodeDeploy tidak mendukung secara resmi (seperti versi kustom berdasarkan repositori CodeDeploy agen di GitHub). Nilai harus berupa jalur ke bucket Amazon S3 yang berisi salah satu:
  - Skrip instalasi CodeDeploy agen (untuk Linux- atau sistem Unix-based operasi, mirip dengan file instal di repositori CodeDeploy agen di GitHub).
  - File paket penginstal CodeDeploy agen (.msi) (untuk sistem Windows-based operasi).

  Jika opsi ini tidak ditentukan, CodeDeploy akan melakukan upaya terbaik untuk menginstal dari lokasinya sendiri versi CodeDeploy agen yang didukung secara resmi yang kompatibel dengan sistem operasi pada instans lokal.

Contoh:

```
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
```

Per install intah melakukan hal berikut:

1. Memeriksa apakah instans lokal adalah instans Amazon EC2. Jika ya, pesan kesalahan muncul.
2. Menyalin file konfigurasi instans lokal dari lokasi yang ditentukan pada instans ke lokasi tempat CodeDeploy agen mengharapkan untuk menemukannya, asalkan file tersebut belum berada di lokasi tersebut.

   Untuk Ubuntu Server dan Red Hat Enterprise Linux (RHEL)), ini adalah`/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`.

   Untuk Windows Server, ini adalah`C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.

   Jika `--override-config` opsi ditentukan, membuat atau mengganti file.
3. Menginstal CodeDeploy agen pada instans lokal dan kemudian memulainya.

Langkah 4: Menyebarkan revisi aplikasi ke instans lokal

Anda sekarang siap untuk menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag.

Anda menyebarkan revisi aplikasi ke instans lokal dengan cara yang mirip dengan menerapkan revisi aplikasi ke instans Amazon EC2. Untuk petunjuk, lihat Buat penerapan dengan CodeDeploy. Instruksi ini menautkan ke prasyarat, termasuk membuat aplikasi, membuat grup penerapan, dan menyiapkan revisi aplikasi. Jika Anda memerlukan revisi aplikasi sampel sederhana untuk diterapkan, Anda dapat membuat yang dijelaskan Langkah 2: Buat contoh revisi aplikasi diTutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux).

> **Important**
>
> Jika Anda menggunakan kembali peran CodeDeploy layanan yang ada sebagai bagian dari pembuatan grup penerapan yang menargetkan instans lokal, Anda `Tag:get*` harus menyer `Action` takan bagian pernyataan kebijakan peran layanan tersebut. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Langkah 5: Melacak penerapan ke instans lokal

Setelah menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag, Anda dapat melacak kemajuan penerapan.

Anda melacak penerapan ke instans lokal dengan cara yang mirip dengan melacak penerapan ke instans Amazon EC2. Untuk petunjuk, lihat Lihat CodeDeploy detail penerapan.

Untuk opsi lainnya, lihatMengelola operasi instans lokal di CodeDeploy.

Gunakan perintah register-on-premises-instance (ARN pengguna IAM) untuk mendaftarkan instance lokal

> **Important**
>
> Mendaftarkan instance menggunakan pengguna IAM tidak disarankan karena menggunakan kredenSIAL statis (permanen) untuk otentikasi. Untuk keamanan yang lebih baik, sebaiknya daftarkan instance menggunakan kredentif sementara untuk otentikasi. Untuk informasi selengkapnya, lihat Gunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal.

> **Important**
>
> Pastikan Anda memiliki rencana untuk memutar kunci akses pengguna IAM (kredenSIAL permanen). Untuk informasi selengkapnya, lihat Memutar tombol akses.

Ikuti petunjuk ini untuk mengonfigurasi instans lokal dan mendaftarkan dan menandainya dengan CodeDeploy sebagian besar Anda sendiri, menggunakan kredentif pengguna IAM statis untuk otentikasi.

Topik

- Langkah 1: Membuat pengguna IAM untuk instans lokal
- Langkah 2: Tetapkan izin ke pengguna IAM
- Langkah 3: Dapatkan kredenSIAL pengguna IAM
- Langkah 4: Menambahkan file konfigurasi ke instance lokal
- Langkah 5: Instal dan konfigurasikan AWS CLI
- Langkah 6: Atur variabel lingkungan AWS_REGION (Ubuntu Server dan RHEL saja)
- Langkah 7: Instal CodeDeploy agen
- Langkah 8: Mendaftarkan instans lokal dengan CodeDeploy

- Langkah 9: Tandai instance lokal
- Langkah 10: Menyebarkan revisi aplikasi ke instans lokal
- Langkah 11: Melacak penerapan ke instans lokal

Langkah 1: Membuat pengguna IAM untuk instans lokal

Buat pengguna IAM yang akan digunakan instans lokal untuk mengotentikasi dan berinteraksi dengannya. CodeDeploy

> **Important**
>
> Anda harus membuat pengguna IAM terpisah untuk setiap instans lokal yang berpartisipasi. Jika Anda mencoba menggunakan kembali pengguna IAM individual untuk beberapa instans lokal, Anda mungkin tidak berhasil mendaftarkan atau menandai instans lokal tersebut dengan. CodeDeploy Penerapan ke instans lokal tersebut mungkin macet dalam status tertunda abadi atau gagal sama sekali.

Kami menyarankan Anda menetapkan nama pengguna IAM yang mengidentifikasi tujuannya, seperti CodeDeployUser-OnPrem.

Anda dapat menggunakan AWS CLI atau konsol IAM untuk membuat pengguna IAM. Untuk selengkapnya, lihat Membuat pengguna IAM di AWS akun Anda.

> **Important**
>
> Apakah Anda menggunakan AWS CLI atau konsol IAM untuk membuat pengguna IAM baru, catat ARN pengguna yang disediakan untuk pengguna. Anda akan membutuhkan informasi ini nanti di Langkah 4: Menambahkan file konfigurasi ke instance lokal danLangkah 8: Mendaftarkan instans lokal dengan CodeDeploy.

Langkah 2: Tetapkan izin ke pengguna IAM

Jika instans lokal Anda akan menerapkan revisi aplikasi dari bucket Amazon S3, Anda harus menetapkan izin kepada pengguna IAM untuk berinteraksi dengan bucket tersebut. Anda dapat menggunakan AWS CLI atau konsol IAM untuk menetapkan izin.

> **Note**
>
> Jika Anda akan menerapkan revisi aplikasi hanya dari GitHub repositori, lewati langkah ini dan langsung ke. Langkah 3: Dapatkan kredenSIAL pengguna IAM (Anda masih memerlukan informasi tentang pengguna IAM yang Anda buat. Langkah 1: Membuat pengguna IAM untuk instans lokal Ini akan digunakan pada langkah selanjutnya.)

Untuk menetapkan izin (CLI)

1. Buat file dengan konten kebijakan berikut di instans atau perangkat Amazon EC2 yang Anda gunakan untuk memanggil. AWS CLI Beri nama file seperti**`CodeDeploy-OnPrem-Permissions.json`**, lalu simpan file tersebut.

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

> **Note**
>
> Sebaiknya batasi kebijakan ini hanya untuk bucket Amazon S3 yang perlu diakses instans lokal Anda. Jika Anda membatasi kebijakan ini, pastikan juga memberikan akses ke bucket Amazon S3 yang berisi AWS CodeDeploy agen. Jika tidak, kesalahan mungkin terjadi setiap kali CodeDeploy agen diinstal atau diperbarui pada instans lokal terkait.
> Contoh:
>
> ```
> {
>   "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. Panggil perintah put-user-policy, tentukan nama pengguna IAM (dengan `--user-name` opsi), nama untuk kebijakan (dengan `--policy-name` opsi), dan jalur ke dokumen kebijakan

yang baru dibuat (dengan opsi). `--policy-document` Misalnya, dengan asumsi bahwa **`CodeDeploy-OnPrem-Permissions.json`** file berada di direktori (folder) yang sama dari mana Anda memanggil perintah ini:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-
OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

Untuk menetapkan izin (konsol)

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pada panel navigasi, pilih Kebijakan, lalu pilih Buat kebijakan. (Jika tombol Memulai muncul, pilihlah, lalu pilih Buat Kebijakan.)
3. Di sebelah Buat Kebijakan Anda Sendiri, pilih Pilih.
4. Di kotak Nama Kebijakan, ketik nama untuk kebijakan ini (misalnya,**`CodeDeploy-OnPrem-Permissions`**).
5. Di kotak Dokumen Kebijakan, ketik atau tempel ekspresi izin berikut, yang AWS CodeDeploy memungkinkan penerapan revisi aplikasi dari bucket Amazon S3 apa pun yang ditentukan dalam kebijakan ke instance lokal atas nama pengguna IAM:

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

6. Pilih Buat Kebijakan.
7. Di panel navigasi, pilih Users (Pengguna).
8. Dalam daftar pengguna, telusuri dan pilih nama pengguna IAM yang Anda buat. Langkah 1: Membuat pengguna IAM untuk instans lokal
9. Pada tab Izin, di Kebijakan Ter kelola, pilih Lampirkan Kebijakan.
10. Pilih kebijakan bernama**`CodeDeploy-OnPrem-Permissions`**, lalu pilih Lamp irkan Kebijakan.

Langkah 3: Dapatkan kredenSIAL pengguna IAM

Dapatkan ID kunci rahasia dan kunci akses rahasia untuk pengguna IAM. Anda akan membutuhkannyaLangkah 4: Menambahkan file konfigurasi ke instance lokal. Anda dapat menggunakan AWS CLI atau konsol IAM untuk mendapatkan ID kunci rahasia dan kunci akses rahasia.

> **Note**
>
> Jika Anda sudah memiliki ID kunci rahasia dan kunci akses rahasia, lewati langkah ini dan langsung keLangkah 4: Menambahkan file konfigurasi ke instance lokal.
> Pengguna memerlukan akses terprogram jika mereka ingin berinteraksi dengan AWS di luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.
> Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredenSIAL konsol sebagai kredenSIAL sementara untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDK, atau AWS API.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		• Untuk AWS SDK, lihat Login untuk pengembangan AWS lokal di Panduan Refer AWS ensi SDK dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredenSIAL sementara untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDK, atau AWS API.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Meng onfigurasi AWS CLI untuk digunakan AWS IAM Identity Center di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDK, alat, dan AWS API, lihat otentikasi Pusat Identitas IAM di Panduan Refer AWS ensi SDK dan Alat.
IAM	Gunakan kredenSIAL sementara untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDK, atau AWS API.	Mengikuti petunjuk dalam Menggunakan kredenSIAL sementara dengan AWS sumber daya di Panduan Pengguna IAM.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Tidak direkomendasikan) Gunakan kredentif jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDK, atau AWS API.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Meng autentikasi menggunakan kredentif pengguna IAM di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDK dan alat, lihat Meng autentikasi menggunakan kredentif jangka panjang di Panduan Refer AWS ensi SDK dan Alat. • Untuk AWS API, lihat M engelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Untuk mendapatkan kredenSIAL (CLI)

1. Panggil perintah list-access-keys, tentukan nama pengguna IAM (dengan `--user-name` opsi) dan menanyakan hanya ID kunci akses (dengan opsi dan). `--query` `--output` Contoh:

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
 "AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Jika tidak ada tombol yang muncul di output atau informasi tentang hanya satu tombol muncul di output, panggil perintah create-access-key, tentukan nama pengguna IAM (dengan opsi): `--user-name`

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

Dalam output panggilan ke create-access-key perintah, perhatikan nilai `SecretAccessKey` bidang `AccessKeyId` dan. Anda akan membutuhkan informasi iniLangkah 4: Menambahkan file konfigurasi ke instance lokal.

> **Important**
>
> Ini akan menjadi satu-satunya waktu Anda akan memiliki akses ke kunci akses rahasia ini. Jika Anda lupa atau kehilangan akses ke kunci akses rahasia ini, Anda harus membuat yang baru dengan mengikuti langkah-langkah diLangkah 3: Dapatkan kredenSIAL pengguna IAM.

3. Jika dua kunci akses sudah terdaftar, Anda harus menghapus salah satunya dengan memanggil perintah delete-access-key, menentukan nama pengguna IAM (dengan `--user-name` opsi), dan ID kunci akses untuk dihapus (dengan opsi). `--access-key-id` Kemudian panggil create-access-key perintah, seperti yang dijelaskan sebelumnya pada langkah ini. Berikut contoh memanggil delete-access-key perintah:

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-key-ID
```

> **Important**
>
> Jika Anda memanggil delete-access-key perintah untuk menghapus salah satu kunci akses ini, dan instance lokal sudah menggunakan kunci akses ini seperti yang dijelaskan dalamLangkah 4: Menambahkan file konfigurasi ke instance lokal, Anda harus mengikuti instruksi Langkah 4: Menambahkan file konfigurasi ke instance lokal lagi untuk menentukan ID kunci akses yang berbeda dan kunci akses rahasia yang terkait dengan pengguna IAM ini. Jika tidak, penerapan apa pun ke instans lokal tersebut mungkin macet dalam status tertunda abadi atau gagal sama sekali.

Untuk mendapatkan kredenSIAL (konsol)

1. a. Buka konsol IAM di https://console.aws.amazon.com/iam/.

b. Jika daftar pengguna tidak ditampilkan, di panel navigasi, pilih Pengguna.

c. Dalam daftar pengguna, telusuri dan pilih nama pengguna IAM yang Anda buat. Langkah 1: Membuat pengguna IAM untuk instans lokal

2. Pada tab Kredensibilitas keamanan, jika tidak ada kunci atau hanya satu kunci yang terdaftar, pilih Buat kunci akses.

   Jika dua kunci akses terdaftar, Anda harus menghapus salah satunya. Pilih H apus di sebelah salah satu tombol akses, lalu pilih Buat kunci akses.

   > **Important**
   >
   > Jika Anda memilih H apus di sebelah salah satu kunci akses ini, dan instans lokal sudah menggunakan kunci akses ini seperti yang dijelaskan dalamLangkah 4: Menambahkan file konfigurasi ke instance lokal, Anda harus mengikuti instruksi Langkah 4: Menambahkan file konfigurasi ke instance lokal lagi untuk menentukan ID kunci akses dan kunci akses rahasia yang berbeda yang terkait dengan pengguna IAM ini. Jika tidak, penerapan ke instans lokal tersebut mungkin macet dalam status tertunda abadi atau gagal sama sekali.

3. Pilih Tampilkan dan catat ID kunci akses dan kunci akses rahasia. Anda akan membutuhkan informasi ini untuk langkah selanjutnya. Atau, Anda dapat memilih Unduh file.csv untuk menyimpan salinan ID kunci akses dan kunci akses rahasia.

   > **Important**
   >
   > Kecuali Anda membuat catatan atau mengunduh kredensialnya, ini akan menjadi satu-satunya waktu Anda akan memiliki akses ke kunci akses rahasia ini. Jika Anda lupa atau kehilangan akses ke kunci akses rahasia ini, Anda harus membuat yang baru dengan mengikuti langkah-langkah diLangkah 3: Dapatkan kredenSIAL pengguna IAM.

4. Pilih Tut up untuk kembali ke *IAM User Name* halaman Pengguna >.

Langkah 4: Menambahkan file konfigurasi ke instance lokal

Tambahkan file konfigurasi ke instans lokal, menggunakan izin root atau administrator. File konfigurasi ini akan digunakan untuk mendeklarasikan kredenSIAL pengguna IAM dan Wil AWS ayah target yang akan digunakan CodeDeploy. File harus ditambahkan ke lokasi tertentu pada instance

lokal. File harus menyertakan ARN pengguna IAM, ID kunci rahasia, kunci akses rahasia, dan Wil AWS ayah target. File harus mengikuti format tertentu.

1. Buat file bernama `codedeploy.onpremises.yml` (untuk instans lokal Ubuntu Server atau RHEL) atau `conf.onpremises.yml` (untuk instans lokal Windows Server) di lokasi berikut pada instance lokal:

   - Untuk Ubuntu Server: `/etc/codedeploy-agent/conf`
   - Untuk Windows Server: `C:\ProgramData\Amazon\CodeDeploy`

2. Gunakan editor teks untuk menambahkan informasi berikut ke `conf.onpremises.yml` file `codedeploy.onpremises.yml` atau yang baru dibuat:

   ```
   ---
   aws_access_key_id: secret-key-id
   aws_secret_access_key: secret-access-key
   iam_user_arn: iam-user-arn
   region: supported-region
   ```

   Di mana:

   - *`secret-key-id`*adalah ID kunci rahasia pengguna IAM yang sesuai yang Anda catat di Langkah 1: Membuat pengguna IAM untuk instans lokal atauLangkah 3: Dapatkan kredenSIAL pengguna IAM.
   - *`secret-access-key`*adalah kunci akses rahasia pengguna IAM yang sesuai yang Anda catat di Langkah 1: Membuat pengguna IAM untuk instans lokal atauLangkah 3: Dapatkan kredenSIAL pengguna IAM.
   - *`iam-user-arn`*adalah ARN pengguna IAM yang Anda catat sebelumnya. Langkah 1: Membuat pengguna IAM untuk instans lokal
   - *`supported-region`*adalah pengidentifikasi Wilayah yang didukung CodeDeploy oleh lokasi CodeDeploy aplikasi, grup penerapan, dan revisi aplikasi Anda (misalnya,`us-west-2`). Untuk daftar Wilayah, lihat Wil ayah dan titik akhir di Referensi Umum AWS.

   Important

   Jika Anda memilih H apus di sebelah salah satu kunci akses diLangkah 3: Dapatkan kredenSIAL pengguna IAM, dan instans lokal Anda sudah menggunakan ID kunci akses

terkait dan kunci akses rahasia, Anda harus mengikuti petunjuk di dalam Langkah 4: Menambahkan file konfigurasi ke instance lokal untuk menentukan ID kunci akses dan kunci akses rahasia yang berbeda yang terkait dengan pengguna IAM ini. Jika tidak, penerapan apa pun ke instans lokal Anda mungkin macet dalam status tertunda abadi atau gagal sama sekali.

Langkah 5: Instal dan konfigurasikan AWS CLI

Instal dan konfigurasikan AWS CLI pada instans lokal. (AWS CLI Ini akan digunakan Langkah 7: Instal CodeDeploy agen untuk mengunduh dan menginstal CodeDeploy agen pada instance lokal.)

1. Untuk menginstal instans AWS CLI di lokasi, ikuti petunjuk dalam Menyiapkan dengan AWS CLI di Panduan AWS Command Line Interface Pengguna.

   Note

   CodeDeploy perintah untuk bekerja dengan instans lokal tersedia di versi 1.7.19 dari. AWS CLI Jika Anda memiliki versi yang AWS CLI sudah diinstal, Anda dapat memeriksa versinya dengan meneleponaws --version.

2. Untuk mengonfigurasi AWS CLI instans lokal, ikuti petunjuk di Konfigurasi AWS CLI di Panduan AWS Command Line Interface Pengguna.

   Important

   Saat Anda mengkonfigurasi AWS CLI (misalnya, dengan memanggil aws configure perintah), pastikan untuk menentukan ID kunci rahasia dan kunci akses rahasia pengguna IAM yang memiliki, minimal, izin AWS akses berikut selain izin akses yang ditentukan dalamPrasyarat untuk mengonfigurasi instans lokal. Hal ini memungkinkan Anda untuk mengunduh dan menginstal CodeDeploy agen pada instans lokal:

   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" : [
        "*"
      ]
    }
```

```
    ]
}
```

Izin akses ini dapat ditetapkan ke pengguna IAM yang Anda buat di Langkah 1: Membuat pengguna IAM untuk instans lokal atau pengguna IAM yang berbeda. Untuk menetapkan izin ini ke pengguna IAM, ikuti petunjuk diLangkah 1: Membuat pengguna IAM untuk instans lokal, menggunakan izin akses ini alih-alih yang ada di langkah itu.

Langkah 6: Atur variabel lingkungan AWS_REGION (Ubuntu Server dan RHEL saja)

Jika Anda tidak menjalankan Ubuntu Server atau RHEL di instans lokal Anda, lewati langkah ini dan langsung ke. Langkah 7: Instal CodeDeploy agen

Instal CodeDeploy agen di Ubuntu Server atau instans lokal RHEL dan aktifkan instance untuk memperbarui CodeDeploy agen setiap kali versi baru tersedia. Anda melakukan ini dengan menyet `AWS_REGION` el variabel lingkungan pada instance ke pengenal salah satu Wilayah yang didukung oleh CodeDeploy. Sebaiknya tetapkan nilai ke Wilayah tempat CodeDeploy aplikasi, grup penerapan, dan revisi aplikasi Anda berada (misalnya,`us-west-2`). Untuk daftar Wilayah, lihat Wil ayah dan titik akhir di Referensi Umum AWS.

Untuk mengatur variabel lingkungan, panggil yang berikut dari terminal:

```
export AWS_REGION=supported-region
```

Dim *`supported-region`* ana pengidentifikasi Wilayah (misalnya,`us-west-2`).

Langkah 7: Instal CodeDeploy agen

Instal CodeDeploy agen pada instans lokal:

- Untuk instance Ubuntu Server lokal, ikuti petunjuk diInstal CodeDeploy agen untuk Server Ubuntu, lalu kembali ke halaman ini.
- Untuk instance lokal RHEL, ikuti petunjuk diInstal CodeDeploy agen untuk Amazon Linux atau RHEL, lalu kembali ke halaman ini.
- Untuk instance lokal Windows Server, ikuti petunjuk diInstal CodeDeploy agen untuk Windows Server, lalu kembali ke halaman ini.

Langkah 8: Mendaftarkan instans lokal dengan CodeDeploy

Petunjuk dalam langkah ini mengasumsikan Anda mendaftarkan instance lokal dari instans lokal itu sendiri. Anda dapat mendaftarkan instance lokal dari perangkat atau instans terpisah yang telah AWS CLI diinstal dan dikonfigurasi, seperti yang dijelaskan dalamLangkah 5: Instal dan konfigurasikan AWS CLI.

Gunakan AWS CLI untuk mendaftarkan instance lokal CodeDeploy sehingga dapat digunakan dalam penerapan.

1. Sebelum Anda dapat menggunakan AWS CLI, Anda memerlukan ARN pengguna dari pengguna IAM yang Anda buat. Langkah 1: Membuat pengguna IAM untuk instans lokal Jika Anda belum memiliki ARN pengguna, panggil perintah get-user, tentukan nama pengguna IAM (dengan `--user-name` opsi) dan menanyakan hanya ARN pengguna (dengan opsi dan): `--query` `--output`

   ```
   aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
   ```

2. Panggil perintah register-on-premises- instance, tentukan:

   - Nama yang secara unik mengidentifikasi instance lokal (dengan `--instance-name` opsi).

     > **Important**
     >
     > Untuk membantu mengidentifikasi instance lokal, terutama untuk tujuan debugging, kami sangat menyarankan Anda menentukan nama yang dipetakan ke beberapa karakteristik unik dari instance lokal (misalnya, nomor seri atau pengenal aset internal, jika berlaku). Jika Anda menentukan alamat MAC sebagai nama, ketahuilah bahwa alamat MAC berisi karakter yang CodeDeploy tidak memungkinkan, seperti titik dua (`:`). Untuk daftar karakter yang diizinkan, lihatCodeDeploy kuota.

   - ARN pengguna dari pengguna IAM yang Anda buat Langkah 1: Membuat pengguna IAM untuk instans lokal (dengan `--iam-user-arn` opsi).

     Contoh:

     ```
     aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --
     iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
     ```

Langkah 9: Tandai instance lokal

Anda dapat menggunakan salah satu AWS CLI atau CodeDeploy konsol untuk menandai instance lokal. (CodeDeploymenggunakan tag instans lokal untuk mengidentifikasi target penerapan selama penerapan.)

Untuk menandai instance lokal (CLI)

- Panggil perintah add-tags-to-on- premises-instance, tentukan:

  - Nama yang secara unik mengidentifikasi instance lokal (dengan `--instance-names` opsi).
  - Nama kunci tag instans lokal dan nilai tag yang ingin Anda gunakan (dengan `--tags` opsi). Anda harus menentukan nama dan nilai. CodeDeploy tidak mengizinkan tag instans lokal yang hanya memiliki nilai.

    Contoh:

    ```
    aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
     --tags Key=Name,Value=CodeDeployDemo-OnPrem
    ```

Untuk menandai instance lokal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Dari CodeDeploy menu, pilih On-premises instance.
3. Dalam daftar instans lokal, pilih panah di sebelah instance lokal yang ingin Anda tandai.
4. Dalam daftar tag, pilih atau masukkan kunci tag dan nilai tag yang diinginkan. Setelah Anda memasukkan kunci tag dan nilai tag, baris lain muncul. Anda dapat mengulangi ini hingga 10 tag. Untuk menghapus tag, pilih ikon hapus ( ).
5. Setelah Anda menambahkan tag, pilih Per barui Tag.

Langkah 10: Menyebarkan revisi aplikasi ke instans lokal

Anda sekarang siap untuk menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag.

Anda menyebarkan revisi aplikasi ke instans lokal dengan cara yang mirip dengan menerapkan revisi aplikasi ke instans Amazon EC2. Untuk petunjuk, lihat Buat penerapan dengan CodeDeploy. Instruksi ini mencakup tautan ke prasyarat, termasuk membuat aplikasi, membuat grup penerapan, dan menyiapkan revisi aplikasi. Jika Anda memerlukan revisi aplikasi sampel sederhana untuk diterapkan, Anda dapat membuat yang dijelaskan Langkah 2: Buat contoh revisi aplikasi diTutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux).

> **Important**
>
> Jika Anda menggunakan kembali peran CodeDeploy layanan sebagai bagian dari pembuatan grup penerapan yang menargetkan instans lokal, Anda `Tag:get*` harus menyer `Action` takan bagian pernyataan kebijakan peran layanan tersebut. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Langkah 11: Melacak penerapan ke instans lokal

Setelah menerapkan revisi aplikasi ke instans lokal yang terdaftar dan diberi tag, Anda dapat melacak kemajuan penerapan.

Anda melacak penerapan ke instans lokal dengan cara yang mirip dengan melacak penerapan ke instans Amazon EC2. Untuk petunjuk, lihat Lihat CodeDeploy detail penerapan.

Mengelola operasi instans lokal di CodeDeploy

Ikuti petunjuk di bagian ini untuk mengelola operasi pada instans lokal setelah Anda mendaftarkannya CodeDeploy, seperti mendapatkan informasi selengkapnya, menghapus tag dari, dan menghapus instalasi dan membatalkan pendaftaran instans lokal.

Topik

- Mendapatkan informasi tentang satu instans lokal
- Mendapatkan informasi tentang beberapa instans lokal
- Hapus tag instans lokal secara manual dari instans lokal

- Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal
- Secara otomatis membatalkan pendaftaran instans lokal
- Deregister instans lokal secara manual

Mendapatkan informasi tentang satu instans lokal

Anda bisa mendapatkan informasi tentang satu instans lokal dengan mengikuti petunjuk diLihat CodeDeploy detail penerapan. Anda dapat menggunakan AWS CLI atau CodeDeploy konsol untuk mendapatkan informasi selengkapnya tentang satu instans lokal.

Untuk mendapatkan informasi tentang satu instans lokal (CLI)

- Panggil perintah get-on-premises-instance, dengan menentukan nama yang secara unik mengidentifikasi instance lokal (dengan opsi): `--instance-name`

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Untuk mendapatkan informasi tentang satu instans lokal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih On-premises instance.
3. Dalam daftar instance lokal, pilih nama instans lokal untuk melihat detailnya.

Mendapatkan informasi tentang beberapa instans lokal

Anda bisa mendapatkan informasi tentang instans lokal dengan mengikuti petunjuk di. Lihat CodeDeploy detail penerapan Anda dapat menggunakan AWS CLI atau CodeDeploy konsol untuk mendapatkan informasi selengkapnya tentang instans lokal.

Untuk mendapatkan informasi tentang beberapa instans lokal (CLI)

1. Untuk daftar nama instans lokal, panggil perintah list-on-premises-instance, dengan menentukan:

   - Apakah akan mendapatkan informasi tentang semua instans lokal yang terdaftar atau dideregistrasi (dengan `--registration-status` opsi dan `Registered` atau, masing-masing). `Deregistered` Jika Anda menghilangkan ini, maka nama instans lokal yang terdaftar dan dideregistrasi akan dikembalikan.
   - Apakah akan mendapatkan informasi hanya tentang instans lokal yang ditandai dengan tag instans lokal tertentu (dengan opsi). `--tag-filters` Untuk setiap tag instans lokal, tentukan`Key`,`Value`, dan `Type` (yang seharusnya selalu`KEY_AND_VALUE`). Pisahkan beberapa tag instans lokal dengan spasi di antara masing-masing `Key` `Value`, dan `Type` triplet.

   Contoh:

   ```
   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. Untuk informasi lebih detail, panggil perintah batch-get-on-premises-instances, dengan nama instance lokal (dengan opsi): `--instance-names`

   ```
   aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
    AssetTag09920444EX
   ```

Untuk mendapatkan informasi tentang beberapa instans lokal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih On-premises instance.

Informasi tentang instans lokal ditampilkan.

Hapus tag instans lokal secara manual dari instans lokal

Biasanya, Anda menghapus tag instans lokal dari instans lokal saat tag tersebut tidak lagi digunakan, atau Anda ingin menghapus instans lokal dari grup penerapan apa pun yang bergantung pada tag tersebut. Anda dapat menggunakan AWS CLI atau AWS CodeDeploy konsol untuk menghapus tag instans lokal dari instans lokal.

Anda tidak perlu menghapus tag instans lokal dari instans lokal sebelum membatalkan pendaftarannya.

Menghapus tag instans lokal secara manual dari instans lokal tidak membatalkan pendaftaran instance. Itu tidak menghapus instalan CodeDeploy agen dari instance. Itu tidak menghapus file konfigurasi dari instance. Itu tidak menghapus pengguna IAM yang terkait dengan instance.

Untuk membatalkan pendaftaran instans lokal secara otomatis, lihat. Secara otomatis membatalkan pendaftaran instans lokal

Untuk membatalkan pendaftaran instans lokal secara manual, lihat. Deregister instans lokal secara manual

Untuk menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instance lokal, lihat. Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal

Untuk menghapus instalan CodeDeploy agen secara manual dari instans lokal, lihat. Mengelola operasi CodeDeploy agen

Untuk menghapus pengguna IAM terkait secara manual, lihat Menghapus pengguna IAM dari akun Anda. AWS

Untuk menghapus tag instans lokal dari instans lokal (CLI)

- Panggil instans hapus tag-dari-pada-premis-, dengan menentukan:

  - Nama-nama yang mengidentifikasi instans lokal secara unik (dengan opsi). `--instance-names`
  - Nama dan nilai tag yang ingin Anda hapus (dengan `--tags` opsi).

Contoh:

```
aws deploy remove-tags-from-on-premises-instances --instance-names
 AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Untuk menghapus tag instans lokal dari instans lokal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih On-premises instance.
3. Dalam daftar instance lokal, pilih nama instance lokal tempat Anda ingin menghapus tag.
4. Di Tag, pilih Hapus di samping setiap tag yang ingin Anda hapus.
5. Setelah Anda menghapus tag, pilih Perbarui tag.

Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal

Biasanya, Anda menghapus instalan CodeDeploy agen dan menghapus file konfigurasi dari instance lokal setelah Anda tidak lagi berencana untuk menerapkannya.

> **Note**
>
> Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal tidak membatalkan pendaftaran instance lokal. Itu tidak memisahkan tag instans lokal yang terkait dengan instans lokal. Itu tidak menghapus pengguna IAM yang terkait dengan instance lokal.
> Untuk membatalkan pendaftaran instans lokal secara otomatis, lihat. Secara otomatis membatalkan pendaftaran instans lokal
> Untuk membatalkan pendaftaran instans lokal secara manual, lihat. Deregister instans lokal secara manual

Untuk memisahkan tag instans lokal terkait secara manual, lihat. Hapus tag instans lokal secara manual dari instans lokal
Untuk menghapus instalan CodeDeploy agen secara manual dari instans lokal, lihat. Mengelola operasi CodeDeploy agen
Untuk menghapus pengguna IAM terkait secara manual, lihat Menghapus pengguna IAM dari akun Anda. AWS

Dari instance lokal, gunakan AWS CLI untuk memanggil perintah uninstall.

Contoh:

```
aws deploy uninstall
```

uninstallPerintah melakukan hal berikut:

1. Menghentikan CodeDeploy agen yang sedang berjalan pada instance lokal.
2. Menghapus instalan CodeDeploy agen dari instans lokal.
3. Menghapus file konfigurasi dari instance lokal. (Untuk Server Ubuntu dan RHEL, ini adalah`/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`. Untuk Windows Server, ini adalah`C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

Secara otomatis membatalkan pendaftaran instans lokal

Biasanya, Anda membatalkan pendaftaran instans lokal setelah Anda tidak lagi berencana untuk menerapkannya. Saat Anda membatalkan pendaftaran instans lokal, meskipun instans lokal mungkin merupakan bagian dari tag instans lokal grup penerapan, instans lokal tidak akan disertakan dalam penerapan apa pun. Anda dapat menggunakan AWS CLI untuk membatalkan pendaftaran instance lokal.

> **Note**
>
> Anda tidak dapat menggunakan CodeDeploy konsol untuk membatalkan pendaftaran instans lokal. Selain itu, membatalkan pendaftaran instans lokal akan menghapus tag instans lokal yang terkait dengan instans lokal. Itu tidak menghapus instalan CodeDeploy agen dari instance lokal. Itu tidak menghapus file konfigurasi instans lokal dari instans lokal.
> Untuk menggunakan CodeDeploy konsol untuk melakukan beberapa (tetapi tidak semua) aktivitas di bagian ini, lihat bagian CodeDeploy konsolDeregister instans lokal secara manual.

Untuk memisahkan tag instans lokal terkait secara manual, lihat. Hapus tag instans lokal secara manual dari instans lokal
Untuk menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instance lokal, lihat. Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal
Untuk menghapus instalan CodeDeploy agen secara manual dari instans lokal, lihat. Mengelola operasi CodeDeploy agen

Gunakan AWS CLI untuk memanggil perintah deregister, dengan menentukan:

- Nama yang secara unik mengidentifikasi instans lokal CodeDeploy (dengan opsi). `--instance-name`
- Secara opsional, apakah akan menghapus pengguna IAM yang terkait dengan instans lokal. Perilaku default adalah menghapus pengguna IAM. Jika Anda tidak ingin menghapus pengguna IAM yang terkait dengan instans lokal, tentukan `--no-delete-iam-user` opsi dalam perintah.
- Secara opsional, AWS Wilayah tempat instans lokal terdaftar CodeDeploy (dengan `--region` opsi). Ini harus menjadi salah satu Wilayah yang didukung yang terdaftar di Wilayah dan titik akhir di Referensi Umum AWS(misalnya,`us-west-2`). Jika opsi ini tidak ditentukan, AWS Wilayah default yang terkait dengan pengguna IAM panggilan akan digunakan.

Contoh yang membatalkan pendaftaran instance dan menghapus pengguna:

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Contoh yang membatalkan pendaftaran instance dan tidak menghapus pengguna:

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region
 us-west-2
```

deregisterPerintah melakukan hal berikut:

1. Menderegister instance lokal dengan. CodeDeploy
2. Jika ditentukan, menghapus pengguna IAM yang terkait dengan instans lokal.

Setelah membatalkan pendaftaran instans lokal:

- Itu berhenti muncul di konsol segera.
- Anda dapat membuat instance lain dengan nama yang sama segera.

Jika perintah ini menemukan kesalahan, pesan kesalahan muncul, menjelaskan bagaimana Anda dapat menyelesaikan langkah-langkah yang tersisa secara manual. Jika tidak, pesan sukses muncul, menjelaskan cara memanggil uninstall perintah.

Deregister instans lokal secara manual

Biasanya, Anda membatalkan pendaftaran instans lokal setelah Anda tidak lagi berencana untuk menerapkannya. Anda menggunakan AWS CLI untuk membatalkan pendaftaran instans lokal secara manual.

Membatalkan pendaftaran instans lokal secara manual tidak menghapus instalan agen. CodeDeploy Itu tidak menghapus file konfigurasi dari instance. Itu tidak menghapus pengguna IAM yang terkait dengan instance. Itu tidak menghapus tag apa pun yang terkait dengan instance.

Untuk menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instance lokal, lihat. Menghapus instalan CodeDeploy agen secara otomatis dan menghapus file konfigurasi dari instans lokal

Untuk menghapus instalan hanya CodeDeploy agen secara manual, lihatMengelola operasi CodeDeploy agen.

Untuk menghapus pengguna IAM terkait secara manual, lihat Menghapus pengguna IAM dari akun Anda. AWS

Untuk menghapus hanya tag instans lokal terkait secara manual, lihatHapus tag instans lokal secara manual dari instans lokal.

- Panggil perintah deregister-on-premises-instance, dengan menentukan nama yang secara unik mengidentifikasi instance lokal (dengan opsi): `--instance-name`

  ```
  aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
  ```

  Setelah membatalkan pendaftaran instans lokal:

  - Itu berhenti muncul di konsol segera.
  - Anda dapat membuat instance lain dengan nama yang sama segera.

Lihat detail contoh dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang instance yang digunakan dalam penerapan.

Untuk informasi tentang menggunakan tindakan CodeDeploy API untuk melihat instance, lihat GetDeploymentInstanceListDeploymentInstances, dan ListOnPremisesInstances.

Topik

- Lihat detail contoh (konsol)
- Lihat detail contoh (CLI)

Lihat detail contoh (konsol)

Untuk melihat detail contoh:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Deployments.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

3. Untuk menampilkan detail penerapan, pilih ID penerapan untuk instance.
4. Anda dapat melihat semua instance di bagian Aktivitas instans pada halaman penerapan.
5. Untuk melihat informasi tentang peristiwa siklus hidup penerapan individual untuk sebuah instance, pada halaman detail penerapan, di kolom Peristiwa, pilih Lihat peristiwa.

> **Note**
>
> Jika Gagal ditampilkan untuk salah satu peristiwa siklus hidup, pada halaman detail instans, pilih Lihat log, Lihat di EC2, atau keduanya. Anda dapat menemukan tips pemecahan masalah di. Memecahkan masalah contoh

6. Jika Anda ingin melihat informasi selengkapnya tentang instans Amazon EC2, pilih ID instance di kolom ID Instans.

Lihat detail contoh (CLI)

Untuk menggunakan detail instance AWS CLI to view, panggil `get-deployment-instance` perintah atau `list-deployment-instances` perintah.

Untuk melihat detail tentang satu instance, panggil perintah get-deployment-instance, dengan menentukan:

- ID penyebaran unik. Untuk mendapatkan ID penerapan, panggil perintah list-deployments.
- ID contoh unik. Untuk mendapatkan ID instance, panggil perintah list-deployment-instances.

Untuk melihat daftar ID untuk instance yang digunakan dalam penerapan, panggil perintah list-deployment-instance, dengan menentukan:

- ID penyebaran unik. Untuk mendapatkan ID penerapan, panggil perintah list-deployments.
- Secara opsional, apakah hanya menyertakan ID instans tertentu berdasarkan status penerapannya. (Jika tidak ditentukan, semua ID instance yang cocok akan dicantumkan, terlepas dari status penerapannya.)

CodeDeploy misalnya kesehatan

CodeDeploy memantau status kesehatan instance dalam grup penyebaran. Penerapan gagal jika jumlah instance sehat berada di bawah jumlah minimum instance sehat yang telah ditentukan untuk grup penerapan selama penerapan. Misalnya, jika 85% instance harus tetap sehat selama penerapan, dan grup penerapan berisi 10 instance, penerapan keseluruhan akan gagal jika penerapan bahkan ke satu instance gagal. Ini karena ketika sebuah instance diambil offline sehingga revisi aplikasi terbaru dapat diinstal, jumlah instans sehat yang tersedia sudah turun menjadi 90%.

Instance yang gagal ditambah instance offline lainnya berarti hanya 80% instance yang sehat dan tersedia. CodeDeployakan gagal dalam penerapan keseluruhan.

Penting untuk diingat bahwa agar penerapan keseluruhan berhasil, berikut ini harus benar:

- CodeDeploy dapat menyebarkan ke setiap instance dalam penerapan.
- Penerapan ke setidaknya satu instance harus berhasil. Ini berarti bahwa meskipun nilai host sehat minimum adalah 0, penerapan ke setidaknya satu instance harus berhasil (yaitu, setidaknya satu instance harus sehat) agar penerapan keseluruhan berhasil.

Topik

- Status kondisi
- Tentang jumlah minimum contoh sehat
- Tentang jumlah minimum instans sehat per Availability Zone

Status kondisi

CodeDeploy memberikan dua nilai status kesehatan untuk setiap contoh: kesehatan revisi dan kesehatan instans.

Revisi kesehatan

Kesehatan revisi didasarkan pada revisi aplikasi yang saat ini diinstal pada instance. Ini memiliki nilai status berikut:

- Saat ini: Revisi yang diinstal pada instance cocok dengan revisi untuk penerapan terakhir grup penerapan yang berhasil.
- Lama: Revisi yang diinstal pada instance cocok dengan versi aplikasi yang lebih lama.
- Tidak diketahui: Revisi aplikasi belum berhasil diinstal pada instance.

Kesehatan contoh

Kesehatan instans didasarkan pada apakah penerapan ke suatu instance telah berhasil. Ini memiliki nilai-nilai berikut:

- Sehat: Penerapan terakhir ke instance berhasil.
- Tidak sehat: Upaya untuk menerapkan revisi ke instance gagal, atau revisi belum diterapkan ke instance.

CodeDeploy menggunakan kesehatan revisi dan kesehatan instans untuk menjadwalkan penerapan ke instance grup penerapan dalam urutan berikut:

1. Kesehatan contoh yang tidak sehat.
2. Kesehatan revisi tidak diketahui.
3. Kesehatan revisi lama.
4. Kesehatan revisi saat ini.

Jika penerapan keseluruhan berhasil, revisi diperbarui dan nilai status kesehatan grup penerapan diperbarui untuk mencerminkan penerapan terbaru.

- Semua instance saat ini yang memiliki penerapan yang berhasil tetap terkini. Kalau tidak, mereka menjadi tidak dikenal.
- Semua instance lama atau tidak dikenal yang memiliki penerapan yang berhasil menjadi terkini. Kalau tidak, mereka tetap tua atau tidak dikenal.
- Semua contoh sehat yang memiliki penyebaran yang berhasil tetap sehat. Kalau tidak, mereka menjadi tidak sehat.
- Semua contoh tidak sehat yang memiliki penyebaran yang sukses menjadi sehat. Kalau tidak, mereka tetap tidak sehat.

Jika penerapan keseluruhan gagal atau dihentikan:

- Setiap instance yang CodeDeploy mencoba menerapkan revisi aplikasi memiliki kesehatan instance-nya yang disetel ke sehat atau tidak sehat, tergantung pada apakah upaya penerapan untuk instance itu berhasil atau gagal.
- Setiap instance yang CodeDeploy tidak mencoba menerapkan revisi aplikasi mempertahankan nilai kesehatan instancenya saat ini.
- Revisi grup penyebaran tetap sama.

Tentang jumlah minimum contoh sehat

Jumlah minimum instans sehat yang diperlukan didefinisikan sebagai bagian dari konfigurasi penerapan.

> **Important**
>
> Selama blue/green penerapan, konfigurasi penerapan dan nilai host sehat minimum berlaku untuk instance di lingkungan pengganti, bukan yang ada di lingkungan asli. Namun, ketika instance di lingkungan asli dideregistrasi dari penyeimbang beban, penerapan keseluruhan ditandai sebagai Gagal jika bahkan satu instance asli gagal dideregistrasi dengan sukses.

CodeDeploy menyediakan tiga konfigurasi penerapan default yang biasanya menggunakan nilai host sehat minimum:

Nama konfigurasi penerapan default	Nilai host sehat minimum yang telah ditentukan sebelumnya
CodeDeployDefault.OneAtATime	1
CodeDeployDefault.HalfAtATime	50%
CodeDeployDefault.AllAtOnce	0

Anda akan menemukan informasi lebih lanjut tentang konfigurasi penerapan default di. Bekerja dengan konfigurasi penerapan di CodeDeploy

Anda dapat membuat konfigurasi penerapan khusus CodeDeploy untuk menentukan nilai host sehat minimum Anda sendiri. Anda dapat menentukan nilai-nilai ini sebagai bilangan bulat atau persentase saat menggunakan operasi berikut:

- Seperti `minimum-healthy-hosts` saat Anda menggunakan perintah create-deployment-config di. AWS CLI
- Seperti `Value` pada tipe MinimumHealthyHostsdata di CodeDeploy API.
- Seperti `MinimumHealthyHosts` saat Anda menggunakan AWS::CodeDeploy:: DeploymentConfig dalam CloudFormation templat.

CodeDeploy memungkinkan Anda menentukan jumlah minimum instance sehat untuk penyebaran untuk dua tujuan utama:

- Untuk menentukan apakah penerapan keseluruhan berhasil atau gagal. Penerapan berhasil jika revisi aplikasi berhasil diterapkan ke setidaknya jumlah minimum instance sehat.
- Untuk menentukan jumlah instance yang harus sehat selama penerapan agar penerapan dapat dilanjutkan.

Anda dapat menentukan jumlah minimum instans sehat untuk grup penyebaran Anda sebagai sejumlah instance atau sebagai persentase dari jumlah total instans. Jika Anda menentukan persentase, maka pada awal penerapan, CodeDeploy mengonversi persentase ke jumlah instance yang setara, membulatkan instance fraksional apa pun.

CodeDeploy melacak status kesehatan instans grup penyebaran selama proses penerapan dan menggunakan jumlah minimum instans sehat yang ditentukan penerapan untuk menentukan apakah akan melanjutkan penerapan. Prinsip dasarnya adalah bahwa penerapan tidak boleh menyebabkan jumlah instance sehat turun di bawah jumlah minimum yang telah Anda tentukan. Satu pengecualian untuk aturan ini adalah ketika grup penerapan awalnya memiliki kurang dari jumlah minimum instance sehat yang ditentukan. Dalam hal ini, proses penyebaran tidak mengurangi jumlah instance sehat lebih jauh.

> **Note**
>
> CodeDeploy akan mencoba menerapkan ke semua instance dalam grup penerapan, bahkan yang saat ini dalam status Berhenti. Dalam perhitungan host sehat minimum, instance yang dihentikan memiliki dampak yang sama dengan instance yang gagal. Untuk mengatasi kegagalan penerapan karena terlalu banyak instance yang dihentikan, mulai ulang instance atau ubah tag mereka untuk mengecualikannya dari grup penerapan.

CodeDeploy memulai proses penerapan dengan mencoba menerapkan revisi aplikasi ke instance tidak sehat grup penerapan. Untuk setiap penerapan yang berhasil, CodeDeploy ubah status kesehatan instans menjadi sehat dan menambahkannya ke instans sehat grup penerapan. CodeDeploy kemudian membandingkan jumlah contoh sehat saat ini dengan jumlah minimum contoh sehat yang ditentukan.

- Jika jumlah instans sehat kurang dari atau sama dengan jumlah minimum instans sehat yang ditentukan, CodeDeploy batalkan penerapan untuk memastikan jumlah instans sehat tidak berkurang dengan lebih banyak penerapan.

- Jika jumlah instans sehat lebih besar dari jumlah minimum instans sehat yang ditentukan oleh setidaknya satu, CodeDeploy menyebarkan revisi aplikasi ke kumpulan asli instans sehat.

Jika penerapan ke instance sehat gagal, CodeDeploy ubah status kesehatan instans tersebut menjadi tidak sehat. Saat penerapan berlangsung, CodeDeploy perbarui jumlah instans sehat saat ini dan membandingkannya dengan jumlah minimum instans sehat yang ditentukan. Jika jumlah instans sehat jatuh ke angka minimum yang ditentukan pada titik mana pun dalam proses penerapan, CodeDeploy hentikan penerapan. Praktik ini mencegah kemungkinan penerapan berikutnya akan gagal, menjatuhkan jumlah instance sehat di bawah angka minimum yang ditentukan.

> **Note**
>
> Pastikan jumlah minimum instans sehat yang Anda tentukan kurang dari jumlah total instance dalam grup penyebaran. Jika Anda menentukan nilai persentase, ingat itu akan dibulatkan ke atas. Jika tidak, saat penerapan dimulai, jumlah instance sehat sudah kurang dari atau sama dengan jumlah minimum instans sehat yang ditentukan, dan CodeDeploy akan segera gagal dalam penerapan keseluruhan.

CodeDeploy juga menggunakan jumlah minimum instans sehat yang ditentukan dan jumlah sebenarnya dari instans sehat untuk menentukan apakah dan bagaimana menerapkan revisi aplikasi ke beberapa instance. Secara default, CodeDeploy menerapkan revisi aplikasi ke sebanyak mungkin instance tanpa risiko jumlah instans sehat berada di bawah jumlah minimum instans sehat yang ditentukan.

Untuk menentukan jumlah instance yang harus digunakan sekaligus, CodeDeploy gunakan perhitungan berikut:

```
[total-hosts] - [minimum-healthy-hosts] =
        [number-of-hosts-to-deploy-to-at-once]
```

Contoh:

- Jika grup penerapan Anda memiliki 10 instans dan Anda menetapkan nomor instans sehat minimum menjadi 9, CodeDeploy deploy ke 1 instans sekaligus.
- Jika grup penerapan Anda memiliki 10 instans dan Anda menetapkan nomor instans sehat minimum menjadi 3, CodeDeploy deploy ke 7 instans pada saat yang sama di batch pertama, dan kemudian ke 3 sisanya di batch kedua.

- Jika grup penerapan Anda memiliki 10 instans dan Anda menyetel nomor instans sehat minimum ke 0, CodeDeploy deploy ke 10 instans secara bersamaan.

Contoh

Contoh berikut mengasumsikan grup penerapan dengan 10 instance.

Contoh sehat minimum: 95%

CodeDeploy membulatkan jumlah instans sehat minimum hingga 10 kali, yang sama dengan jumlah contoh sehat. Penerapan keseluruhan segera gagal tanpa menerapkan revisi ke instance apa pun.

Contoh sehat minimum: 9

CodeDeploy menyebarkan revisi ke satu instance pada satu waktu. Jika penerapan ke salah satu instance gagal, CodeDeploy segera gagal penerapan keseluruhan karena jumlah instans sehat sama dengan jumlah instans sehat minimum. Pengecualian untuk aturan ini adalah bahwa jika instance terakhir gagal, penerapan masih berhasil.

CodeDeploy melanjutkan penerapan, satu instance pada satu waktu, hingga penerapan apa pun gagal atau penerapan keseluruhan selesai. Jika semua 10 penerapan berhasil, grup penerapan sekarang memiliki 10 instance sehat.

Contoh sehat minimum: 8

CodeDeploy menyebarkan revisi ke dua instance sekaligus. Jika dua penerapan ini gagal, CodeDeploy segera gagal penerapan keseluruhan. Pengecualian untuk aturan ini adalah bahwa jika instance terakhir adalah yang kedua gagal, penerapan masih berhasil.

Contoh sehat minimum: 0

CodeDeploy menyebarkan revisi ke seluruh grup penyebaran sekaligus. Setidaknya satu penerapan ke instance harus berhasil agar penerapan keseluruhan berhasil. Jika 0 instance sehat, maka penerapan gagal. Ini karena persyaratan bahwa untuk menandai penerapan keseluruhan sebagai berhasil, setidaknya satu instance harus sehat ketika penerapan keseluruhan selesai, bahkan jika nilai instans sehat minimum adalah 0.

Tentang jumlah minimum instans sehat per Availability Zone

> **Note**
>
> Bagian ini menggunakan istilah instance dan host secara bergantian untuk merujuk ke instans Amazon EC2.

Jika Anda menerapkan ke instance di beberapa Availability Zone, Anda dapat mengaktifkan zonal configuration fitur secara opsional, yang memungkinkan CodeDeploy untuk menerapkan ke satu Availability Zone pada satu waktu.

Ketika fitur ini diaktifkan, CodeDeploy akan memastikan bahwa jumlah host sehat tetap di atas 'host sehat minimum per zona' dan nilai 'host sehat minimum'. Jika jumlah host sehat turun di bawah salah satu nilai, CodeDeploy gagal penerapan di semua Availability Zone.

Untuk menghitung jumlah host yang akan digunakan sekaligus, CodeDeploy gunakan nilai 'host sehat minimum per zona' dan 'host sehat minimum'. CodeDeploy akan menggunakan perhitungan yang lebih rendah `[A]` dan`[B]`, di mana `[A]` dan `[B]` adalah:

```
[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]
```

Setelah menentukan jumlah host yang akan diterapkan sekaligus, CodeDeploy menyebarkan ke host dalam batch nomor itu, satu Availability Zone pada satu waktu, dengan jeda opsional (atau 'waktu pemanggangan') di antara zona.

Contoh

Jika penerapan Anda dikonfigurasi seperti ini:

- `[total-hosts]`adalah `200`
- `[minimum-healthy-hosts]`adalah `160`
- `[total-hosts-per-AZ]`adalah `100`
- `[minimum-healthy-hosts-per-AZ]`adalah `50`

Maka...

- `[A]` = `200 - 160 = 40`
- `[B]` = `100 - 50 = 50`
- `40`kurang dari `50`

Oleh karena itu, CodeDeploy akan menyebarkan ke `40` host sekaligus.

Dalam skenario ini, penerapan terungkap sebagai berikut:

1. CodeDeploy menyebarkan ke Availability Zone pertama:
   a. CodeDeploy menyebarkan ke `40` host pertama.
   b. CodeDeploy menyebarkan ke `40` host berikutnya.
   c. CodeDeploy menyebarkan ke `20` host yang tersisa.

   Penerapan ke Availability Zone pertama sekarang selesai.
2. (Opsional) CodeDeploy menunggu sementara penerapan ke zona pertama 'bakes', seperti yang ditentukan oleh durasi Monitor atau Tambahkan durasi monitor untuk pengaturan zona pertama. Jika tidak ada masalah, CodeDeploy lanjutkan.
3. CodeDeploy menyebarkan ke Availability Zone kedua:
   a. CodeDeploy menyebarkan ke `40` host pertama.
   b. CodeDeploy menyebarkan ke `40` host berikutnya.
   c. CodeDeploy menyebarkan ke `20` host yang tersisa.

   Penerapan ke Availability Zone kedua dan terakhir sekarang selesai.

Untuk mempelajari tentang fitur konfigurasi zona, dan cara menentukan jumlah minimum instans sehat per Availability Zone, lihat. zonal configuration

Bekerja dengan konfigurasi penerapan di CodeDeploy

Konfigurasi penerapan adalah seperangkat aturan dan kondisi keberhasilan dan kegagalan yang digunakan CodeDeploy selama penerapan. Aturan dan ketentuan ini berbeda, tergantung pada apakah Anda menerapkan ke platform komputasi, platform EC2/On-Premises komputasi AWS Lambda, atau platform komputasi Amazon ECS.

Konfigurasi penerapan pada platform komputasi EC2/on-premises

Saat Anda menerapkan ke platform EC2/On-Premises komputasi, konfigurasi penerapan menentukan, melalui penggunaan nilai 'host sehat minimum' dan nilai opsional 'host sehat minimum per zona', jumlah atau persentase instance yang harus tetap tersedia kapan saja selama penerapan.

Anda dapat menggunakan salah satu dari tiga konfigurasi penerapan standar yang disediakan oleh AWS atau membuat konfigurasi penerapan kustom. Untuk informasi selengkapnya tentang membuat konfigurasi penerapan kustom, lihat. Create a Deployment Configuration Jika Anda tidak menentukan konfigurasi penerapan, CodeDeploy gunakan konfigurasi CodeDeployDefault.OneAtATime penerapan.

Untuk informasi selengkapnya tentang cara CodeDeploy memantau dan mengevaluasi kesehatan instans selama penerapan, lihat. Instance Health Untuk melihat daftar konfigurasi penerapan yang sudah terdaftar ke AWS akun Anda, lihat. View Deployment Configuration Details

Konfigurasi penerapan yang telah ditentukan sebelumnya untuk platform komputasi EC2/on-premises

Tabel berikut mencantumkan konfigurasi penerapan yang telah ditentukan.

> **Note**
>
> Tidak ada konfigurasi penerapan standar yang mendukung zonal configuration fitur (yang merupakan fitur yang memungkinkan Anda menentukan jumlah host sehat per Availability Zone). Jika Anda ingin menggunakan fitur ini, Anda harus membuat konfigurasi penerapan Anda sendiri.

Konfigurasi deployment	Deskripsi
CodeDeployDefault.AllAtOnce	In-place penyebaran: Mencoba untuk menyebarkan revisi aplikasi ke sebanyak mungkin contoh sekaligus. Status penyebaran keseluruhan ditampilkan sebagai Berhasil jika revisi aplikasi diterapkan ke satu atau beberapa instance. Status penerapan keseluruhan ditampilkan sebagai Gagal jika revisi aplikasi tidak diterapkan ke salah satu instance. Menggunakan contoh sembilan instance, CodeDeployDefault.AllAtOnce upaya untuk menyebarkan ke sembilan instance sekaligus. Penerapan keseluruhan berhasil jika penerapan bahkan ke satu instance berhasil. Gagal hanya jika penerapan ke sembilan instance gagal. Blue/green penyebaran: • Penerapan ke lingkungan pengganti: Mengikuti aturan penerapan yang sama seperti CodeDeployDefault.AllAtOnce untuk penerapan di tempat. • Pengalihan rute lalu lintas: Rutekan lalu lintas ke semua instance di lingkungan pengganti sekaligus. Berhasil jika lalu lintas berhasil dialihkan ke setidaknya satu instance. Gagal setelah mengalihkan rute ke semua instance gagal.
CodeDeployDefault.HalfAtATime	In-place penyebaran: Menyebarkan hingga setengah dari instance sekaligus (dengan pecahan dibulatkan ke bawah). Penerapan keseluruhan berhasil jika revisi aplikasi diterapkan ke

Konfigurasi deployment	Deskripsi
	setidaknya setengah dari instance (dengan pecahan dibulatkan). Jika tidak, deploymen t gagal. Dalam contoh sembilan contoh, ia menyebarkan hingga empat instance sekaligus. Penerapan keseluruhan berhasil jika penerapan ke lima instance atau lebih berhasil. Jika tidak, deployment gagal. **Note** Jika Anda menerapkan ke instance di beberapa grup Auto Scaling CodeDeplo y , akan menerapkan hingga setengah dari instans sekaligus terlepas dari grup Auto Scaling tempat mereka berada. Misalnya, anggap Anda memiliki dua grup Auto Scaling`ASG2`, `ASG1` dan masing-masing dengan 10 instance. Dalam skenario ini, CodeDeploy mungkin menerapkan ke 10 instance secara adil `ASG1` dan menganggap ini berhasil karena telah diterapkan ke setidaknya setengah dari instance. Blue/green penyebaran: • Penerapan ke lingkungan pengganti: Mengikuti aturan penerapan yang sama seperti CodeDeployDefault.HalfAtATime untuk penerapan di tempat. • Pengalihan rute lalu lintas: Rutekan lalu lintas hingga setengah dari instance di lingkungan pengganti sekaligus. Berhasil jika mengalihk

Konfigurasi deployment	Deskripsi
	an ke setidaknya setengah dari instance berhasil. Jika tidak, gagal.

Konfigurasi deployment	Deskripsi
CodeDeployDefault.OneAtATime	In-place penyebaran: Menyebarkan revisi aplikasi hanya untuk satu instance pada satu waktu. Untuk grup penerapan yang berisi lebih dari satu instance: • Penerapan keseluruhan berhasil jika revisi aplikasi diterapkan ke semua instance. Pengecualian untuk aturan ini adalah jika penerapan ke instance terakhir gagal, penerapan keseluruhan masih berhasil. Ini karena CodeDeploy memungkinkan hanya satu instance pada satu waktu untuk diambil offline dengan CodeDeployDefault. OneAtATime konfigurasi. • Penerapan keseluruhan gagal segera setelah revisi aplikasi gagal diterapkan ke salah satu kecuali contoh terakhir. • Dalam contoh menggunakan sembilan instance, itu menyebarkan ke satu instance pada satu waktu. Penerapan keseluruhan berhasil jika penerapan ke delapan instance pertama berhasil. Penerapan keseluruh an gagal jika penerapan ke salah satu dari delapan instance pertama gagal. Untuk grup penerapan yang hanya berisi satu instance, penerapan keseluruhan hanya berhasil jika penerapan ke instance tunggal berhasil. Blue/green penyebaran:

Konfigurasi deployment	Deskripsi
	• Penerapan ke lingkungan pengganti: Mengikuti aturan penerapan yang sama seperti CodeDeployDefault.OneAtATime untuk penerapan di tempat. • Pengalihan rute lalu lintas: Rutekan lalu lintas ke satu contoh di lingkungan pengganti pada satu waktu. Berhasil jika lalu lintas berhasil dialihkan ke semua instance pengganti. Gagal setelah kegagalan pengalihan rute pertama. Pengecualian untuk aturan ini adalah bahwa jika instance terakhir gagal mendaftar, penerapan keseluruhan masih berhasil.

Konfigurasi deployment pada platform komputasi Amazon ECS

Saat Anda menerapkan ke platform komputasi Amazon ECS, konfigurasi penerapan menentukan cara lalu lintas dialihkan ke set tugas Amazon ECS yang diperbarui. Anda dapat mengalihkan lalu lintas menggunakan konfigurasi penerapan canary, linear, atau all-at-once. Untuk informasi selengkapnya, lihat Konfigurasi deployment.

Anda juga dapat membuat konfigurasi penyebaran kenari atau linier kustom Anda sendiri. Untuk informasi selengkapnya, lihat Create a Deployment Configuration.

Konfigurasi penerapan yang telah ditentukan sebelumnya untuk platform komputasi Amazon ECS

Tabel berikut mencantumkan konfigurasi standar yang tersedia untuk penerapan Amazon ECS.

> **Note**
>
> Jika Anda menggunakan Network Load Balancer, hanya konfigurasi penerapan yang `CodeDeployDefault.ECSAllAtOnce` telah ditentukan yang didukung.

Konfigurasi deployment	Deskripsi
CodeDeployDefault.ECSLinear10PercentEvery1Minutes	Menggeser 10 persen lalu lintas setiap menit sampai semua lalu lintas bergeser.
CodeDeployDefault.ECSLinear10PercentEvery3Minutes	Menggeser 10 persen lalu lintas setiap tiga menit sampai semua lalu lintas bergeser.
CodeDeployDefault.ECSCanary10Percent5Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. 90 persen sisanya di-deploy lima menit kemudian.
CodeDeployDefault.ECSCanary10Percent15Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. Sisa 90 persen di-deploy 15 menit kemudian.
CodeDeployDefault.ECSAllAtOnce	Menggeser semua lalu lintas ke wadah Amazon ECS yang diperbarui sekaligus.

Konfigurasi penerapan untuk CloudFormation blue/green penerapan (Amazon ECS)

Saat Anda menerapkan ke platform komputasi Amazon ECS melalui CloudFormation blue/green penerapan, konfigurasi penerapan menentukan cara lalu lintas dialihkan ke wadah Amazon ECS yang diperbarui. Anda dapat mengalihkan lalu lintas menggunakan konfigurasi penerapan canary, linear, atau all-at-once. Untuk informasi selengkapnya, lihat Konfigurasi deployment.

Dengan CloudFormation blue/green penerapan, Anda tidak dapat membuat konfigurasi canary atau penerapan linier kustom Anda sendiri. Untuk petunjuk langkah demi langkah tentang penggunaan CloudFormation untuk mengelola blue/green penerapan Amazon ECS, lihat Mengotomatiskan penerapan ECS melalui blue/green penggunaan di Panduan Pengguna. CodeDeploy CloudFormationCloudFormation

> **Note**
>
> Mengelola blue/green penyebaran Amazon ECS tidak CloudFormation tersedia di Wilayah Eropa (Milan), Afrika (Cape Town), dan Asia Pasifik (Osaka).

Konfigurasi penerapan pada sebuah AWS Lambda platform komputasi

Saat Anda menerapkan ke platform komputasi AWS Lambda, konfigurasi penerapan menentukan cara lalu lintas dialihkan ke versi fungsi Lambda baru di aplikasi Anda. Anda dapat mengalihkan lalu lintas menggunakan konfigurasi penerapan canary, linear, atau all-at-once. Untuk informasi selengkapnya, lihat Konfigurasi deployment.

Anda juga dapat membuat konfigurasi penyebaran kenari atau linier kustom Anda sendiri. Untuk informasi selengkapnya, lihat Create a Deployment Configuration.

Konfigurasi penerapan yang telah ditentukan untuk sebuah AWS Lambda platform komputasi

Tabel berikut mencantumkan konfigurasi standar yang tersedia untuk AWS Lambda penerapan.

Konfigurasi deployment	Deskripsi
CodeDeployDefault.LambdaCanary10Percent5Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. 90 persen sisanya di-deploy lima menit kemudian.
CodeDeployDefault.LambdaCanary10Percent10Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. Sisanya 90 persen dikerahkan 10 menit kemudian.
CodeDeployDefault.LambdaCanary10Percent15Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. Sisa 90 persen di-deploy 15 menit kemudian.

Konfigurasi deployment	Deskripsi
CodeDeployDefault.LambdaCanary10Percent30Minutes	Menggeser 10 persen lalu lintas pada kenaikan pertama. Sisanya 90 persen dikerahkan 30 menit kemudian.
CodeDeployDefault.LambdaLinear10PercentEvery1Minute	Menggeser 10 persen lalu lintas setiap menit sampai semua lalu lintas bergeser.
CodeDeployDefault.LambdaLinear10PercentEvery2Minutes	Menggeser 10 persen lalu lintas setiap dua menit sampai semua lalu lintas bergeser.
CodeDeployDefault.LambdaLinear10PercentEvery3Minutes	Menggeser 10 persen lalu lintas setiap tiga menit sampai semua lalu lintas bergeser.
CodeDeployDefault.LambdaLinear10PercentEvery10Minutes	Menggeser 10 persen lalu lintas setiap 10 menit sampai semua lalu lintas bergeser.
CodeDeployDefault.LambdaAllAtOnce	Menggeser semua lalu lintas ke fungsi Lambda yang diperbarui sekaligus.

Topik

- Create a Deployment Configuration
- View Deployment Configuration Details
- Delete a Deployment Configuration

Buat konfigurasi penerapan dengan CodeDeploy

Jika Anda tidak ingin menggunakan salah satu konfigurasi penerapan default yang disediakan CodeDeploy, Anda dapat membuatnya sendiri menggunakan petunjuk berikut.

Anda dapat menggunakan CodeDeploy konsol AWS CLI, CodeDeploy API, atau CloudFormation templat untuk membuat konfigurasi penerapan khusus.

Untuk informasi tentang menggunakan CloudFormation templat untuk membuat konfigurasi penerapan, lihatCloudFormation template untuk CodeDeploy referensi.

Topik

- Membuat konfigurasi penerapan (konsol)
- Membuat konfigurasi penerapan dengan CodeDeploy (AWS CLI)

Membuat konfigurasi penerapan (konsol)

Gunakan petunjuk berikut untuk membuat konfigurasi penerapan menggunakan AWS konsol.

Untuk membuat konfigurasi penerapan dalam CodeDeploy menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, pilih konfigurasi Deployment.

   Daftar konfigurasi penerapan bawaan muncul.

3. Pilih Buat konfigurasi penerapan.
4. Dalam nama konfigurasi Deployment, masukkan nama untuk konfigurasi deployment. Misalnya, **`my-deployment-config`**.
5. Di bawah platform Compute, pilih salah satu dari berikut ini:

   - EC2/On-premises
   - AWS Lambda
   - Amazon ECS

6. Lakukan salah satu tindakan berikut:

   - Jika Anda memilih EC2/On-premises:

     1. Di bawah Host sehat minimum, tentukan jumlah atau persentase instans yang harus tetap tersedia kapan saja selama penerapan. Untuk informasi selengkapnya tentang cara CodeDeploy memantau dan mengevaluasi kesehatan instans selama penerapan, lihat. Instance Health

2. (Opsional) Di bawah konfigurasi Zonal, pilih Aktifkan konfigurasi zona untuk CodeDeploy menerapkan aplikasi Anda ke satu Availability Zone pada satu waktu, di dalam Region. AWS Dengan menerapkan ke satu Availability Zone pada satu waktu, Anda dapat mengekspos penerapan Anda ke audiens yang semakin besar seiring meningkatnya kepercayaan terhadap kinerja dan kelayakan penerapan. Jika Anda tidak mengaktifkan konfigurasi zona, CodeDeploy gunakan aplikasi Anda ke pilihan host acak di seluruh Wilayah.

   Jika Anda mengaktifkan fitur konfigurasi zona, perhatikan hal berikut:

   - Fitur konfigurasi zona hanya didukung dengan penerapan di tempat ke instans Amazon EC2. (Blue/greenpenerapan dan instans lokal tidak didukung.) Untuk informasi selengkapnya tentang penerapan di tempat, lihat. Tipe deployment
   - Fitur konfigurasi zona tidak didukung dengan konfigurasi penerapan yang telah ditentukan sebelumnya. Untuk menggunakan konfigurasi zona, Anda harus membuat konfigurasi penerapan kustom, seperti yang dijelaskan di sini.
   - Jika CodeDeploy perlu memutar kembali penyebaran, CodeDeploy akan melakukan operasi rollback pada host acak. (tidak CodeDeploy akan memutar kembali satu zona pada satu waktu, seperti yang Anda duga.) Perilaku rollback ini dipilih karena alasan kinerja. Untuk informasi selengkapnya tentang rollback, lihat. Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy

3. Jika Anda memilih kotak centang Aktifkan konfigurasi zona, secara opsional tentukan opsi berikut:

   - (Opsional) Dalam durasi Monitor, tentukan periode waktu, dalam hitungan detik, yang CodeDeploy harus menunggu setelah menyelesaikan penerapan ke Availability Zone. CodeDeploy akan menunggu jumlah waktu ini sebelum memulai penerapan ke Availability Zone berikutnya. Pertimbangkan untuk menambahkan durasi monitor untuk memberikan waktu penerapan untuk membuktikan dirinya (atau 'memanggang') di satu Availability Zone sebelum dirilis di zona berikutnya. Jika Anda tidak menentukan durasi monitor, segera CodeDeploy mulai menerapkan ke Availability Zone berikutnya. Untuk informasi selengkapnya tentang cara kerja setelan durasi Monitor, lihatTentang jumlah minimum instans sehat per Availability Zone.
   - (Opsional) Pilih Tambahkan durasi monitor untuk zona pertama untuk menetapkan durasi monitor yang hanya berlaku untuk Availability Zone pertama. Anda dapat mengatur opsi ini jika Anda ingin mengizinkan waktu pemanggangan ekstra untuk Availability Zone

pertama. Jika Anda tidak menentukan nilai dalam Tambahkan durasi monitor zona pertama, CodeDeploy gunakan nilai durasi Monitor untuk Availability Zone pertama.

- (Opsional) Di bawah Host sehat minimum per zona, tentukan jumlah atau persentase instance yang harus tetap tersedia per Zona Ketersediaan selama penerapan. Pilih FLEET_PERCENT untuk menentukan persentase, atau HOST_COUNT untuk menentukan nomor. Bidang ini bekerja sama dengan bidang host sehat Minimum. Untuk informasi selengkapnya, lihat Tentang jumlah minimum instans sehat per Availability Zone.

  Jika Anda tidak menentukan nilai di bawah Host sehat minimum per zona, maka CodeDeploy gunakan nilai default `0` persen.

- Jika Anda memilih AWS Lambdaatau Amazon ECS:

  1. Untuk Tipe, pilih Linear atau Canary.
  2. Di bidang Langkah dan Interval, lakukan salah satu hal berikut:

     - Jika Anda memilih Canary, untuk Langkah, masukkan persentase lalu lintas, antara 1 dan 99, yang akan digeser. Ini adalah persentase lalu lintas yang bergeser pada kenaikan pertama. Lalu lintas yang tersisa digeser setelah interval yang dipilih dalam kenaikan kedua.

       Untuk Interval, masukkan jumlah menit antara shift lalu lintas pertama dan kedua.

     - Jika Anda memilih Linear, untuk Langkah, masukkan persentase lalu lintas, antara 1 dan 99, yang akan digeser. Ini adalah persentase lalu lintas yang digeser pada awal setiap interval.

       Untuk Interval, masukkan jumlah menit antara setiap shift tambahan.

7. Pilih Buat konfigurasi penerapan.

   Anda sekarang memiliki konfigurasi penerapan yang dapat Anda kaitkan dengan grup penyebaran.

Membuat konfigurasi penerapan dengan CodeDeploy (AWS CLI)

Untuk menggunakan AWS CLI untuk membuat konfigurasi penerapan, panggil perintah create-deployment-config.

Contoh berikut membuat konfigurasi EC2/On-Premises penerapan bernama `ThreeQuartersHealthy` yang membutuhkan 75% instance target agar tetap sehat selama penerapan:

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --
minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Contoh berikut membuat konfigurasi EC2/On-Premises penerapan bernama yang memerlukan 300 instance target `300Total50PerAZ` agar tetap sehat secara total per penerapan, dan 50 untuk tetap sehat per Availability Zone. Ini juga menetapkan durasi monitor 1 jam.

```
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}}'
```

Contoh berikut membuat konfigurasi penyebaran AWS Lambda bernama. `Canary25Percent45Minutes` Ini menggunakan pergeseran lalu lintas kenari untuk menggeser 25 persen lalu lintas pada kenaikan pertama. Sisanya 75 persen bergeser 45 menit kemudian:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

Contoh berikut membuat konfigurasi penyebaran Amazon ECS bernama. `Canary25Percent45Minutes` Ini menggunakan pergeseran lalu lintas kenari untuk menggeser 25 persen lalu lintas pada kenaikan pertama. Sisanya 75 persen bergeser 45 menit kemudian:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
 --traffic-routing-config
 "type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

Lihat detail konfigurasi penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang konfigurasi penerapan yang terkait dengan akun Anda AWS . Untuk deskripsi konfigurasi CodeDeploy penerapan yang telah ditentukan sebelumnya, lihat. Konfigurasi penerapan yang telah ditentukan sebelumnya untuk platform komputasi EC2/on-premises

Topik

- Lihat detail konfigurasi penerapan (konsol)
- Lihat konfigurasi penerapan (CLI)

Lihat detail konfigurasi penerapan (konsol)

Untuk menggunakan CodeDeploy konsol untuk melihat daftar nama konfigurasi penerapan:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih konfigurasi Deployment.

   Di sini Anda dapat melihat nama dan kriteria konfigurasi penerapan untuk setiap konfigurasi penerapan.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

Lihat konfigurasi penerapan (CLI)

Untuk menggunakan detail konfigurasi AWS CLI untuk melihat penerapan, panggil `get-deployment-config` perintah atau `list-deployment-configs` perintah.

Untuk melihat detail tentang konfigurasi penerapan tunggal, panggil perintah get-deployment-config, dengan menentukan nama konfigurasi penerapan yang unik.

Untuk melihat detail tentang beberapa konfigurasi penerapan, panggil perintah list-deployments.

Hapus konfigurasi penerapan dengan CodeDeploy

Anda dapat menggunakan AWS CLI atau CodeDeploy API untuk menghapus konfigurasi penerapan kustom yang terkait dengan akun Anda AWS . Anda tidak dapat menghapus konfigurasi penerapan bawaan, seperti, `CodeDeployDefault.AllAtOnceCodeDeployDefault.HalfAtATime`, dan. `CodeDeployDefault.OneAtATime`

> **Warning**
>
> Anda tidak dapat menghapus konfigurasi penerapan kustom yang masih digunakan. Jika Anda menghapus konfigurasi penerapan kustom yang tidak digunakan, Anda tidak akan lagi dapat mengaitkannya dengan penerapan baru dan grup penerapan baru. Tindakan ini tidak dapat dibatalkan.

Untuk menggunakan konfigurasi AWS CLI to delete deployment, panggil perintah delete-deployment-config, dengan menentukan nama konfigurasi deployment. Untuk melihat daftar nama konfigurasi penerapan, panggil perintah list-deployment-configs.

Contoh berikut menghapus konfigurasi deployment bernama. ThreeQuartersHealthy

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Bekerja dengan aplikasi di CodeDeploy

Setelah Anda mengonfigurasi instance, tetapi sebelum Anda dapat menerapkan revisi, Anda harus membuat aplikasi di. CodeDeploy Aplikasi hanyalah nama atau wadah yang digunakan oleh CodeDeploy untuk memastikan revisi yang benar, konfigurasi penerapan, dan grup penyebaran direferensikan selama penerapan.

Gunakan informasi dalam tabel berikut untuk langkah selanjutnya:

Platform komputasi	Skenario	Informasi untuk langkah selanjutnya
EC2/On-Premises	Saya belum membuat instance.	LihatBekerja dengan instance untuk CodeDeploy, lalu kembali ke halaman ini.
EC2/On-Premises	Saya telah membuat instance, tetapi saya belum selesai menandai mereka.	LihatTagging Instances for Deployments, lalu kembali ke halaman ini.
EC2/On-Premises, AWS Lambda, dan Amazon ECS	Saya belum membuat aplikasi.	Lihat Buat aplikasi dengan CodeDeploy
EC2/On-Premises, AWS Lambda, dan Amazon ECS	Saya sudah membuat aplikasi, tetapi saya belum membuat grup penerapan.	Lihat Buat grup penerapan dengan CodeDeploy.
EC2/On-Premises, AWS Lambda, dan Amazon ECS	Saya telah membuat grup aplikasi dan penerapan, tetapi saya belum membuat revisi aplikasi.	Lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
EC2/On-Premises, AWS Lambda, dan Amazon ECS	Saya telah membuat grup aplikasi dan penerapan, dan saya telah mengunggah revisi aplikasi saya. Saya siap untuk menyebarkan.	Lihat Buat penerapan dengan CodeDeploy.

Topik

- Buat aplikasi dengan CodeDeploy
- Lihat detail aplikasi dengan CodeDeploy
- Membuat aturan notifikasi
- Ganti nama aplikasi CodeDeploy
- Hapus aplikasi di CodeDeploy

Buat aplikasi dengan CodeDeploy

Aplikasi hanyalah nama atau wadah yang digunakan CodeDeploy untuk memastikan bahwa revisi yang benar, konfigurasi penerapan, dan grup penyebaran direferensikan selama penerapan. Anda dapat menggunakan CodeDeploy konsol, CodeDeploy API, atau CloudFormation template untuk membuat aplikasi. AWS CLI

Kode Anda, atau revisi aplikasi, diinstal ke instance melalui proses yang disebut penerapan. CodeDeploy mendukung dua jenis penerapan:

- In-place deployment: Aplikasi pada setiap instance dalam grup penyebaran dihentikan, revisi aplikasi terbaru diinstal, dan versi baru aplikasi dimulai dan divalidasi. Anda dapat menggunakan penyeimbang beban sehingga setiap instance dideregistrasi selama penerapannya dan kemudian dikembalikan ke layanan setelah penerapan selesai. Hanya penerapan yang menggunakan platform EC2/On-Premises komputasi yang dapat menggunakan penerapan di tempat. Untuk informasi selengkapnya tentang penerapan di tempat, lihat. Ikhtisar penerapan di tempat
- Blue/green deployment: Perilaku penerapan Anda bergantung pada platform komputasi yang Anda gunakan:
  - Blue/green pada platform EC2/On-Premises komputasi: Instance dalam grup penerapan (lingkungan asli) digantikan oleh serangkaian instance yang berbeda (lingkungan pengganti) menggunakan langkah-langkah ini:
    - Instans disediakan untuk lingkungan pengganti.
    - Revisi aplikasi terbaru diinstal pada instance pengganti.
    - Waktu tunggu opsional terjadi untuk kegiatan seperti pengujian aplikasi dan verifikasi sistem.
    - Instans di lingkungan penggantian terdaftar dengan satu atau lebih penyeimbang beban Elastic Load Balancing, menyebabkan lalu lintas dialihkan ke sana. Contoh di lingkungan asli dideregistrasi dan dapat dihentikan atau terus berjalan untuk penggunaan lain.

> **Note**
>
> Jika Anda menggunakan platform EC2/On-Premises komputasi, ketahuilah bahwa blue/green penerapan hanya berfungsi dengan instans Amazon EC2.

- Blue/green pada platform komputasi Amazon ECS AWS Lambda atau Amazon: Lalu lintas digeser secara bertahap sesuai dengan konfigurasi penerapan kenari, linier, atau all-at-once.
- Blue/green penerapan melalui CloudFormation: Lalu lintas dialihkan dari sumber daya Anda saat ini ke sumber daya yang diperbarui sebagai bagian dari pembaruan CloudFormation tumpukan. Saat ini, hanya blue/green penerapan ECS yang didukung.

Untuk informasi selengkapnya tentang blue/green penerapan, lihat. Ikhtisar blue/green penerapan

Saat Anda menggunakan CodeDeploy konsol untuk membuat aplikasi, Anda mengonfigurasi grup penyebaran pertamanya secara bersamaan. Bila Anda menggunakan AWS CLI untuk membuat aplikasi, Anda membuat grup penyebaran pertama dalam langkah terpisah.

Untuk melihat daftar aplikasi yang sudah terdaftar ke AWS akun Anda, lihatLihat detail aplikasi dengan CodeDeploy. Untuk informasi tentang menggunakan CloudFormation templat untuk membuat aplikasi, lihatCloudFormation template untuk CodeDeploy referensi.

Kedua jenis penyebaran tidak berlaku untuk semua tujuan. Tabel berikut mencantumkan jenis penerapan mana yang bekerja dengan penerapan ke tiga jenis tujuan penerapan.

Tujuan penyebaran	In-place	Blue/green
Amazon EC2	Ya	Ya
On-premises	Ya	Tidak
Fungsi Lambda tanpa server AWS	Tidak	Ya
Aplikasi Amazon ECS	Tidak	Ya

Topik

- Buat aplikasi untuk penyebaran di tempat (konsol)
- Buat aplikasi untuk blue/green penyebaran (konsol)
- Membuat aplikasi untuk penyebaran layanan Amazon ECS (konsol)
- Buat aplikasi untuk AWS Lambda penyebaran fungsi (konsol)
- Buat aplikasi (CLI)

Buat aplikasi untuk penyebaran di tempat (konsol)

Untuk menggunakan CodeDeploy konsol untuk membuat aplikasi untuk penyebaran di tempat:

> **Warning**
>
> Jangan ikuti langkah-langkah ini jika:
>
> - Anda belum menyiapkan instance Anda untuk digunakan dalam CodeDeploy penerapan. Untuk menyiapkan instans Anda, ikuti petunjuknyaBekerja dengan instance untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.
> - Anda ingin membuat aplikasi yang menggunakan konfigurasi penerapan khusus, tetapi Anda belum membuat konfigurasi penerapan. Ikuti instruksi diCreate a Deployment Configuration, dan kemudian ikuti langkah-langkah dalam topik ini.
> - Anda tidak memiliki peran layanan yang dipercaya CodeDeploy dengan kepercayaan dan izin minimum yang diperlukan. Untuk membuat dan mengonfigurasi peran layanan dengan izin yang diperlukan, ikuti petunjuk diLangkah 2: Buat peran layanan untuk CodeDeploy, lalu kembali ke langkah-langkah dalam topik ini.
> - Anda ingin memilih Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing untuk penyebaran di tempat, tetapi belum membuatnya.

Untuk membuat aplikasi untuk penyebaran di tempat menggunakan konsol: CodeDeploy

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Memulai.
3. Pilih Create application (Buat aplikasi).
4. Dalam nama Aplikasi, masukkan nama aplikasi Anda.
5. Dari Compute Platform, pilih EC2/On-premises.
6. Pilih Create application (Buat aplikasi).
7. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group.
8. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

> **Note**
>
> Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran; tag, nama grup Amazon EC2 Auto Scaling, atau keduanya; dan konfigurasi penerapan), tentukan setelan tersebut di halaman ini. Meskipun grup penyebaran baru ini dan grup penyebaran yang ada memiliki nama yang sama, CodeDeploy memperlakukannya sebagai grup penyebaran terpisah, karena masing-masing terkait dengan aplikasi terpisah.

9. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke instance target Anda.
10. Dalam jenis Deployment, pilih In-place.
11. Dalam konfigurasi Lingkungan, pilih salah satu dari berikut ini:

    a. Grup Amazon EC2 Auto Scaling: Masukkan atau pilih nama grup Amazon EC2 Auto Scaling untuk menerapkan revisi aplikasi Anda. Saat instans Amazon EC2 baru diluncurkan sebagai bagian dari CodeDeploy grup Penskalaan Otomatis Amazon EC2, Anda dapat menerapkan revisi Anda ke instans baru secara otomatis. Anda dapat menambahkan hingga 10 grup Amazon EC2 Auto Scaling ke grup penerapan.

    b. Instans atau On-premisesinstans Amazon EC2: Di bidang Kunci dan Nilai, masukkan nilai pasangan nilai kunci yang Anda gunakan untuk menandai instans. Anda dapat menandai hingga 10 pasangan nilai kunci dalam satu grup tag.

i. Anda dapat menggunakan wildcard di bidang Nilai untuk mengidentifikasi semua instance yang ditandai dalam pola tertentu, seperti instans Amazon EC2 yang serupa, pusat biaya, dan nama grup, dan sebagainya. Misalnya, jika Anda memilih Nama di bidang Kunci dan masukkan **GRP-*a** di bidang Nilai, CodeDeploy mengidentifikasi semua instance yang sesuai dengan pola tersebut, seperti, **GRP-1aGRP-2a**, dan. **GRP-XYZ-a**

ii. Bidang Nilai peka huruf besar/kecil.

iii. Untuk menghapus pasangan kunci-nilai dari daftar, pilih Hapus tag.

Saat CodeDeploy menemukan instance yang cocok dengan setiap pasangan nilai kunci yang ditentukan atau nama grup Amazon EC2 Auto Scaling, ini menampilkan jumlah instance yang cocok. Pilih nomor untuk melihat informasi lebih lanjut tentang instans.

Jika Anda ingin menyempurnakan kriteria untuk instance yang disebarkan, pilih Tambahkan grup tag untuk membuat grup tag. Anda dapat membuat hingga tiga grup tag dengan masing-masing hingga sepuluh pasangan nilai kunci. Bila Anda menggunakan beberapa grup tag dalam grup penyebaran, hanya instance yang diidentifikasi oleh semua grup tag yang disertakan dalam grup penyebaran. Itu berarti sebuah instance harus cocok dengan setidaknya satu tag di setiap grup yang akan disertakan dalam grup penerapan.

Untuk informasi tentang menggunakan grup tag untuk menyempurnakan grup penerapan Anda, lihat. Tagging Instances for Deployments

12. Di setelan Deployment, pilih konfigurasi deployment untuk mengontrol laju penerapan aplikasi Anda ke instance, seperti satu per satu atau sekaligus. Untuk informasi selengkapnya tentang konfigurasi penerapan, lihat. Bekerja dengan konfigurasi penerapan di CodeDeploy
13. (Opsional) Di Load balancer, pilih Aktifkan load balancing, lalu dari daftar, pilih Classic Load Balancers, grup target Application Load Balancer, dan grup target Network Load Balancer untuk mengelola lalu lintas ke instans selama penyebaran. CodeDeploy Anda dapat memilih hingga 10 Classic Load Balancer dan 10 grup target, dengan total 20 item. Pastikan instans Amazon EC2 yang ingin Anda gunakan terdaftar dengan penyeimbang beban yang dipilih (Classic Load Balancer) atau grup target (Application Load Balancers dan Network Load Balancer).

    Selama penerapan, instance asli dideregistrasi dari penyeimbang beban dan grup target yang dipilih untuk mencegah lalu lintas dirutekan ke instance ini selama penerapan. Saat penerapan selesai, setiap instance didaftarkan ulang dengan semua Classic Load Balancer dan grup target yang dipilih.

Untuk informasi selengkapnya tentang penyeimbang beban untuk CodeDeploy penerapan, lihat. Integrating CodeDeploy with Elastic Load Balancing

14. (Opsional) Perluas Lanjutan, dan konfigurasikan opsi apa pun yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, atau rollback otomatis.

    Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup deployment.

15. Pilih Buat grup penyebaran.

Langkah selanjutnya adalah menyiapkan revisi untuk menyebarkan ke grup aplikasi dan penyebaran. Untuk petunjuk, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.

Buat aplikasi untuk blue/green penyebaran (konsol)

Untuk menggunakan CodeDeploy konsol untuk membuat aplikasi untuk blue/green penyebaran:

> **Note**
>
> Penerapan ke platform komputasi AWS Lambda selalu merupakan penerapan. blue/green Anda tidak menentukan opsi jenis penerapan.

> **Warning**
>
> Jangan ikuti langkah-langkah ini jika:
>
> - Anda tidak memiliki instance dengan CodeDeploy agen yang diinstal yang ingin Anda ganti selama proses blue/green penerapan. Untuk menyiapkan instans Anda, ikuti petunjuknyaBekerja dengan instance untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.
> - Anda ingin membuat aplikasi yang menggunakan konfigurasi penerapan khusus, tetapi Anda belum membuat konfigurasi penerapan. Ikuti instruksi diCreate a Deployment Configuration, dan kemudian ikuti langkah-langkah dalam topik ini.
> - Anda tidak memiliki peran layanan yang mempercayai CodeDeploy , setidaknya, kepercayaan dan izin yang dijelaskan dalam. Langkah 2: Buat peran layanan untuk CodeDeploy Untuk membuat dan mengonfigurasi peran layanan, ikuti petunjuk diLangkah 2: Buat peran layanan untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.

- Anda belum membuat Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing untuk pendaftaran instans di lingkungan pengganti Anda. Untuk informasi selengkapnya, lihat Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Memulai.
3. Dalam nama Aplikasi, masukkan nama aplikasi Anda.
4. Dari platform Compute, pilih EC2/On-Premises.
5. Pilih Create application (Buat aplikasi).
6. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group.
7. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan setelan yang sama yang digunakan di grup penerapan lain (termasuk tag nama grup penerapan, nama grup Amazon EC2 Auto Scaling, dan konfigurasi penerapan), pilih pengaturan tersebut di halaman ini. Meskipun grup penyebaran baru ini dan grup penyebaran yang ada memiliki nama yang sama, CodeDeploy memperlakukannya sebagai grup penyebaran terpisah, karena masing-masing dikaitkan dengan aplikasi terpisah.

8. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke instance target Anda.
9. Di Jenis Deployment pilih Blue/green.
10. Dalam konfigurasi Lingkungan, pilih metode yang akan digunakan untuk menyediakan instance untuk lingkungan pengganti Anda:

a. Secara otomatis menyalin grup Amazon EC2 Auto Scaling: membuat CodeDeploy grup Penskalaan Otomatis Amazon EC2 dengan menyalin grup yang Anda tentukan.

b. Penyediaan instance secara manual: Anda tidak akan menentukan instance untuk lingkungan pengganti hingga Anda membuat penerapan. Anda harus membuat instance sebelum memulai penerapan. Sebagai gantinya, di sini Anda menentukan instance yang ingin Anda ganti.

11. Tergantung pada pilihan Anda di langkah 10, lakukan salah satu hal berikut:

    - Jika Anda memilih Secara otomatis menyalin grup Amazon EC2 Auto Scaling: Di grup Amazon EC2 Auto Scaling, pilih atau masukkan nama grup Amazon EC2 Auto Scaling yang ingin Anda gunakan sebagai templat untuk grup Amazon EC2 Auto Scaling untuk instans di lingkungan pengganti Anda. Jumlah instans yang saat ini sehat di grup Amazon EC2 Auto Scaling yang Anda pilih dibuat di lingkungan pengganti Anda.
    - Jika Anda memilih Instans penyediaan secara manual: Aktifkan grup Penskalaan Otomatis Amazon EC2, instans Amazon EC2, atau keduanya untuk menentukan instans yang akan ditambahkan ke grup penerapan ini. Masukkan nilai tag Amazon EC2 atau nama grup Amazon EC2 Auto Scaling untuk mengidentifikasi instance di lingkungan asli Anda (yaitu instans yang ingin Anda ganti atau yang menjalankan revisi aplikasi saat ini).

12. Di Load balancer, pilih Aktifkan load balancing, lalu dari daftar, pilih Classic Load Balancer, grup target Application Load Balancer, dan grup target Network Load Balancer yang ingin Anda daftarkan instans Amazon EC2 pengganti Anda. Setiap instance pengganti akan didaftarkan dengan semua Classic Load Balancer dan grup target yang dipilih. Anda dapat memilih hingga 10 Classic Load Balancer dan 10 grup target, dengan total 20 item.

    Lalu lintas akan dialihkan dari yang asli ke instance pengganti sesuai dengan pengaturan pengalihan rute lalu lintas dan konfigurasi Deployment yang Anda pilih.

    Untuk informasi selengkapnya tentang penyeimbang beban untuk CodeDeploy penerapan, lihat. Integrating CodeDeploy with Elastic Load Balancing

13. Dalam pengaturan Deployment, tinjau opsi default untuk mengalihkan lalu lintas ke lingkungan pengganti, konfigurasi penerapan mana yang akan digunakan untuk penerapan, dan bagaimana instance di lingkungan asli ditangani setelah penerapan.

    Jika Anda ingin mengubah pengaturan, lanjutkan ke langkah berikutnya. Jika tidak, lewati ke langkah 15.

14. Untuk mengubah pengaturan penyebaran untuk blue/green penyebaran, ubah salah satu pengaturan berikut.

Pengaturan	Opsi
Pengalihan rute lalu lintas	• Rute lalu lintas segera: Segera setelah instance di lingkungan penggantian disediakan dan revisi aplikasi terbaru diinstal pada mereka, mereka terdaftar dengan penyeimbang beban yang ditentukan dan grup target secara otomatis, menyebabkan lalu lintas dialihkan ke mereka. Contoh di lingkungan asli kemudian dideregistrasi. • Saya akan memilih apakah akan mengalihkan lalu lintas: Contoh di lingkungan penggantian tidak terdaftar dengan penyeimbang beban dan grup target yang ditentukan kecuali Anda mengubah rute lalu lintas secara manual. Jika waktu tunggu yang Anda tentukan berlalu tanpa lalu lintas dialihkan, status penerapan diubah menjadi Berhenti.

Pengaturan	Opsi
Konfigurasi deployment	Pilih tingkat di mana instance di lingkungan penggantian terdaftar dengan penyeimbang beban dan kelompok sasaran, seperti satu per satu atau sekaligus. **Note** Setelah lalu lintas berhasil dialihkan ke lingkungan pengganti, instance di lingkungan asli dideregistrasi sekaligus tidak peduli konfigurasi penerapan mana yang dipilih. Untuk informasi selengkapnya, lihat Bekerja dengan konfigurasi penerapan di CodeDeploy.
Contoh asli	• Mengakhiri instance asli dalam grup penyebaran: Setelah lalu lintas dialihkan ke lingkungan pengganti, instance yang dideregistrasi dari penyeimbang beban dan grup target dihentikan setelah periode tunggu yang Anda tentukan. • Jaga agar instance asli dalam grup penerapan tetap berjalan: Setelah lalu lintas dialihkan ke lingkungan pengganti , instance yang dideregistrasi dari penyeimbang beban dan grup target tetap berjalan.

15. (Opsional) Di Lanjutan, konfigurasikan opsi yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, atau rollback otomatis.

    Untuk informasi tentang menentukan opsi lanjutan dalam grup penerapan, lihat. Mengonfigurasi opsi lanjutan untuk grup deployment

16. Pilih Buat grup penyebaran.

Langkah selanjutnya adalah menyiapkan revisi untuk menyebarkan ke grup aplikasi dan penyebaran. Untuk petunjuk, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.

Membuat aplikasi untuk penyebaran layanan Amazon ECS (konsol)

Anda dapat menggunakan CodeDeploy konsol untuk membuat aplikasi untuk penyebaran layanan Amazon ECS.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih Memulai.
3. Pada halaman Buat aplikasi, pilih Gunakan CodeDeploy.
4. Dalam nama Aplikasi, masukkan nama aplikasi Anda.
5. Dari platform Compute, pilih Amazon ECS.
6. Pilih Create application (Buat aplikasi).
7. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group. Untuk informasi selengkapnya tentang apa yang Anda perlukan untuk membuat grup penerapan untuk penyebaran Amazon ECS, lihat. Sebelum memulai deployment Amazon ECS
8. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran dan konfigurasi penerapan), pilih pengaturan tersebut di halaman ini. Meskipun grup baru ini dan grup yang ada mungkin memiliki nama yang sama, CodeDeploy memperlakukan mereka sebagai grup penyebaran terpisah, karena masing-masing dikaitkan dengan aplikasi terpisah.

9. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke Amazon ECS. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.
10. Dari nama Load balancer, pilih nama penyeimbang beban yang melayani lalu lintas ke layanan Amazon ECS Anda.
11. Dari port pendengar Produksi, pilih port dan protokol untuk pendengar yang menyajikan lalu lintas produksi ke layanan Amazon ECS Anda.
12. (Opsional) Dari port pendengar Uji, pilih port dan protokol pendengar pengujian yang menyajikan lalu lintas ke tugas pengganti yang ditetapkan dalam layanan Amazon ECS Anda selama penerapan. Anda dapat menentukan satu atau beberapa fungsi Lambda dalam AppSpec file yang berjalan selama hook. `AfterAllowTestTraffic` Fungsi dapat menjalankan tes validasi. Jika tes validasi gagal, rollback penerapan dipicu. Jika tes validasi berhasil, hook berikutnya dalam siklus hidup penerapan, `BeforeAllowTraffic` akan dipicu. Jika port pendengar uji tidak ditentukan, tidak ada yang terjadi selama `AfterAllowTestTraffic` pengait. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan Amazon ECS.
13. Dari nama grup target 1 dan nama Grup target 2, pilih grup target yang digunakan untuk merutekan lalu lintas selama penyebaran Anda. CodeDeploy mengikat satu grup target ke set tugas asli layanan Amazon ECS Anda dan yang lainnya ke set tugas penggantinya. Untuk informasi selengkapnya, lihat Grup Target untuk Penyeimbang Beban Aplikasi Anda.
14. Pilih Rute lalu lintas segera atau Tentukan kapan harus mengalihkan lalu lintas untuk menentukan kapan harus mengalihkan lalu lintas ke layanan Amazon ECS Anda yang diperbarui.

    Jika Anda memilih Reroute lalu lintas segera, maka penerapan secara otomatis mengalihkan lalu lintas setelah set tugas pengganti disediakan.

    Jika Anda memilih Tentukan kapan akan mengubah rute lalu lintas, lalu pilih jumlah hari, jam, dan menit untuk menunggu setelah set tugas pengganti berhasil disediakan. Selama waktu tunggu ini, tes validasi dalam fungsi Lambda yang ditentukan dalam file AppSpec dijalankan. Jika waktu tunggu berakhir sebelum lalu lintas dialihkan, maka status penerapan berubah menjadi. `Stopped`
15. Untuk penghentian revisi Asli, pilih jumlah hari, jam, dan menit untuk menunggu setelah penerapan berhasil sebelum tugas asli yang ditetapkan dalam layanan Amazon ECS Anda dihentikan.
16. (Opsional) Di Advanced, konfigurasikan opsi apa pun yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, atau rollback otomatis.

Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup deployment.

Buat aplikasi untuk AWS Lambda penyebaran fungsi (konsol)

Anda dapat menggunakan CodeDeploy konsol untuk membuat aplikasi untuk penerapan AWS Lambda fungsi.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih Memulai.
3. Pada halaman Buat aplikasi, pilih Gunakan CodeDeploy.
4. Masukkan nama aplikasi Anda di Nama aplikasi.
5. Dari platform Compute, pilih AWS Lambda.
6. Pilih Create application (Buat aplikasi).
7. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group.
8. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran dan konfigurasi penerapan), pilih pengaturan tersebut di halaman ini. Meskipun grup penyebaran baru ini dan grup penyebaran yang ada mungkin memiliki nama yang sama, CodeDeploy memperlakukannya sebagai grup penyebaran terpisah, karena masing-masing dikaitkan dengan aplikasi terpisah.

9. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke AWS Lambda. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

10. Jika Anda ingin menggunakan konfigurasi penerapan yang telah ditentukan sebelumnya, pilih salah satu dari konfigurasi Deployment, lalu lewati ke langkah 12. Untuk membuat konfigurasi khusus, lanjutkan ke langkah berikutnya.

    Untuk informasi selengkapnya tentang konfigurasi penerapan, lihat. Konfigurasi penerapan pada sebuah AWS Lambda platform komputasi

11. Untuk membuat konfigurasi kustom, pilih Buat konfigurasi penerapan, lalu lakukan hal berikut:

    a. Untuk nama konfigurasi Deployment, masukkan nama untuk konfigurasi.

    b. Dari Type, pilih tipe konfigurasi. Jika Anda memilih Canary, lalu lintas digeser dalam dua peningkatan. Jika Anda memilih Linear, lalu lintas digeser dalam peningkatan yang sama, dengan jumlah menit yang sama antara setiap kenaikan.

    c. Untuk Langkah, masukkan persentase lalu lintas, antara 1 dan 99, yang akan digeser. Jika jenis konfigurasi Anda adalah Canary, ini adalah persentase lalu lintas yang digeser pada kenaikan pertama. Lalu lintas yang tersisa digeser setelah interval yang dipilih dalam kenaikan kedua. Jika jenis konfigurasi Anda Linear, ini adalah persentase lalu lintas yang digeser pada awal setiap interval.

    d. Dalam Interval, masukkan jumlah menit. Jika jenis konfigurasi Anda adalah Canary, ini adalah jumlah menit antara shift lalu lintas pertama dan kedua. Jika jenis konfigurasi Anda adalah Linear, ini adalah jumlah menit antara setiap pergeseran tambahan.

    > **Note**
    >
    > Panjang maksimum AWS Lambda penyebaran adalah dua hari, atau 2.880 menit. Oleh karena itu, nilai maksimum yang ditentukan untuk Interval untuk konfigurasi kenari adalah 2.800 menit. Nilai maksimum untuk konfigurasi linier tergantung pada nilai untuk Langkah. Misalnya, jika persentase langkah dari pergeseran lalu lintas linier adalah 25%, maka ada empat pergeseran lalu lintas. Nilai interval maksimum adalah 2.880 dibagi empat, atau 720 menit.

    e. Pilih Buat konfigurasi penerapan.

12. (Opsional) Di Advanced, konfigurasikan opsi apa pun yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, atau rollback otomatis.

    Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup deployment.

13. Pilih Buat grup penyebaran.

Buat aplikasi (CLI)

Untuk menggunakan AWS CLI untuk membuat aplikasi, panggil perintah create-application, menentukan nama yang secara unik mewakili aplikasi. (Dalam AWS akun, nama CodeDeploy aplikasi hanya dapat digunakan sekali per Wilayah. Anda dapat menggunakan kembali nama aplikasi di Wilayah yang berbeda.)

Setelah Anda menggunakan AWS CLI untuk membuat aplikasi, langkah selanjutnya adalah membuat grup penyebaran yang menentukan instance untuk menyebarkan revisi. Untuk petunjuk, lihat Buat grup penerapan dengan CodeDeploy.

Setelah Anda membuat grup penyebaran, langkah selanjutnya adalah menyiapkan revisi untuk diterapkan ke grup aplikasi dan penyebaran. Untuk petunjuk, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.

Lihat detail aplikasi dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang semua aplikasi yang terkait dengan AWS akun Anda.

Topik

- Lihat detail aplikasi (konsol)
- Lihat detail aplikasi (CLI)

Lihat detail aplikasi (konsol)

Untuk menggunakan CodeDeploy konsol untuk melihat detail aplikasi:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, dan pilih Memulai.
3. Untuk melihat detail aplikasi tambahan, pilih nama aplikasi dalam daftar.

Lihat detail aplikasi (CLI)

Untuk menggunakan AWS CLI untuk melihat detail aplikasi, panggil get-application perintah, batch-get-application perintah, atau list-applications perintah.

Untuk melihat detail tentang satu aplikasi, panggil perintah get-application, dengan menentukan nama aplikasi.

Untuk melihat detail tentang beberapa aplikasi, panggil perintah batch-get-applications, dengan menentukan beberapa nama aplikasi.

Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.

Membuat aturan notifikasi

Anda dapat menggunakan aturan notifikasi untuk memberi tahu pengguna ketika ada perubahan pada aplikasi penerapan, seperti keberhasilan dan kegagalan penerapan. Aturan notifikasi menentukan peristiwa dan topik Amazon SNS yang digunakan untuk mengirim notifikasi. Untuk informasi selengkapnya, lihat Apa itu notifikasi?

Anda dapat menggunakan konsol atau AWS CLI untuk membuat aturan notifikasi untuk AWS CodeDeploy.

Untuk membuat aturan notifikasi (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
2. Pilih Aplikasi, lalu pilih aplikasi tempat Anda ingin menambahkan notifikasi.
3. Pada halaman aplikasi, pilih Beri tahu, lalu pilih Buat aturan pemberitahuan. Anda juga dapat pergi ke halaman Pengaturan untuk aplikasi dan memilih Buat aturan pemberitahuan.
4. Di Nama notifikasi, masukkan nama untuk aturan.
5. Di Jenis detail, pilih Dasar jika Anda hanya menginginkan informasi yang diberikan ke Amazon yang EventBridge disertakan dalam notifikasi. Pilih Lengkap jika Anda ingin menyertakan informasi yang diberikan ke Amazon EventBridge dan informasi yang mungkin diberikan oleh CodeDeploy atau pengelola notifikasi.

   Untuk informasi selengkapnya, lihat Memahami konten dan keamanan notifikasi.
6. Di Peristiwa yang memicu notifikasi, pilih peristiwa yang ingin Anda kirimkan notifikasi.

Kategori	Peristiwa
Deployment	Gagal
	Berhasil
	Dimulai

7. Di Target, pilih Buat topik SNS.

> **Note**
>
> Saat Anda membuat topik, kebijakan yang memungkinkan CodeDeploy untuk mempublikasikan acara ke topik diterapkan untuk Anda. Menggunakan topik yang dibuat khusus untuk CodeDeploy notifikasi juga membantu memastikan bahwa Anda hanya menambahkan pengguna ke daftar langganan untuk topik yang ingin Anda lihat pemberitahuan tentang aplikasi penerapan ini.

Setelah prefiks codestar-notifications-, masukkan nama untuk topik, lalu pilih nama, lalu pilih Kirim.

> **Note**
>
> Jika Anda ingin menggunakan topik Amazon SNS yang sudah ada alih-alih membuat yang baru, di Target, pilih ARN-nya. Pastikan topik memiliki kebijakan akses yang sesuai dan bahwa daftar pelanggan hanya berisi pengguna yang diizinkan untuk melihat informasi tentang aplikasi penyebaran. Untuk informasi selengkapnya, lihat Mengonfigurasi topik Amazon SNS yang ada untuk notifikasi serta Memahami konten dan keamanan notifikasi.

8. Untuk menyelesaikan pembuatan aturan, pilih Kirim.
9. Anda harus membuat pengguna berlangganan ke topik Amazon SNS untuk aturan tersebut sebelum mereka dapat menerima notifikasi. Untuk informasi selengkapnya, lihat Berlangganan pengguna ke topik Amazon SNS yang menjadi target. Anda juga dapat mengatur integrasi antara notifikasi dan Pengembang Amazon Q di aplikasi obrolan untuk mengirim pemberitahuan

ke ruang obrolan Amazon Chime atau saluran Slack. Untuk informasi selengkapnya, lihat Mengonfigurasi integrasi antara notifikasi dan Pengembang Amazon Q di aplikasi obrolan.

Untuk membuat aturan notifikasi (AWS CLI)

1. Di terminal atau prompt perintah, jalankan perintah create-notification rule untuk menghasilkan kerangka JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

   Anda dapat memberi nama file apa pun yang Anda inginkan. Dalam contoh ini, file dinamai *rule.json*.

2. Buka file JSON di editor teks biasa dan edit untuk menyertakan sumber daya, jenis acara, dan target Amazon SNS yang Anda inginkan untuk aturan tersebut. Contoh berikut menunjukkan aturan notifikasi bernama **MyNotificationRule** untuk aplikasi bernama *MyDeploymentApplication* dalam AWS akun dengan ID*123456789012*. Pemberitahuan dikirim dengan tipe detail lengkap ke topik Amazon SNS yang diberi nama *codestar-notifications-MyNotificationTopic* saat penerapan berhasil:

```
{
    "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"
}
```

   Simpan file tersebut.

3. Menggunakan file yang baru saja Anda edit, di terminal atau baris perintah, jalankan create-notification-rule perintah lagi untuk membuat aturan notifikasi:

```
aws codestar-notifications create-notification-rule --cli-input-json
 file://rule.json
```

4. Jika berhasil, perintah tersebut mengembalikan ARN aturan notifikasi, yang serupa dengan berikut ini:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Ganti nama aplikasi CodeDeploy

Anda dapat menggunakan AWS CLI atau CodeDeploy API untuk mengubah nama aplikasi.

Untuk melihat daftar nama aplikasi, gunakan AWS CLI untuk memanggil perintah daftar-aplikasi.

Untuk informasi tentang penggunaan AWS CLI untuk mengubah nama aplikasi, lihat perbaruan-aplikasi.

Untuk informasi tentang penggunaan CodeDeploy API untuk mengubah nama aplikasi, lihat API_UpdateApplication.

Hapus aplikasi di CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, tindakan AWS CLI, atau CodeDeploy API untuk menghapus aplikasi. Untuk informasi tentang penggunaan tindakan CodeDeploy API, lihat DeleteApplication.

> **Warning**
>
> Menghapus aplikasi menghapus informasi tentang aplikasi dari CodeDeploy sistem, termasuk semua informasi grup penyebaran terkait dan rincian penyebaran. Menghapus aplikasi yang dibuat untuk EC2/On-Premises penerapan tidak menghapus revisi aplikasi apa pun dari instance dan juga tidak menghapus revisi dari bucket Amazon S3. Menghapus aplikasi yang

dibuat untuk EC2/On-Premises penerapan tidak menghentikan instans Amazon EC2 atau membatalkan pendaftaran instans lokal apa pun. Tindakan ini tidak dapat dibatalkan.

Topik

- Hapus aplikasi (konsol)
- Hapus aplikasi (AWS CLI)

Hapus aplikasi (konsol)

Untuk menggunakan CodeDeploy konsol untuk menghapus aplikasi:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dalam daftar aplikasi, pilih aplikasi yang ingin Anda hapus.

   Sebuah halaman muncul berisi rincian tentang aplikasi.
4. Pilih Hapus aplikasi di kanan atas.
5. Saat diminta, masukkan **`delete`** untuk mengonfirmasi bahwa Anda ingin menghapus aplikasi, lalu pilih Hapus.

Hapus aplikasi (AWS CLI)

Untuk menggunakan AWS CLI untuk menghapus aplikasi, panggil perintah hapus-aplikasi, tentukan nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.

Bekerja dengan grup penyebaran di CodeDeploy

Anda dapat menentukan satu atau beberapa grup penyebaran untuk CodeDeploy aplikasi. Setiap penerapan aplikasi menggunakan salah satu grup penyebarannya. Grup penyebaran berisi pengaturan dan konfigurasi yang digunakan selama penerapan. Sebagian besar pengaturan grup penyebaran bergantung pada platform komputasi yang digunakan oleh aplikasi Anda. Beberapa pengaturan, seperti rollback, trigger, dan alarm dapat dikonfigurasi untuk grup penerapan untuk platform komputasi apa pun.

Grup deployment dalam deployment platform komputasi Amazon ECS

Dalam penerapan Amazon ECS, grup penyebaran menentukan layanan Amazon ECS, penyeimbang beban, pendengar pengujian opsional, dan dua grup target. Ini juga menentukan kapan harus mengalihkan lalu lintas ke set tugas pengganti dan kapan harus menghentikan set tugas asli dan aplikasi Amazon ECS setelah penerapan berhasil.

Grup penyebaran di AWS Lambda komputasi penyebaran platform

Dalam penyebaran AWS Lambda, grup penyebaran mendefinisikan satu set konfigurasi CodeDeploy untuk penerapan fungsi di masa mendatang. AWS Lambda Misalnya, grup penyebaran menentukan cara merutekan lalu lintas ke versi baru fungsi Lambda. Ini juga mungkin menentukan alarm dan rollback. Penerapan tunggal dalam grup penyebaran AWS Lambda dapat mengganti satu atau beberapa konfigurasi grup.

Grup penyebaran dalam penerapan EC2/On-Premises Compute Platform

Dalam EC2/On-Premises penerapan, grup penyebaran adalah sekumpulan instance individual yang ditargetkan untuk penerapan. Grup penerapan berisi instans yang ditandai secara individual, instans Amazon EC2 di grup Penskalaan Otomatis Amazon EC2, atau keduanya.

Dalam penerapan di tempat, instance dalam grup penyebaran diperbarui dengan revisi aplikasi terbaru.

Dalam blue/green penerapan, lalu lintas dialihkan dari satu set instance ke yang lain dengan membatalkan pendaftaran instans asli dari satu atau lebih penyeimbang beban dan mendaftarkan serangkaian instance pengganti yang biasanya memiliki revisi aplikasi terbaru yang sudah diinstal.

Anda dapat mengaitkan lebih dari satu grup penyebaran dengan aplikasi di CodeDeploy. Hal ini memungkinkan untuk menyebarkan revisi aplikasi ke set instance yang berbeda pada waktu yang berbeda. Misalnya, Anda dapat menggunakan satu grup penyebaran untuk menyebarkan revisi aplikasi ke sekumpulan instance yang diberi tag `Test` di mana Anda memastikan kualitas kode. Selanjutnya, Anda menerapkan revisi aplikasi yang sama ke grup penerapan dengan instance yang ditandai `Staging` untuk verifikasi tambahan. Terakhir, ketika Anda siap untuk merilis aplikasi terbaru ke pelanggan, Anda menyebarkan ke grup penyebaran yang menyertakan instance yang diberi tag. `Production`

Anda juga dapat menggunakan beberapa grup tag untuk lebih menyempurnakan kriteria untuk instance yang disertakan dalam grup penerapan. Untuk informasi, lihat Tagging Instances for Deployments.

Saat Anda menggunakan CodeDeploy konsol untuk membuat aplikasi, Anda mengonfigurasi grup penyebaran pertamanya secara bersamaan. Bila Anda menggunakan AWS CLI untuk membuat aplikasi, Anda membuat grup penyebaran pertama dalam langkah terpisah.

Untuk melihat daftar grup penyebaran yang telah dikaitkan dengan AWS akun Anda, lihatLihat detail grup penerapan dengan CodeDeploy.

Untuk informasi tentang tag instans Amazon EC2, lihat Bekerja dengan tag menggunakan konsol. Untuk informasi tentang instance lokal, lihat. Working with On-Premises Instances Untuk informasi tentang Amazon EC2 Auto Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling Scaling, lihat.

Topik

- the section called “Buat grup penyebaran”
- the section called “Lihat detail grup penyebaran”
- the section called “Ubah setelan grup penerapan”
- the section called “Mengonfigurasi opsi lanjutan untuk grup deployment”
- the section called “Menghapus grup penyebaran”

Buat grup penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, CodeDeploy API, atau CloudFormation templat untuk membuat grup penerapan. AWS CLI Untuk informasi tentang menggunakan CloudFormation templat untuk membuat grup penerapan, lihatCloudFormation template untuk CodeDeploy referensi.

Saat Anda menggunakan CodeDeploy konsol untuk membuat aplikasi, Anda mengonfigurasi grup penyebaran pertamanya secara bersamaan. Bila Anda menggunakan AWS CLI untuk membuat aplikasi, Anda membuat grup penyebaran pertama dalam langkah terpisah.

Sebagai bagian dari membuat grup penyebaran, Anda harus menentukan peran layanan. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Topik

- Buat grup penerapan untuk penerapan di tempat (konsol)
- Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol)
- Membuat grup penerapan untuk penyebaran Amazon ECS (konsol)
- Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2
- Siapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS
- Buat grup penyebaran (CLI)

Buat grup penerapan untuk penerapan di tempat (konsol)

Untuk menggunakan CodeDeploy konsol untuk membuat grup penerapan untuk penerapan di tempat:

> **Warning**
>
> Jangan ikuti langkah-langkah ini jika:
>
> - Anda belum menyiapkan instance Anda untuk digunakan dalam CodeDeploy penerapan pertama aplikasi. Untuk menyiapkan instans Anda, ikuti petunjuknyaBekerja dengan instance untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.

- Anda ingin membuat grup penyebaran yang menggunakan konfigurasi penerapan kustom, tetapi Anda belum membuat konfigurasi penerapan. Ikuti instruksi diCreate a Deployment Configuration, dan kemudian ikuti langkah-langkah dalam topik ini.
- Anda tidak memiliki peran layanan yang mempercayai CodeDeploy , setidaknya, kepercayaan dan izin yang dijelaskan dalam. Langkah 2: Buat peran layanan untuk CodeDeploy Untuk membuat dan mengonfigurasi peran layanan, ikuti petunjuk diLangkah 2: Buat peran layanan untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.
- Anda ingin memilih Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing untuk penyebaran di tempat, tetapi belum membuatnya.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pada halaman Aplikasi, pilih nama aplikasi yang ingin Anda buat grup penyebaran.
4. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group.
5. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran; tag, nama grup Amazon EC2 Auto Scaling, atau keduanya; dan konfigurasi penerapan), tentukan setelan tersebut di halaman ini. Meskipun grup penyebaran baru ini dan grup penyebaran yang ada memiliki nama yang sama, CodeDeploy memperlakukannya sebagai grup penyebaran terpisah, karena masing-masing terkait dengan aplikasi terpisah.

6. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke instance target Anda.
7. Dalam jenis Deployment, pilih In-place.

8. Dalam konfigurasi Lingkungan, lakukan hal berikut:

   a. Jika Anda ingin menerapkan aplikasi ke grup Penskalaan Otomatis Amazon EC2, pilih grup Amazon EC2 Auto Scaling, lalu pilih nama grup Amazon EC2 Auto Scaling untuk menerapkan revisi aplikasi. Saat instans Amazon EC2 baru diluncurkan sebagai bagian dari CodeDeploy grup Penskalaan Otomatis Amazon EC2, Anda dapat menerapkan revisi Anda ke instans baru secara otomatis. Anda dapat menambahkan hingga 10 grup Amazon EC2 Auto Scaling ke grup penerapan. Untuk informasi selengkapnya, lihat Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling.

   b. Jika Anda memilih grup Amazon EC2 Auto Scaling, pilih secara opsional Tambahkan hook penghentian ke grup Auto Scaling agar menginstal hook CodeDeploy penghentian ke grup Auto Scaling saat Anda membuat atau memperbarui grup penerapan. Ketika hook ini diinstal, CodeDeploy akan melakukan penyebaran terminasi. Untuk informasi selengkapnya, lihat Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling.

   c. Jika Anda ingin menandai instans Anda, pilih instans atau On-premises instans Amazon EC2. Di bidang Kunci dan Nilai, masukkan nilai pasangan kunci-nilai yang Anda gunakan untuk menandai instance. Anda dapat menandai hingga 10 pasangan nilai kunci dalam satu grup tag.

      i. Anda dapat menggunakan wildcard di bidang Nilai untuk mengidentifikasi semua instance yang ditandai dalam pola tertentu, seperti instans Amazon EC2 yang serupa, pusat biaya, dan nama grup, dan sebagainya. Misalnya, jika Anda memilih Nama di bidang Kunci dan masukkan **`GRP-*a`** di bidang Nilai, CodeDeploy mengidentifikasi semua instance yang sesuai dengan pola tersebut, seperti, **`GRP-1aGRP-2a`**, dan. **`GRP-XYZ-a`**

      ii. Bidang Nilai peka huruf besar/kecil.

      iii. Untuk menghapus pasangan kunci-nilai dari daftar, pilih ikon hapus.

      Saat CodeDeploy menemukan instance yang cocok dengan setiap pasangan nilai kunci yang ditentukan atau nama grup Amazon EC2 Auto Scaling, ini menampilkan jumlah instance yang cocok. Untuk melihat informasi selengkapnya tentang instans, klik nomornya.

      Jika Anda ingin menyaring kriteria untuk instance yang disebarkan, pilih Tambahkan grup tag untuk membuat grup tag. Anda dapat membuat hingga tiga grup tag dengan masing-masing hingga 10 pasangan nilai kunci. Bila Anda menggunakan beberapa grup tag dalam grup penyebaran, hanya instance yang diidentifikasi oleh semua grup tag yang disertakan

dalam grup penyebaran. Itu berarti sebuah instance harus cocok dengan setidaknya satu tag di setiap grup yang akan disertakan dalam grup penerapan.

Untuk informasi tentang menggunakan grup tag untuk menyempurnakan grup penerapan Anda, lihat. Tagging Instances for Deployments

9. Dalam konfigurasi Agen dengan Systems Manager, tentukan cara Anda ingin menginstal dan memperbarui CodeDeploy agen pada instance di grup penerapan Anda. Untuk informasi lebih lanjut tentang CodeDeploy agen, lihat Bekerja dengan CodeDeploy agen. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu Systems Manager?

   a. Jangan Pernah: Lewati konfigurasi CodeDeploy instalasi dengan Systems Manager. Instans harus memiliki agen yang diinstal untuk digunakan dalam penerapan, jadi hanya pilih opsi ini jika Anda akan menginstal CodeDeploy agen dengan cara lain.

   b. Hanya sekali: Systems Manager akan menginstal CodeDeploy agen satu kali pada setiap instance di grup penyebaran Anda.

   c. Sekarang dan jadwalkan pembaruan: Systems Manager akan membuat asosiasi dengan State Manager yang menginstal CodeDeploy agen pada jadwal yang Anda konfigurasikan. Untuk informasi selengkapnya tentang Manajer Negara dan asosiasi, lihat Tentang Manajer Negara.

10. Dalam konfigurasi Deployment, pilih konfigurasi penerapan untuk mengontrol kecepatan penggunaan instance, seperti satu per satu atau sekaligus. Untuk informasi selengkapnya tentang konfigurasi penerapan, lihat. Bekerja dengan konfigurasi penerapan di CodeDeploy

11. (Opsional) Di Load balancer, pilih Aktifkan load balancing, lalu dari daftar, pilih Classic Load Balancers, grup target Application Load Balancer, dan grup target Network Load Balancer untuk mengelola lalu lintas ke instans selama penyebaran. CodeDeploy Anda dapat memilih hingga 10 Classic Load Balancer dan 10 grup target, dengan total 20 item. Pastikan instans Amazon EC2 yang ingin Anda gunakan terdaftar dengan penyeimbang beban yang dipilih (Classic Load Balancer) atau grup target (Application Load Balancers dan Network Load Balancer).

    Selama penerapan, instance asli dideregistrasi dari penyeimbang beban dan grup target yang dipilih untuk mencegah lalu lintas dialihkan ke instance ini selama penerapan. Saat penerapan selesai, setiap instance didaftarkan ulang dengan semua Classic Load Balancer dan grup target yang dipilih.

    Untuk informasi selengkapnya tentang penyeimbang beban untuk CodeDeploy penerapan, lihat. Integrating CodeDeploy with Elastic Load Balancing

> **Warning**
>
> Jika Anda mengonfigurasi grup Auto Scaling dan penyeimbang beban Elastic Load Balancing di grup penerapan ini, dan Anda ingin melampirkan load balancer ke grup Auto Scaling, sebaiknya lengkapi lampiran ini sebelum membuat deployment dari grup deployment ini. CodeDeploy Mencoba menyelesaikan lampiran setelah membuat penerapan dapat menyebabkan semua instance dideregistrasi dari penyeimbang beban secara tidak terduga.

12. (Opsional) Perluas Lanjutan dan konfigurasikan opsi apa pun yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, opsi Auto Scaling, atau rollback otomatis.

    Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup deployment.

13. Pilih Buat grup penyebaran.

Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol)

Untuk menggunakan CodeDeploy konsol untuk membuat grup penyebaran untuk blue/green penerapan:

> **Warning**
>
> Jangan ikuti langkah-langkah ini jika:
>
> - Anda tidak memiliki instance dengan CodeDeploy agen yang diinstal yang ingin Anda ganti selama proses blue/green penerapan. Untuk menyiapkan instans Anda, ikuti petunjuknyaBekerja dengan instance untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.
> - Anda ingin membuat aplikasi yang menggunakan konfigurasi penerapan khusus, tetapi Anda belum membuat konfigurasi penerapan. Ikuti instruksi diCreate a Deployment Configuration, dan kemudian ikuti langkah-langkah dalam topik ini.
> - Anda tidak memiliki peran layanan yang mempercayai CodeDeploy , setidaknya, kepercayaan dan izin yang dijelaskan dalam. Langkah 2: Buat peran layanan untuk

CodeDeploy Untuk membuat dan mengonfigurasi peran layanan, ikuti petunjuk diLangkah 2: Buat peran layanan untuk CodeDeploy, lalu ikuti langkah-langkah dalam topik ini.

- Anda belum membuat Classic Load Balancer atau Application Load Balancer dalam Elastic Load Balancing untuk pendaftaran instans di lingkungan pengganti Anda. Untuk informasi selengkapnya, lihat Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pada halaman Aplikasi, pilih nama aplikasi yang ingin Anda buat grup penyebaran.
4. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group.
5. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran, tag, nama grup Amazon EC2 Auto Scaling, dan konfigurasi penerapan), pilih pengaturan tersebut di halaman ini. Meskipun grup penyebaran baru ini dan grup penyebaran yang ada memiliki nama yang sama, CodeDeploy memperlakukannya sebagai grup penyebaran terpisah, karena mereka terkait dengan aplikasi terpisah.

6. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke instance target Anda.
7. Di Jenis Deployment pilih Blue/green.
8. Dalam konfigurasi Lingkungan, lakukan hal berikut:

   - Pilih metode yang akan digunakan untuk menyediakan instance untuk lingkungan pengganti Anda. Anda memiliki opsi berikut:

- Secara otomatis menyalin grup Amazon EC2 Auto Scaling: membuat CodeDeploy grup Penskalaan Otomatis Amazon EC2 dengan menyalin grup yang Anda tentukan.
- Penyediaan instance secara manual: Anda tidak akan menentukan instance untuk lingkungan pengganti hingga Anda membuat penerapan. Anda harus membuat instance sebelum memulai penerapan. Sebagai gantinya, di sini Anda menentukan instance yang ingin Anda ganti.

- Jika Anda memilih Secara otomatis menyalin grup Amazon EC2 Auto Scaling, pilih secara opsional Tambahkan hook penghentian ke grup Auto Scaling agar menginstal hook CodeDeploy penghentian ke grup Auto Scaling saat Anda membuat atau memperbarui grup penerapan. Ketika hook ini diinstal, CodeDeploy akan melakukan penyebaran terminasi. Untuk informasi selengkapnya, lihat Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling.

9. Dalam konfigurasi Agen dengan Systems Manager, tentukan cara Anda ingin menginstal dan memperbarui CodeDeploy agen pada instance di grup penerapan Anda. Untuk informasi lebih lanjut tentang CodeDeploy agen, lihat Bekerja dengan CodeDeploy agen. Untuk informasi selengkapnya tentang Systems Manager, lihat Apa itu Systems Manager?

   a. Jangan Pernah: Lewati konfigurasi CodeDeploy instalasi dengan Systems Manager. Instans harus memiliki agen yang diinstal untuk digunakan dalam penerapan, jadi hanya pilih opsi ini jika Anda akan menginstal CodeDeploy agen dengan cara lain.

   b. Hanya sekali: Systems Manager akan menginstal CodeDeploy agen satu kali pada setiap instance di grup penyebaran Anda.

   c. Sekarang dan jadwalkan pembaruan: Systems Manager akan membuat asosiasi dengan State Manager yang menginstal CodeDeploy agen pada jadwal yang Anda konfigurasikan. Untuk informasi selengkapnya tentang Manajer Negara dan asosiasi, lihat Tentang Manajer Negara.

10. Tergantung pada pilihan Anda di langkah 8, lakukan salah satu hal berikut:

    - Jika Anda memilih Secara otomatis menyalin grup Amazon EC2 Auto Scaling: Di grup Amazon EC2 Auto Scaling, pilih atau masukkan nama grup Penskalaan Otomatis Amazon EC2 yang ingin Anda gunakan sebagai templat untuk grup Penskalaan Otomatis Amazon EC2 yang dibuat untuk instans di lingkungan pengganti Anda. Jumlah instans yang saat ini sehat di grup Penskalaan Otomatis Amazon EC2 yang Anda pilih dibuat di lingkungan pengganti Anda.
    - Jika Anda memilih Instans penyediaan secara manual: Pilih grup Penskalaan Otomatis Amazon EC2, instans Penskalaan Otomatis Amazon EC2, atau keduanya untuk menentukan instans yang akan ditambahkan ke grup penerapan ini. Masukkan nilai tag Amazon EC2

Auto Scaling atau nama grup Amazon EC2 Auto Scaling untuk mengidentifikasi instance di lingkungan asli Anda (yaitu instans yang ingin Anda ganti atau yang menjalankan revisi aplikasi saat ini).

11. Di Load balancer, pilih Aktifkan load balancing, lalu dari daftar, pilih Classic Load Balancer, grup target Application Load Balancer, dan grup target Network Load Balancer yang ingin Anda daftarkan instans Amazon EC2 pengganti Anda. Setiap instance pengganti akan didaftarkan dengan semua Classic Load Balancer dan grup target yang dipilih. Anda dapat memilih hingga 10 Classic Load Balancer dan 10 grup target, dengan total 20 item.

    Lalu lintas akan dialihkan dari yang asli ke instance pengganti sesuai dengan pengaturan pengalihan rute lalu lintas dan konfigurasi Deployment yang Anda pilih.

    Untuk informasi selengkapnya tentang penyeimbang beban untuk CodeDeploy penerapan, lihat. Integrating CodeDeploy with Elastic Load Balancing

    > ⚠ Warning
    >
    > Jika Anda mengonfigurasi grup Auto Scaling dan penyeimbang beban Elastic Load Balancing di grup penerapan ini, dan Anda ingin melampirkan load balancer ke grup Auto Scaling, sebaiknya lengkapi lampiran ini sebelum membuat deployment dari grup deployment ini. CodeDeploy Mencoba menyelesaikan lampiran setelah membuat penerapan dapat menyebabkan semua instance dideregistrasi dari penyeimbang beban secara tidak terduga.

12. Dalam pengaturan Deployment, tinjau opsi default untuk mengalihkan lalu lintas ke lingkungan pengganti, konfigurasi penerapan mana yang akan digunakan untuk penerapan, dan bagaimana instance di lingkungan asli ditangani setelah penerapan.

    Jika Anda ingin mengubah pengaturan, lanjutkan ke langkah berikutnya. Jika tidak, lewati ke langkah 14.

13. Untuk mengubah pengaturan penyebaran untuk blue/green penyebaran, pilih salah satu pengaturan berikut.

Pengaturan	Opsi
Pengalihan rute lalu lintas	• Rute lalu lintas segera: Segera setelah instance di lingkungan penggantian

Pengaturan	Opsi
	disediakan dan revisi aplikasi terbaru diinstal pada mereka, mereka terdaftar dengan penyeimbang beban yang ditentukan dan grup target secara otomatis, menyebabkan lalu lintas dialihkan ke mereka. Contoh di lingkungan asli kemudian dideregistrasi. • Saya akan memilih apakah akan mengalihkan lalu lintas: Contoh di lingkungan penggantian tidak terdaftar dengan penyeimbang beban dan grup target yang ditentukan kecuali Anda mengubah rute lalu lintas secara manual. Jika waktu tunggu yang Anda tentukan berlalu tanpa lalu lintas dialihkan, status penerapan diubah menjadi Berhenti.
Konfigurasi deployment	Pilih tingkat di mana instance di lingkungan penggantian terdaftar dengan penyeimbang beban dan kelompok sasaran, seperti satu per satu atau sekaligus. **Note** Setelah lalu lintas berhasil dialihkan ke lingkungan pengganti, instance di lingkungan asli dideregistrasi sekaligus tidak peduli konfigurasi penerapan mana yang dipilih. Untuk informasi selengkapnya, lihat Bekerja dengan konfigurasi penerapan di CodeDeploy.

Pengaturan	Opsi
Contoh asli	• Mengakhiri instance asli dalam grup penyebaran: Setelah lalu lintas dialihkan ke lingkungan pengganti, instance yang dideregistrasi dari penyeimbang beban dan grup target dihentikan setelah periode tunggu yang Anda tentukan. Perhatikan hal-hal berikut: • Jika grup Auto Scaling Anda mengaktifkan Perlindungan Penghapusan (`prevent-force-deletion` atau `prevent-all-deletion`), CodeDeploy blue/green penerapan akan menonaktifkan perlindungan ini pada grup Auto Scaling asli (biru) sebelum menghapusnya selama pembersihan penerapan. Grup Auto Scaling pengganti (hijau) akan mempertahankan pengaturan Perlindungan Penghapusan yang sama dengan grup aslinya. • Jaga agar instance asli dalam grup penerapan tetap berjalan: Setelah lalu lintas dialihkan ke lingkungan pengganti, instance yang dideregistrasi dari penyeimbang beban dan grup target tetap berjalan.

14. (Opsional) Di Lanjutan, konfigurasikan opsi yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, opsi Auto Scaling, atau rollback otomatis.

    Untuk informasi tentang menentukan opsi lanjutan dalam grup penerapan, lihat. Mengonfigurasi opsi lanjutan untuk grup deployment

15. Pilih Buat grup penyebaran.

Membuat grup penerapan untuk penyebaran Amazon ECS (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dari tabel Aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran yang ingin Anda edit.
4. Pada halaman aplikasi Anda, dari grup Deployment, pilih nama grup penyebaran yang ingin Anda edit.
5. Pada halaman aplikasi Anda, dari tab Deployment groups, pilih Create deployment group. Untuk informasi selengkapnya tentang apa yang Anda perlukan untuk membuat grup penerapan untuk penyebaran Amazon ECS, lihat. Sebelum memulai deployment Amazon ECS
6. Dalam nama grup Deployment, masukkan nama yang menjelaskan grup penyebaran.

   > **Note**
   >
   > Jika Anda ingin menggunakan pengaturan yang sama yang digunakan dalam grup penyebaran lain (termasuk nama grup penyebaran dan konfigurasi penerapan), pilih pengaturan tersebut di halaman ini. Meskipun grup baru ini dan grup yang ada mungkin memiliki nama yang sama, CodeDeploy memperlakukan mereka sebagai grup penyebaran terpisah, karena masing-masing dikaitkan dengan aplikasi terpisah.

7. Dalam peran Layanan, pilih peran layanan yang memberikan CodeDeploy akses ke Amazon ECS. Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.
8. Dari nama Load balancer, pilih nama penyeimbang beban yang melayani lalu lintas ke layanan Amazon ECS Anda.
9. Dari port pendengar Produksi, pilih port dan protokol untuk pendengar yang menyajikan lalu lintas produksi ke layanan Amazon ECS Anda.
10. (Opsional) Dari port pendengar Uji, pilih port dan protokol pendengar pengujian yang menyajikan lalu lintas ke tugas pengganti yang ditetapkan dalam layanan Amazon ECS Anda selama

penerapan. Anda dapat menentukan satu atau beberapa fungsi Lambda dalam AppSpec file yang berjalan selama hook. `AfterAllowTestTraffic` Fungsi dapat menjalankan tes validasi. Jika tes validasi gagal, rollback penerapan dipicu. Jika tes validasi berhasil, hook berikutnya dalam siklus hidup penerapan, `BeforeAllowTraffic` akan dipicu. Jika port pendengar uji tidak ditentukan, tidak ada yang terjadi selama `AfterAllowTestTraffic` pengait. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan Amazon ECS.

11. Dari nama grup target 1 dan nama Grup target 2, pilih grup target yang digunakan untuk merutekan lalu lintas selama penyebaran Anda. CodeDeploy mengikat satu grup target ke set tugas asli layanan Amazon ECS Anda dan yang lainnya ke set tugas penggantinya. Untuk informasi selengkapnya, lihat Grup Target untuk Penyeimbang Beban Aplikasi Anda.
12. Pilih Rute lalu lintas segera atau Tentukan kapan harus mengalihkan lalu lintas untuk menentukan kapan harus mengalihkan lalu lintas ke layanan Amazon ECS Anda yang diperbarui.

    Jika Anda memilih Reroute lalu lintas segera, maka penerapan secara otomatis mengalihkan lalu lintas setelah set tugas pengganti disediakan.

    Jika Anda memilih Tentukan kapan akan mengubah rute lalu lintas, lalu pilih jumlah hari, jam, dan menit untuk menunggu setelah set tugas pengganti berhasil disediakan. Selama waktu tunggu ini, tes validasi dalam fungsi Lambda yang ditentukan dalam file AppSpec dijalankan. Jika waktu tunggu berakhir sebelum lalu lintas dialihkan, maka status penerapan berubah menjadi. `Stopped`
13. Untuk penghentian revisi Asli, pilih jumlah hari, jam, dan menit untuk menunggu setelah penerapan berhasil sebelum tugas asli yang ditetapkan dalam layanan Amazon ECS Anda dihentikan.
14. (Opsional) Di Advanced, konfigurasikan opsi apa pun yang ingin Anda sertakan dalam penerapan, seperti pemicu notifikasi Amazon SNS, alarm CloudWatch Amazon, atau rollback otomatis.

    Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup deployment.

Siapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2

Sebelum Anda menjalankan blue/green penerapan apa pun, atau penerapan di tempat yang ingin Anda tentukan penyeimbang beban opsional dalam grup penyebaran, Anda harus telah membuat

setidaknya satu Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing. Untuk blue/green penerapan, Anda menggunakan penyeimbang beban tersebut untuk mendaftarkan instance yang membentuk lingkungan pengganti Anda. Instans di lingkungan asli Anda secara opsional dapat didaftarkan dengan penyeimbang beban yang sama ini. Untuk penerapan di tempat, penyeimbang beban digunakan untuk membatalkan pendaftaran instance yang sedang dikerjakan oleh CodeDeploy, dan mendaftarkannya kembali saat pekerjaan selesai.

CodeDeploy mendukung blue/green dan penyebaran di tempat ke instans Amazon EC2 di belakang penyeimbang beban muliple. Misalnya, Anda memiliki 200 instans Amazon EC2, di mana 100 di antaranya terdaftar dengan 2 Classic Load Balancer, dan 100 lainnya terdaftar dengan 4 grup target dalam 2 Application Load Balancer. Dalam skenario ini, CodeDeploy akan memungkinkan Anda untuk melakukan blue/green dan menerapkan di tempat ke semua 200 instance, meskipun mereka tersebar di 2 Classic Load Balancer, 2 Application Load Balancers, dan 4 grup target.

CodeDeploy mendukung hingga 10 Classic Load Balancer dan 10 grup target, dengan total 20 item.

Untuk mengonfigurasi satu atau beberapa Classic Load Balancer, ikuti petunjuk di Tutorial: Buat Classic Load Balancer di Panduan Pengguna untuk Classic Load Balancer. Perhatikan hal-hal berikut:

- Pada Langkah 2: Tentukan Load Balancer, di Create LB Inside, pilih VPC yang sama yang Anda pilih saat Anda membuat instance Anda.
- Pada Langkah 5: Daftarkan Instans EC2 dengan Load Balancer Anda, pilih instans yang saat ini ada di grup penerapan Anda (penerapan di tempat) atau yang telah Anda tetapkan untuk berada di lingkungan asli (penerapan). blue/green
- Pada Langkah 7: Buat dan Verifikasi Load Balancer Anda, catat alamat DNS penyeimbang beban Anda.

  Misalnya, jika Anda menamai penyeimbang bebanmy-load-balancer, alamat DNS Anda akan muncul dalam format seperti. my-load-balancer-1234567890.us-east-2.elb.amazonaws.com

Untuk mengonfigurasi satu atau beberapa Application Load Balancer, ikuti petunjuk di salah satu topik berikut:

- Buat Application Load Balancer
- Tutorial: Membuat Application Load Balancer menggunakan AWS CLI

Untuk mengonfigurasi satu atau beberapa Network Load Balancer, ikuti petunjuk di salah satu topik berikut:

- Membuat Network Load Balancer
- Tutorial: Membuat Network Load Balancer menggunakan AWS CLI

Siapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS

Sebelum menjalankan penerapan menggunakan platform komputasi Amazon ECS, Anda harus membuat Application Load Balancer atau Network Load Balancer, dua grup target, dan satu atau dua pendengar. Topik ini menunjukkan cara membuat Application Load Balancer. Untuk informasi selengkapnya, lihat Sebelum memulai deployment Amazon ECS.

Salah satu grup target mengarahkan lalu lintas ke set tugas asli aplikasi Amazon ECS Anda. Kelompok sasaran lainnya mengarahkan lalu lintas ke set tugas penggantinya. Selama penerapan, CodeDeploy membuat set tugas pengganti dan mengalihkan lalu lintas dari tugas asli yang disetel ke yang baru. CodeDeploy menentukan kelompok target mana yang digunakan untuk setiap set tugas.

Listener digunakan oleh penyeimbang beban Anda untuk mengarahkan lalu lintas ke grup target Anda. Diperlukan satu pendengar produksi. Anda dapat menentukan pendengar pengujian opsional yang mengarahkan lalu lintas ke set tugas pengganti saat menjalankan pengujian validasi.

Penyeimbang beban harus menggunakan VPC dengan dua subnet publik di Availability Zone yang berbeda. Langkah-langkah berikut menunjukkan cara mengonfirmasi VPC default Anda, membuat Application Load Balancer Amazon EC2, dan kemudian membuat dua grup target untuk penyeimbang beban Anda. Untuk informasi selengkapnya, lihat Grup sasaran untuk penyeimbang beban jaringan Anda.

Verifikasi VPC default, subnet publik, dan grup keamanan

Topik ini menunjukkan cara membuat Application Load Balancer Amazon EC2, dua grup target, dan dua port yang dapat digunakan selama penyebaran Amazon ECS. Salah satu port bersifat opsional dan hanya diperlukan jika Anda mengarahkan lalu lintas ke port uji untuk pengujian validasi selama penerapan Anda.

1. Masuk ke Konsol Manajemen AWS dan buka konsol VPC Amazon di. https://console.aws.amazon.com/vpc/

2. Verifikasi VPC default yang akan digunakan. Pada panel navigasi, pilih VPC Anda. Perhatikan VPC mana yang menunjukkan Ya di kolom VPC Default. Ini adalah VPC default Anda. Ini berisi subnet default yang Anda gunakan.
3. Pilih Subnet. Catat ID subnet dari dua subnet yang menunjukkan Ya di kolom subnet Default. Anda menggunakan ID ini saat membuat penyeimbang beban.
4. Pilih setiap subnet, lalu pilih tab Deskripsi. Verifikasi bahwa subnet yang ingin Anda gunakan berada di Availability Zone yang berbeda.
5. Pilih subnet, lalu pilih tab Route Table. Untuk memverifikasi bahwa setiap subnet yang ingin Anda gunakan adalah subnet publik, konfirmasikan bahwa baris dengan tautan ke gateway internet disertakan dalam tabel rute.
6. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
7. Dari panel navigasi, pilih Kelompok Keamanan.
8. Verifikasi grup keamanan yang ingin Anda gunakan tersedia dan catat ID grupnya (misalnya, sg-abcd1234). Anda menggunakan ini ketika Anda membuat penyeimbang beban Anda.

Buat Application Load Balancer Amazon EC2, dua grup target, dan pendengar (konsol)

Untuk menggunakan konsol Amazon EC2 untuk membuat Application Load Balancer Amazon EC2:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
2. Di panel navigasi, pilih Load Balancers.
3. Pilih Buat Penyeimbang Beban.
4. Pilih Application Load Balancer, lalu pilih Create.
5. Di Nama, masukkan nama penyeimbang beban Anda.
6. Dalam Skema, pilih yang menghadap ke internet.
7. Dalam jenis alamat IP, pilih ipv4.
8. (Opsional) Konfigurasikan port listener kedua untuk penyeimbang beban Anda. Anda dapat menjalankan pengujian validasi penerapan menggunakan lalu lintas pengujian yang disajikan ke port ini.

    a. Di bawah Protokol Load Balancer, pilih Tambahkan pendengar.

    b. Di bawah Protokol Load Balancer untuk pendengar kedua, pilih HTTP.

c. Di bawah Port Load Balancer, masukkan. **8080**

9. Di bawah Availability Zones, di VPC, pilih VPC default, lalu pilih dua subnet default yang ingin Anda gunakan.
10. Pilih Selanjutnya: Konfigurasikan Pengaturan Keamanan.
11. Pilih Selanjutnya: Konfigurasikan Grup Keamanan.
12. Pilih Pilih grup keamanan yang ada, pilih grup keamanan default, lalu catat ID-nya.
13. Pilih Selanjutnya: Konfigurasi Perutean.
14. Di Grup target, pilih Grup target baru, dan konfigurasikan grup target pertama Anda:

    a. Di Nama, masukkan nama grup target (misalnya,**target-group-1**).

    b. Pada tipe Target, pilih IP.

    c. Dalam Protokol pilih HTTP. Di Pelabuhan, masukkan**80**.

    d. Pilih Selanjutnya: Daftarkan Target.
15. Pilih Berikutnya: Tinjau, lalu pilih Buat.

Untuk membuat grup target kedua untuk penyeimbang beban Anda

1. Setelah penyeimbang beban Anda disediakan, buka konsol Amazon EC2. Di panel navigasi, pilih Target Groups.
2. PilihBuat grup target.
3. Di Nama, masukkan nama grup target (misalnya,**target-group-2**).
4. Pada tipe Target, pilih IP.
5. Dalam Protokol pilih HTTP. Di Pelabuhan, masukkan**80**.
6. Di VPC, pilih VPC default.
7. Pilih Buat.

> **Note**
>
> Anda harus memiliki dua grup target yang dibuat untuk penyeimbang beban agar penerapan Amazon ECS dapat berjalan. Anda menggunakan ARN dari salah satu grup target Anda saat membuat layanan Amazon ECS Anda. Untuk informasi selengkapnya, lihat Langkah 4: Membuat layanan Amazon ECS di Panduan Pengguna Amazon ECS.

Buat Application Load Balancer Amazon EC2, dua grup target, dan pendengar (CLI)

Untuk membuat Application Load Balancer menggunakan: AWS CLI

1. Gunakan perintah create-load-balancer untuk membuat Application Load Balancer. Tentukan dua subnet yang tidak berada di Availability Zone dan grup keamanan yang sama.

   ```
   aws elbv2 create-load-balancer --name bluegreen-alb \
   --subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
   region us-east-1
   ```

   Outputnya mencakup Nama Sumber Daya Amazon (ARN) dari penyeimbang beban, dalam format berikut:

   ```
   arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
   e5ba62739c16e642
   ```

2. Gunakan perintah create-target-group untuk membuat grup target pertama Anda. CodeDeploy merutekan lalu lintas grup target ini ke tugas asli atau tugas pengganti yang ditetapkan dalam layanan Anda.

   ```
   aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
   --target-type ip --vpc-id vpc-abcd1234 --region us-east-1
   ```

   Outputnya mencakup ARN dari kelompok target pertama, dalam format berikut:

   ```
   arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
   bluegreentarget1/209a844cd01825a4
   ```

3. Gunakan perintah create-target-group untuk membuat grup target kedua Anda. CodeDeploy merutekan lalu lintas grup target ke set tugas yang tidak dilayani oleh grup target pertama Anda. Misalnya, jika grup target pertama Anda merutekan lalu lintas ke set tugas asli, grup target ini merutekan lalu lintas ke set tugas pengganti.

   ```
   aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
   --target-type ip --vpc-id vpc-abcd1234 --region us-east-1
   ```

   Outputnya mencakup ARN dari kelompok target kedua, dalam format berikut:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Gunakan perintah create-listener untuk membuat listener dengan aturan default yang meneruskan lalu lintas produksi ke port 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
```

Outputnya mencakup ARN pendengar, dalam format berikut:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. (Opsional) Gunakan perintah create-listener untuk membuat pendengar kedua dengan aturan default yang meneruskan lalu lintas pengujian ke port 8080. Anda dapat menjalankan pengujian validasi penerapan menggunakan lalu lintas pengujian yang disajikan port ini.

```
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
```

Outputnya mencakup ARN pendengar, dalam format berikut:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

Buat grup penyebaran (CLI)

Untuk menggunakan AWS CLI untuk membuat grup penyebaran, panggil perintah create-deployment-group, dengan menentukan:

- Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama untuk grup deployment. Grup penyebaran dengan nama ini dibuat untuk aplikasi yang ditentukan. Grup penyebaran hanya dapat dikaitkan dengan satu aplikasi.
- Informasi tentang tag, grup tag, atau nama grup Amazon EC2 Auto Scaling yang mengidentifikasi instans yang akan disertakan dalam grup penerapan.
- Pengidentifikasi Amazon Resource Name (ARN) dari peran layanan yang memungkinkan CodeDeploy untuk bertindak atas nama AWS akun Anda saat berinteraksi dengan layanan lain. AWS Untuk mendapatkan peran layanan ARN, lihat. Dapatkan peran layanan ARN (CLI) Untuk informasi selengkapnya tentang peran layanan, lihat Istilah dan konsep peran di Panduan Pengguna IAM.
- Informasi tentang jenis penyebaran, baik di tempat atau blue/green, untuk dikaitkan dengan grup penyebaran.
- (Opsional) Nama konfigurasi penerapan yang ada. Untuk melihat daftar konfigurasi penerapan, lihat. View Deployment Configuration Details Jika tidak ditentukan, CodeDeploy menggunakan konfigurasi penerapan default.
- (Opsional) Perintah untuk membuat pemicu yang mendorong pemberitahuan tentang penerapan dan kejadian instance kepada mereka yang berlangganan topik Layanan Pemberitahuan Sederhana Amazon. Untuk informasi selengkapnya, lihat Monitoring Deployments with Amazon SNS Event Notifications.
- (Opsional) Perintah untuk menambahkan CloudWatch alarm yang ada ke grup penerapan yang diaktifkan jika metrik yang ditentukan dalam alarm jatuh di bawah atau melebihi ambang batas yang ditentukan.
- (Opsional) Perintah untuk penerapan untuk memutar kembali ke revisi baik terakhir yang diketahui ketika penerapan gagal atau CloudWatch alarm diaktifkan.
- (Opsional) Perintah untuk penerapan untuk menghasilkan kait peristiwa siklus hidup selama acara penskalaan Auto Scaling. Untuk informasi selengkapnya, lihat Cara Kerja Amazon EC2 Auto Scaling CodeDeploy.
- Untuk penerapan di tempat:

  - (Opsional) Nama-nama Classic Load Balancer, Application Load Balancers, atau Network Load Balancer di Elastic Load Balancing yang mengelola lalu lintas ke instans selama proses penyebaran.
- Untuk blue/green penerapan:
  - Konfigurasi proses blue/green penyebaran:
    - Bagaimana instance baru di lingkungan penggantian disediakan.
    - Apakah akan mengalihkan lalu lintas ke lingkungan pengganti segera atau menunggu periode tertentu agar lalu lintas dialihkan secara manual.
    - Apakah instance di lingkungan asli harus dihentikan.
  - Nama-nama Classic Load Balancer, Application Load Balancers, atau Network Load Balancers dalam Elastic Load Balancing yang akan digunakan untuk instans yang terdaftar di lingkungan pengganti.

> **Warning**
>
> Jika Anda mengonfigurasi grup Auto Scaling dan penyeimbang beban Elastic Load Balancing di grup penerapan, dan Anda ingin melampirkan penyeimbang beban ke grup Auto Scaling, sebaiknya selesaikan lampiran ini sebelum membuat deployment dari grup deployment ini. CodeDeploy Mencoba menyelesaikan lampiran setelah membuat penerapan dapat menyebabkan semua instance dideregistrasi dari penyeimbang beban secara tidak terduga.

Lihat detail grup penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang semua grup penerapan yang terkait dengan aplikasi.

Topik

- Lihat detail grup penerapan (konsol)
- Lihat detail grup penerapan (CLI)

Lihat detail grup penerapan (konsol)

Untuk menggunakan CodeDeploy konsol untuk melihat detail grup penerapan:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pada halaman Aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

4. Untuk melihat detail tentang grup penyebaran individual, pada tab Grup penyebaran, pilih nama grup penyebaran.

Lihat detail grup penerapan (CLI)

Untuk menggunakan detail grup AWS CLI untuk melihat penyebaran, panggil `get-deployment-group` perintah atau `list-deployment-groups` perintah.

Untuk melihat detail tentang grup penerapan tunggal, panggil perintah get-deployment-group, dengan menentukan:

- Nama aplikasi yang terkait dengan grup penyebaran. Untuk mendapatkan nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran. Untuk mendapatkan nama grup penyebaran, panggil perintah list-deployment-groups.

Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups, dengan menentukan nama aplikasi yang terkait dengan grup penyebaran. Untuk mendapatkan nama aplikasi, panggil perintah daftar-aplikasi.

Ubah pengaturan grup penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk mengubah pengaturan grup penerapan.

> **Warning**
>
> Jangan gunakan langkah-langkah ini jika Anda ingin grup penyebaran menggunakan grup penyebaran kustom yang belum dibuat. Sebagai gantinya, ikuti instruksi diCreate a Deployment Configuration, dan kemudian kembali ke topik ini. Jangan gunakan langkah-langkah ini jika Anda ingin grup penyebaran menggunakan peran layanan yang berbeda dan belum dibuat. Peran layanan harus mempercayai CodeDeploy , setidaknya, izin yang dijelaskan dalamLangkah 2: Buat peran layanan untuk CodeDeploy. Untuk membuat dan mengonfigurasi peran layanan dengan izin yang benar, ikuti petunjuk diLangkah 2: Buat peran layanan untuk CodeDeploy, lalu kembali ke topik ini.

Topik

- Ubah pengaturan grup penerapan (konsol)
- Ubah pengaturan grup penyebaran (CLI)

Ubah pengaturan grup penerapan (konsol)

Untuk menggunakan CodeDeploy konsol untuk mengubah pengaturan grup penyebaran:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dalam daftar aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran yang ingin Anda ubah.

> **Note**
>
> Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

4. Pilih tab Grup penyebaran, lalu pilih nama grup penyebaran yang ingin Anda ubah.
5. Pada halaman grup Deployment, pilih Edit.
6. Edit opsi grup penyebaran sesuai kebutuhan.

   Untuk informasi tentang komponen grup penerapan, lihatBuat grup penerapan dengan CodeDeploy.
7. Pilih Simpan perubahan.

Ubah pengaturan grup penyebaran (CLI)

Untuk menggunakan pengaturan grup AWS CLI untuk mengubah penerapan, panggil perintah update-deployment-group, dengan menentukan:

- Penyebaran for EC2/On-Premises dan AWS Lambda:
  - Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
  - Nama grup penyebaran saat ini. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.
  - (Opsional) Nama grup penyebaran yang berbeda.
  - (Opsional) Nama Sumber Daya Amazon (ARN) berbeda yang sesuai dengan peran layanan yang memungkinkan CodeDeploy untuk bertindak atas nama AWS akun Anda saat berinteraksi dengan layanan lain. AWS Untuk mendapatkan peran layanan ARN, lihat. Dapatkan peran layanan ARN (CLI) Untuk informasi selengkapnya tentang peran layanan, lihat Istilah dan konsep peran di Panduan Pengguna IAM.
  - (Opsional) Nama konfigurasi penerapan. Untuk melihat daftar konfigurasi penerapan, lihat. View Deployment Configuration Details (Jika tidak ditentukan, CodeDeploy gunakan konfigurasi penerapan default.)

  - (Opsional) Perintah untuk menambahkan satu atau beberapa CloudWatch alarm yang ada ke grup penerapan yang diaktifkan jika metrik yang ditentukan dalam alarm jatuh di bawah atau melebihi ambang batas yang ditentukan.
  - (Opsional) Perintah untuk penerapan untuk memutar kembali ke revisi baik terakhir yang diketahui ketika penerapan gagal atau CloudWatch alarm diaktifkan.
  - (Opsional) Perintah untuk penerapan untuk menghasilkan kait peristiwa siklus hidup selama acara penskalaan Auto Scaling. Untuk informasi selengkapnya, lihat Cara Kerja Amazon EC2 Auto Scaling CodeDeploy.
  - (Opsional) Perintah untuk membuat atau memperbarui pemicu yang memublikasikan topik di Amazon Simple Notification Service, sehingga pelanggan topik tersebut menerima pemberitahuan tentang penyebaran dan kejadian instance di grup penerapan ini. Untuk informasi, lihat Monitoring Deployments with Amazon SNS Event Notifications.
- Hanya untuk EC2/On-Premises penerapan:
  - (Opsional) Tag pengganti atau grup tag yang secara unik mengidentifikasi instance yang akan disertakan dalam grup penyebaran.
  - (Opsional) Nama grup Penskalaan Otomatis Amazon EC2 pengganti yang akan ditambahkan ke grup penerapan.
- Hanya untuk penerapan Amazon ECS:
  - Layanan Amazon ECS untuk digunakan.
  - Informasi penyeimbang beban, termasuk Application Load Balancer atau Network Load Balancer, grup target yang diperlukan untuk penyebaran Amazon ECS, serta informasi pendengar uji produksi dan opsional.

Mengonfigurasi opsi lanjutan untuk grup deployment

Saat membuat atau memperbarui grup penyebaran, Anda dapat mengonfigurasi sejumlah opsi untuk memberikan kontrol dan pengawasan lebih lanjut atas penerapan untuk grup penyebaran tersebut.

Gunakan informasi di halaman ini untuk membantu Anda mengonfigurasi opsi lanjutan saat Anda bekerja dengan grup penerapan dalam topik berikut:

- Buat aplikasi dengan CodeDeploy
- Buat grup penerapan dengan CodeDeploy
- Ubah pengaturan grup penerapan dengan CodeDeploy

Pemicu notifikasi Amazon SNS: Anda dapat menambahkan pemicu ke grup CodeDeploy penerapan untuk menerima pemberitahuan tentang peristiwa yang terkait dengan penerapan di grup penerapan tersebut. Pemberitahuan ini dikirim ke penerima yang berlangganan topik Amazon SNS yang telah Anda jadikan bagian dari tindakan pemicu.

Anda harus sudah menyiapkan topik Amazon SNS yang akan ditunjuk oleh pemicu ini, dan CodeDeploy harus memiliki izin untuk mempublikasikan ke topik dari grup penerapan ini. Jika Anda belum menyelesaikan langkah-langkah penyiapan ini, Anda dapat menambahkan pemicu ke grup penerapan nanti.

Jika Anda ingin membuat pemicu sekarang untuk menerima pemberitahuan tentang peristiwa penerapan di grup penyebaran untuk aplikasi ini, pilih Buat pemicu.

Jika penerapan Anda ke instans Amazon EC2, Anda dapat membuat notifikasi untuk dan menerima pemberitahuan tentang instans.

Untuk informasi selengkapnya, lihat Monitoring Deployments with Amazon SNS Event Notifications.

CloudWatch Alarm Amazon: Anda dapat membuat CloudWatch alarm yang mengawasi satu metrik selama periode waktu yang Anda tentukan dan melakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Untuk penerapan Amazon EC2, Anda dapat membuat alarm untuk instans atau grup Penskalaan Otomatis Amazon EC2 yang Anda gunakan dalam operasi Anda. CodeDeploy Untuk penyebaran AWS Lambda dan Amazon ECS, Anda dapat membuat alarm untuk kesalahan dalam fungsi Lambda.

Anda dapat mengonfigurasi penerapan untuk berhenti ketika CloudWatch alarm Amazon mendeteksi bahwa metrik telah jatuh di bawah atau melebihi ambang batas yang ditentukan.

Anda harus sudah membuat alarm CloudWatch sebelum Anda dapat menambahkannya ke grup penyebaran.

1. Untuk menambahkan pemantauan alarm ke grup penyebaran, di Alarm, pilih Tambahkan alarm.
2. Masukkan nama CloudWatch alarm yang telah Anda atur untuk memantau penyebaran ini.

   Anda harus memasukkan CloudWatch alarm persis seperti yang dibuat CloudWatch. Untuk melihat daftar alarm, buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/, lalu pilih ALARM.

Opsi tambahan:

- Jika Anda ingin penerapan dilanjutkan tanpa memperhitungkan alarm akun yang telah Anda tambahkan, pilih Abaikan konfigurasi alarm.

  Pilihan ini berguna ketika Anda ingin menonaktifkan sementara pemantauan alarm untuk grup penyebaran tanpa harus menambahkan alarm yang sama lagi nanti.

- (Opsional) Jika Anda ingin penerapan dilanjutkan jika tidak dapat mengambil status alarm dari Amazon CloudWatch, pilih Lanjutkan penerapan meskipun status alarm tidak tersedia. CodeDeploy

> **Note**
>
> Opsi ini sesuai ignorePollAlarmFailure dengan AlarmConfigurationobjek di CodeDeploy API.

Untuk informasi selengkapnya, lihat Memantau penyebaran dengan CloudWatch alarm di CodeDeploy.

Rollback otomatis: Anda dapat mengonfigurasi grup penyebaran atau penerapan untuk memutar kembali secara otomatis saat penerapan gagal atau saat ambang batas pemantauan yang Anda tentukan terpenuhi. Dalam hal ini, versi baik terakhir yang diketahui dari revisi aplikasi digunakan. Anda dapat mengonfigurasi pengaturan opsional untuk grup penyebaran saat Anda menggunakan konsol untuk membuat aplikasi, membuat grup penyebaran, atau memperbarui grup penerapan. Saat membuat penerapan baru, Anda juga dapat memilih untuk mengganti konfigurasi rollback otomatis yang ditentukan untuk grup penerapan.

- Anda dapat mengaktifkan penerapan untuk memutar kembali ke revisi terbaru yang diketahui saat terjadi kesalahan dengan memilih salah satu atau kedua hal berikut:

  - Gulung kembali saat penerapan gagal. CodeDeploy akan menerapkan kembali revisi bagus terakhir yang diketahui sebagai penerapan baru.
  - Putar kembali saat ambang alarm terpenuhi. Jika Anda menambahkan alarm ke aplikasi ini pada langkah sebelumnya, CodeDeploy akan menyebarkan revisi baik terakhir yang diketahui ketika satu atau lebih dari alarm yang ditentukan diaktifkan.

> **Note**
>
> Untuk mengabaikan konfigurasi rollback sementara, pilih Nonaktifkan rollback. Pilihan ini berguna ketika Anda ingin menonaktifkan sementara rollback otomatis tanpa harus mengatur konfigurasi yang sama lagi nanti.

Untuk informasi selengkapnya, lihat Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.

Pembaruan otomatis ke instans yang sudah ketinggalan zaman: Dalam keadaan tertentu, CodeDeploy dapat menerapkan revisi aplikasi yang sudah ketinggalan zaman ke instans Amazon EC2 Anda. Misalnya, jika instans EC2 Anda diluncurkan ke grup Auto Scaling (ASG) saat CodeDeploy penerapan sedang berlangsung, instans tersebut menerima revisi lama aplikasi Anda, bukan yang terbaru. Untuk memperbarui instans tersebut, CodeDeploy secara otomatis memulai penerapan tindak lanjut (segera setelah yang pertama) untuk memperbarui instance yang sudah ketinggalan zaman. Jika Anda ingin mengubah perilaku default ini sehingga instance EC2 yang sudah ketinggalan zaman dibiarkan pada revisi yang lebih lama, Anda dapat melakukannya melalui CodeDeploy API atau (AWS Command Line Interface CLI).

Untuk mengonfigurasi pembaruan otomatis instance usang melalui API, sertakan parameter `outdatedInstancesStrategy` permintaan dalam tindakan `UpdateDeploymentGroup` atau`CreateDeploymentGroup`. Untuk detailnya, lihat Referensi AWS CodeDeploy API.

Untuk mengkonfigurasi pembaruan otomatis melalui AWS CLI, gunakan salah satu perintah berikut:

```
aws deploy update-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

Atau...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy UPDATE|IGNORE
```

... where *`arguments`* diganti dengan argumen yang diperlukan untuk penerapan Anda, dan *`UPDATE|IGNORE`* diganti dengan baik `UPDATE` untuk mengaktifkan pembaruan otomatis, atau `IGNORE` untuk menonaktifkannya.

Contoh:

```
aws deploy update-deployment-group --application-name "MyApp" --current-
deployment-group-name "MyDG" --region us-east-1 --outdated-instances-
strategy IGNORE
```

Untuk detail tentang AWS CLI perintah ini, lihat Referensi AWS CLI Perintah.

Hapus grup penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk menghapus grup penerapan yang terkait dengan AWS akun Anda.

> **Warning**
>
> Jika Anda menghapus grup penyebaran, semua detail yang terkait dengan grup penyebaran itu juga akan dihapus. CodeDeploy Instance yang digunakan dalam grup penyebaran akan tetap tidak berubah. Tindakan ini tidak dapat dibatalkan.

Topik

- Menghapus grup penyebaran (konsol)
- Hapus grup penyebaran (CLI)

Menghapus grup penyebaran (konsol)

Untuk menggunakan CodeDeploy konsol untuk menghapus grup penyebaran:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Dalam daftar aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran.

4. Pada halaman Detail aplikasi, pada tab Grup Deployment, pilih nama grup penyebaran yang ingin Anda hapus.
5. Pada halaman Detail Deployment, pilih Hapus.
6. Saat diminta, ketik nama grup penyebaran untuk mengonfirmasi bahwa Anda ingin menghapusnya, lalu pilih Hapus.

Hapus grup penyebaran (CLI)

Untuk menggunakan AWS CLI untuk menghapus grup penyebaran, panggil perintah delete-deployment-group, dengan menentukan:

- Nama aplikasi yang terkait dengan grup penyebaran. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran yang terkait dengan aplikasi. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.

Bekerja dengan revisi aplikasi untuk CodeDeploy

Dalam CodeDeploy, revisi berisi versi file sumber yang CodeDeploy akan disebarkan ke instance atau skrip Anda CodeDeploy akan berjalan pada instance Anda.

Anda merencanakan revisi, menambahkan AppSpec file ke revisi, dan kemudian mendorong revisi ke Amazon S3 atau. GitHub Setelah Anda mendorong revisi, Anda dapat menerapkannya.

Topik

- Rencanakan revisi untuk CodeDeploy
- Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy
- Pilih jenis CodeDeploy repositori
- Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan)
- Lihat detail revisi aplikasi dengan CodeDeploy
- Daftarkan revisi aplikasi di Amazon S3 dengan CodeDeploy

Rencanakan revisi untuk CodeDeploy

Perencanaan yang baik membuat penerapan revisi menjadi lebih mudah.

Untuk penerapan ke AWS Lambda atau platform komputasi Amazon ECS, revisi sama dengan file. AppSpec Informasi berikut tidak berlaku. Untuk informasi selengkapnya, lihat Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy

Untuk penerapan ke platform EC2/On-Premises komputasi, mulailah dengan membuat direktori root kosong (folder) pada mesin pengembangan. Di sinilah Anda akan menyimpan file sumber (seperti file teks dan biner, executable, paket, dan sebagainya) untuk digunakan ke instance atau skrip yang akan dijalankan pada instance.

Misalnya, di folder `/tmp/` root di Linux, macOS, atau Unix atau folder `c:\temp` root di 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
```

Folder root juga harus menyertakan file spesifikasi aplikasi (AppSpec file), seperti yang ditunjukkan di sini. Lihat informasi yang lebih lengkap di Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.

Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy

Topik ini menunjukkan cara menambahkan AppSpec file ke penerapan Anda. Ini juga termasuk template untuk membuat AppSpec file untuk AWS Lambda dan EC2/On-Premises penyebaran.

Topik

- Menambahkan AppSpec file untuk penyebaran Amazon ECS
- Menambahkan AppSpec file untuk AWS Penyebaran Lambda
- Menambahkan AppSpec file untuk EC2/On-Premises penerapan

Menambahkan AppSpec file untuk penyebaran Amazon ECS

Untuk penerapan ke platform komputasi Amazon ECS:

- AppSpec File tersebut menentukan definisi tugas Amazon ECS yang digunakan untuk penerapan, nama kontainer, dan pemetaan port yang digunakan untuk merutekan lalu lintas, dan fungsi Lambda opsional yang dijalankan setelah peristiwa siklus hidup penerapan.
- Revisi sama dengan AppSpec file.
- AppSpec File dapat ditulis menggunakan JSON atau YAMAL.
- AppSpec File dapat disimpan sebagai file teks atau dimasukkan langsung ke konsol saat Anda membuat penerapan. Untuk informasi selengkapnya, lihat Membuat penerapan Platform Komputasi Amazon ECS (konsol).

Untuk membuat AppSpec file

1. Salin template JSON atau YAMAL ke editor teks atau ke AppSpec editor di konsol.

2. Ubah templat sesuai kebutuhan.
3. Gunakan validator JSON atau YAMAL untuk memvalidasi file Anda. AppSpec Jika Anda menggunakan AppSpec editor, file tersebut divalidasi saat Anda memilih Buat penerapan.
4. Jika Anda menggunakan editor teks, simpan file. Jika Anda menggunakan file AWS CLI untuk membuat penerapan, rujuk AppSpec file jika ada di hard drive Anda atau di bucket Amazon S3. Jika Anda menggunakan konsol, Anda harus mendorong AppSpec file Anda ke Amazon S3.

Template AppSpec file YAMAL untuk penerapan Amazon ECS dengan instruksi

Berikut ini adalah template YAMAL dari AppSpec file untuk penyebaran Amazon ECS dengan semua opsi yang tersedia. Untuk informasi tentang peristiwa siklus hidup yang akan digunakan di `hooks` bagian ini, lihat. AppSpec Bagian 'kait' untuk penerapan 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 File JSON untuk template penyebaran Amazon ECS

Berikut ini adalah template JSON untuk AppSpec file untuk penyebaran Amazon ECS dengan semua opsi yang tersedia. Untuk instruksi template, lihat komentar dalam versi YAMAL di bagian sebelumnya. Untuk informasi tentang peristiwa siklus hidup yang akan digunakan di `hooks` bagian ini, lihat. AppSpec Bagian 'kait' untuk penerapan 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": ""
  }
 ]
}
```

Menambahkan AppSpec file untuk AWS Penyebaran Lambda

Untuk penerapan ke platform komputasi AWS Lambda:

- AppSpec File berisi instruksi tentang fungsi Lambda yang akan digunakan dan digunakan untuk validasi penerapan.
- Revisi sama dengan AppSpec file.
- AppSpec File dapat ditulis menggunakan JSON atau YAMAL.
- AppSpec File dapat disimpan sebagai file teks atau dimasukkan langsung ke AppSpec editor konsol saat membuat penerapan. Untuk informasi selengkapnya, lihat Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol).

Untuk membuat AppSpec file:

1. Salin template JSON atau YAMAL ke editor teks atau ke AppSpec editor di konsol.
2. Ubah templat sesuai kebutuhan.
3. Gunakan validator JSON atau YAMAL untuk memvalidasi file Anda. AppSpec Jika Anda menggunakan AppSpec editor, file tersebut divalidasi saat Anda memilih Buat penerapan.
4. Jika Anda menggunakan editor teks, simpan file. Jika Anda menggunakan file AWS CLI untuk membuat penerapan, rujuk AppSpec file jika ada di hard drive Anda atau di bucket Amazon S3. Jika Anda menggunakan konsol, Anda harus mendorong AppSpec file Anda ke Amazon S3.

Template AppSpec file YAMAL untuk AWS Lambda penyebaran dengan instruksi

Untuk informasi tentang peristiwa siklus hidup yang akan digunakan di bagian kait, lihat. AppSpec Bagian 'kait' untuk sebuah AWS Penyebaran 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 File JSON untuk AWS Lambda template penerapan

Dalam template berikut, ganti MyFunction "" dengan nama AWS Lambda fungsi Anda. Di bagian Hooks opsional, ganti peristiwa siklus hidup dengan atau. BeforeAllowTraffic AfterAllowTraffic

Untuk informasi tentang peristiwa siklus hidup yang akan digunakan di bagian Hooks, lihat. AppSpec Bagian 'kait' untuk sebuah AWS Penyebaran Lambda

```
{
  "version": 0.0,
  "Resources": [{
   "MyFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "",
     "Alias": "",
     "CurrentVersion": "",
     "TargetVersion": ""
    }
   }
  }],
  "Hooks": [{
    "LifeCycleEvent": ""
   }
  ]
 }
```

Menambahkan AppSpec file untuk EC2/On-Premises penerapan

Tanpa AppSpec file, CodeDeploy tidak dapat memetakan file sumber dalam revisi aplikasi Anda ke tujuan mereka atau menjalankan skrip untuk penerapan Anda ke platform komputasi. EC2/On-Premises

Setiap revisi harus berisi hanya satu AppSpec file.

Untuk menambahkan AppSpec file ke revisi:

1. Salin template ke editor teks.
2. Ubah templat sesuai kebutuhan.
3. Gunakan validator YAMAL untuk memeriksa validitas file Anda. AppSpec
4. Simpan file seperti `appspec.yml` di direktori root revisi.

5. Jalankan salah satu perintah berikut untuk memverifikasi bahwa Anda telah menempatkan AppSpec file Anda di direktori root:

   - Untuk Linux, macOS, atau Unix:

     ```
     find /path/to/root/directory -name appspec.yml
     ```

     Tidak akan ada output jika AppSpec file tidak ditemukan di sana.

   - Untuk Windows:

     ```
     dir path\to\root\directory\appspec.yml
     ```

     Kesalahan File Not Found akan ditampilkan jika AppSpec file tidak disimpan di sana.

6. Dorong revisi ke Amazon GitHub S3 atau.

   Untuk petunjuk, lihat Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan).

AppSpec template file untuk EC2/On-Premises penyebaran dengan instruksi

> **Note**
>
> Penerapan ke instance Windows Server tidak mendukung elemen. `runas` Jika Anda menyebarkan ke instance Windows Server, jangan sertakan dalam file Anda AppSpec .

```
# 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:
```

Pilih jenis CodeDeploy repositori

Lokasi penyimpanan untuk file yang dibutuhkan oleh CodeDeploy disebut repositori. Penggunaan repositori tergantung pada platform komputasi mana yang digunakan penyebaran Anda.

- EC2/On-Premises: Untuk menyebarkan kode aplikasi Anda ke satu atau beberapa instance, kode Anda harus dibundel ke dalam file arsip dan ditempatkan di repositori tempat CodeDeploy dapat mengaksesnya selama proses penyebaran. Anda menggabungkan konten yang dapat di-deploy dan AppSpec file ke dalam file arsip, lalu mengunggahnya ke salah satu jenis repositori yang didukung oleh. CodeDeploy
- AWS Lambda dan Amazon ECS: Penerapan memerlukan AppSpec file, yang dapat diakses selama penerapan dengan salah satu cara berikut:
  - Dari ember Amazon S3.

- Dari teks yang diketik langsung ke AppSpec editor di konsol. Untuk informasi selengkapnya, lihat Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol) dan Membuat penerapan Platform Komputasi Amazon ECS (konsol).
- Jika Anda menggunakan AWS CLI, Anda dapat mereferensikan AppSpec file yang ada di hard drive Anda atau pada drive jaringan. Untuk informasi selengkapnya, lihat Buat sebuah AWS Penerapan Platform Komputasi Lambda (CLI) dan Membuat penerapan Platform Komputasi Amazon ECS (CLI).

CodeDeploy saat ini mendukung jenis repositori berikut:

Jenis Repositori	Rincian Repositori	Platform Komputasi yang Didukung
Amazon S3	Amazon Simple Storage Service (Amazon S3) adalah AWS solusi untuk penyimpanan objek yang aman dan dapat diskalakan. Amazon S3 menyimpan data sebagai objek dalam ember. Objek terdiri dari file dan, opsional, setiap metadata yang menggambarkan file itu. Untuk menyimpan objek di Amazon S3, Anda mengunggah file ke ember. Saat mengunggah file, Anda dapat mengatur izin dan metadata pada objek. Pelajari lebih lanjut: • Buat ember di Amazon S3 • Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan)	Penerapan yang menggunakan platform komputasi berikut dapat menyimpan revisi dalam bucket Amazon S3. • EC2/On-Premises • AWS Lambda • Amazon ECS

	• Secara otomatis menyebarkan dari Amazon S3 menggunakan CodeDeploy	
GitHub	Anda dapat menyimpan revisi aplikasi Anda di GitHubrepositori. Anda dapat memicu penerapan dari GitHub repositori setiap kali kode sumber di repositori itu diubah. Pelajari lebih lanjut: • Integrasi dengan CodeDeploy GitHub • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub	Hanya EC2/On-Premises penerapan yang dapat menyimpan revisi dalam repositori. GitHub

Bitbucket	Anda dapat menyebarkan kode ke grup penyebaran instans EC2 dengan menggunakan pipa di Bitbucket CodeDeploy Pipelines. Bitbucket Pipelines menawarkan integrasi berkelanjutan dan fitur penerapan berkelanjutan (CI/CD), termasuk Penerapan Bitbucket. CodeDeploy Pipa pertama-tama mendorong artefak ke bucket S3 yang telah Anda tentukan, dan kemudian menyebarkan artefak kode dari ember. Pelajari lebih lanjut: • Lihat CodeDeploy pipa untuk Bitbucket	Hanya EC2/On-Premises penerapan yang dapat menyimpan revisi dalam repositori. BitBucket

> **Note**
>
> AWS Lambda Penerapan hanya berfungsi dengan repositori Amazon S3.

Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan)

Setelah Anda merencanakan revisi seperti yang dijelaskan Rencanakan revisi untuk CodeDeploy dan menambahkan AppSpec file ke revisi seperti yang dijelaskan dalamTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy, Anda siap untuk menggabungkan file komponen dan mendorong revisi ke Amazon S3. Untuk penerapan ke instans Amazon EC2, setelah Anda mendorong revisi, Anda dapat CodeDeploy menggunakan untuk menerapkan revisi dari Amazon S3 ke instans.

> **Note**
>
> CodeDeploy juga dapat digunakan untuk menyebarkan revisi yang telah didorong ke. GitHub Untuk informasi selengkapnya, lihat GitHub dokumentasi Anda.

Kami berasumsi Anda telah mengikuti instruksi Memulai dengan CodeDeploy untuk mengatur AWS CLI. Ini sangat penting untuk memanggil push perintah yang dijelaskan nanti.

Pastikan Anda memiliki ember Amazon S3. Ikuti instruksi di Buat ember.

Jika penerapan Anda ke instans Amazon EC2, bucket Amazon S3 target harus dibuat atau ada di Wilayah yang sama dengan instans target. Misalnya, jika Anda ingin menyebarkan revisi ke beberapa contoh di Wilayah AS Timur (Virginia N.) dan contoh lainnya di Wilayah AS Barat (Oregon), maka Anda harus memiliki satu ember di Wilayah AS Timur (Virginia N.) dengan satu salinan revisi dan ember lain di Wilayah AS Barat (Oregon) dengan salinan lain dari revisi yang sama. Dalam skenario ini, Anda kemudian perlu membuat dua penyebaran terpisah, satu di Wilayah AS Timur (Virginia N.) dan satu lagi di Wilayah AS Barat (Oregon), meskipun revisinya sama di Wilayah dan ember.

Anda harus memiliki izin untuk mengunggah ke bucket Amazon S3. Anda dapat menentukan izin ini melalui kebijakan bucket Amazon S3. Misalnya, dalam kebijakan bucket Amazon S3 berikut, menggunakan karakter wildcard (*) memungkinkan AWS akun `111122223333` untuk mengunggah file ke direktori apa pun di bucket Amazon S3 bernama: `amzn-s3-demo-bucket`

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Untuk melihat ID AWS akun, lihat Menemukan ID AWS akun Anda.

Untuk mempelajari cara membuat dan melampirkan kebijakan bucket Amazon S3, lihat contoh kebijakan Bucket.

Pengguna yang memanggil push perintah harus memiliki, setidaknya, izin untuk mengunggah revisi ke setiap bucket Amazon S3 target. Misalnya, kebijakan berikut memungkinkan pengguna untuk mengunggah revisi di mana saja di bucket Amazon S3 yang diberi nama: `amzn-s3-demo-bucket`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Untuk mempelajari cara membuat dan melampirkan kebijakan IAM, lihat Bekerja dengan kebijakan.

Dorong revisi menggunakan AWS CLI

> **Note**
>
> `push`Perintah bundel artefak aplikasi dan AppSpec file ke dalam revisi. Format file revisi ini adalah file ZIP terkompresi. Perintah tidak dapat digunakan dengan AWS Lambda atau penyebaran Amazon ECS karena masing-masing mengharapkan revisi yang merupakan file atau. JSON-formatted YAML-formatted AppSpec

Panggil push perintah untuk bundel dan dorong revisi untuk penerapan. Parameternya adalah:

- --application-name: (string) Diperlukan. Nama CodeDeploy aplikasi yang akan dikaitkan dengan revisi aplikasi.
- --s3-location: (string) Diperlukan. Informasi tentang lokasi revisi aplikasi yang akan diunggah ke Amazon S3. Anda harus menentukan bucket Amazon S3 dan kunci. Kuncinya adalah nama revisi. CodeDeploy zip konten sebelum diunggah. Gunakan format `s3://`*`amzn-s3-demo-bucket`*`/`*`your-key.zip`*.
- --ignore-hidden-filesatau--no-ignore-hidden-files: (boolean) Opsional. Gunakan `--no-ignore-hidden-files` bendera (default) untuk menggabungkan dan mengunggah file tersembunyi ke Amazon S3. Gunakan `--ignore-hidden-files` bendera untuk tidak membundel dan mengunggah file tersembunyi ke Amazon S3.
- --source(string) Opsional. Lokasi konten yang akan digunakan dan AppSpec file pada mesin pengembangan yang akan di-zip dan diunggah ke Amazon S3. Lokasi ditentukan sebagai jalur relatif terhadap direktori saat ini. Jika jalur relatif tidak ditentukan atau jika satu periode digunakan untuk jalur (“.”), direktori saat ini digunakan.
- --description(string) Opsional. Komentar yang merangkum revisi aplikasi. Jika tidak ditentukan, string default “Diunggah oleh AWS CLI 'waktu' UTC” digunakan, di mana 'waktu' adalah waktu sistem saat ini di Coordinated Universal Time (UTC).

Anda dapat menggunakan AWS CLI untuk mendorong revisi untuk penerapan Amazon EC2. Contoh perintah push terlihat seperti ini:

Di Linux, macOS, atau 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 .
```

Di 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 .
```

Perintah ini melakukan hal berikut:

- Mengaitkan file yang dibundel dengan aplikasi bernama. `WordPress_App`
- Melampirkan deskripsi pada revisi.
- Mengabaikan file tersembunyi.
- Beri nama revisi `WordPressApp.zip` dan dorong ke ember bernama. `amzn-s3-demo-bucket`
- Bundel semua file di direktori root ke dalam revisi.

Setelah push berhasil, Anda dapat menggunakan AWS CLI atau CodeDeploy konsol untuk menyebarkan revisi dari Amazon S3. Untuk menerapkan revisi ini dengan: AWS CLI

Di Linux, macOS, atau 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
```

Di 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
```

Lihat informasi yang lebih lengkap di Buat penerapan dengan CodeDeploy.

Lihat detail revisi aplikasi dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang semua revisi aplikasi yang terdaftar ke AWS akun Anda untuk aplikasi tertentu.

Untuk informasi tentang mendaftarkan revisi, lihatDaftarkan revisi aplikasi di Amazon S3 dengan CodeDeploy.

Topik

- Lihat detail revisi aplikasi (konsol)
- Lihat detail revisi aplikasi (CLI)

Lihat detail revisi aplikasi (konsol)

Untuk melihat detail revisi aplikasi:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy dan pilih Applications.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

3. Pilih nama aplikasi yang memiliki revisi yang ingin Anda lihat.
4. Pada halaman Detail aplikasi, pilih tab Revisi, dan tinjau daftar revisi yang terdaftar untuk aplikasi. Pilih revisi, lalu pilih Lihat detail.

Lihat detail revisi aplikasi (CLI)

Untuk menggunakan AWS CLI untuk melihat revisi aplikasi, panggil get-application-revisionperintah atau list-application-revisionsperintah.

> **Note**
>
> Referensi untuk GitHub diterapkan pada penerapan hanya untuk EC2/On-Premises penerapan. Revisi untuk AWS Lambda penerapan tidak berfungsi dengan. GitHub

Untuk melihat detail tentang satu revisi aplikasi, panggil perintah get-application-revision, dengan menentukan:

- Nama aplikasi. Untuk mendapatkan nama aplikasi, panggil perintah daftar-aplikasi.
- Untuk revisi yang disimpan di GitHub, nama GitHub repositori dan ID komit yang mereferensikan revisi aplikasi yang didorong ke repositori.
- Untuk revisi yang disimpan di Amazon S3, nama bucket Amazon S3 yang berisi revisi; nama dan jenis file file arsip yang diunggah; dan, secara opsional, pengenal versi Amazon S3 file arsip dan ETag. Jika pengenal versi, ETag, atau keduanya ditentukan selama panggilan ke register-application-revision, mereka harus ditentukan di sini.

Untuk melihat detail tentang beberapa revisi aplikasi, panggil perintah list-application-revisions, dengan menentukan:

- Nama aplikasi. Untuk mendapatkan nama aplikasi, panggil perintah daftar-aplikasi.
- Secara opsional, untuk melihat detail hanya untuk revisi aplikasi Amazon S3, nama bucket Amazon S3 berisi revisi.
- Secara opsional, untuk melihat detail hanya untuk revisi aplikasi Amazon S3, string awalan untuk membatasi pencarian ke revisi aplikasi Amazon S3. (Jika tidak ditentukan, CodeDeploy akan mencantumkan semua revisi aplikasi Amazon S3 yang cocok.)
- Secara opsional, apakah akan mencantumkan detail revisi berdasarkan apakah setiap revisi adalah revisi target grup penyebaran. (Jika tidak ditentukan, CodeDeploy akan mencantumkan semua revisi yang cocok.)
- Secara opsional, nama kolom dan urutan yang digunakan untuk mengurutkan daftar detail revisi. (Jika tidak ditentukan, CodeDeploy akan mencantumkan hasil dalam urutan sewenang-wenang.)

Anda dapat membuat daftar semua revisi atau hanya revisi yang disimpan di Amazon S3. Anda tidak dapat mencantumkan hanya revisi yang disimpan di dalamnya GitHub.

Daftarkan revisi aplikasi di Amazon S3 dengan CodeDeploy

Jika Anda sudah memanggil perintah push untuk mendorong revisi aplikasi ke Amazon S3, Anda tidak perlu mendaftarkan revisi. Namun, jika Anda mengunggah revisi ke Amazon S3 melalui cara lain dan ingin revisi muncul di CodeDeploy konsol atau melalui AWS CLI, ikuti langkah-langkah ini untuk mendaftarkan revisi terlebih dahulu.

Jika Anda telah mendorong revisi aplikasi ke GitHub repositori dan ingin revisi muncul di CodeDeploy konsol atau melalui AWS CLI, Anda juga harus mengikuti langkah-langkah ini.

Anda hanya dapat menggunakan AWS CLI atau CodeDeploy API untuk mendaftarkan revisi aplikasi di Amazon GitHub S3 atau.

Topik

- Daftarkan revisi di Amazon S3 dengan CodeDeploy (CLI)
- Daftarkan revisi GitHub dengan CodeDeploy (CLI)

Daftarkan revisi di Amazon S3 dengan CodeDeploy (CLI)

1. Unggah revisi ke Amazon S3.
2. Panggil perintah register-application-revision, dengan menentukan:
   - Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
   - Informasi tentang revisi yang akan didaftarkan:
     - Nama bucket Amazon S3 yang berisi revisi.
     - Nama dan jenis file dari revisi yang diunggah. Untuk AWS Lambda penerapan, revisi adalah AppSpec file yang ditulis dalam JSON atau YAMAL. Untuk EC2/On-Premises penerapan, revisi berisi versi file sumber yang CodeDeploy akan diterapkan ke instance atau skrip Anda yang akan berjalan pada instance Anda. CodeDeploy

       > **Note**
       >
       > Format file arsip tar dan tar terkompresi (.tar dan.tar.gz) tidak didukung untuk instance Windows Server.

     - (Opsional) Pengidentifikasi versi Amazon S3 revisi. (Jika pengenal versi tidak ditentukan, CodeDeploy akan menggunakan versi terbaru.)
     - (Opsional) ETag revisi. (Jika ETag tidak ditentukan, CodeDeploy akan melewati validasi objek.)
   - (Opsional) Deskripsi apa pun yang ingin Anda kaitkan dengan revisi.

Informasi tentang revisi di Amazon S3 dapat ditentukan pada baris perintah, menggunakan sintaks ini sebagai bagian dari panggilan. register-application-revision (`version`dan `eTag` bersifat opsional.)

Untuk file revisi untuk EC2/On-Premises penerapan:

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|
zip,version=string,eTag=string
```

Untuk file revisi untuk penyebaran AWS Lambda:

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

Daftarkan revisi GitHub dengan CodeDeploy (CLI)

> **Note**
>
> AWS Lambda penerapan tidak berfungsi dengan. GitHub

1. Unggah revisi ke GitHub repositori Anda.
2. Panggil perintah register-application-revision, dengan menentukan:
   - Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
   - Informasi tentang revisi yang akan didaftarkan:
     - Nama GitHub pengguna atau grup yang ditetapkan ke repositori yang berisi revisi, diikuti dengan garis miring (/), diikuti dengan nama repositori.
     - ID komit yang mereferensikan revisi di repositori.
   - (Opsional) Deskripsi apa pun yang ingin Anda kaitkan dengan revisi.

Informasi tentang revisi di GitHub dapat ditentukan pada baris perintah, menggunakan sintaks ini sebagai bagian dari panggilan: register-application-revision

```
--github-location repository=string,commitId=string
```

Bekerja dengan penerapan di CodeDeploy

Dalam CodeDeploy, penyebaran adalah proses, dan komponen yang terlibat dalam proses, menginstal konten pada satu atau lebih contoh. Konten ini dapat terdiri dari kode, web dan file konfigurasi, executable, paket, skrip, dan sebagainya. CodeDeploy menyebarkan konten yang disimpan dalam repositori sumber, sesuai dengan aturan konfigurasi yang Anda tentukan.

Jika Anda menggunakan platform EC2/On-Premises komputasi, maka dua penerapan ke kumpulan instance yang sama dapat berjalan secara bersamaan.

CodeDeploy menyediakan dua opsi jenis penerapan, penerapan dan penerapan di tempat. blue/green

- In-place deployment: Aplikasi pada setiap instance dalam grup penyebaran dihentikan, revisi aplikasi terbaru diinstal, dan versi baru aplikasi dimulai dan divalidasi. Anda dapat menggunakan penyeimbang beban sehingga setiap instance dideregistrasi selama penerapannya dan kemudian dikembalikan ke layanan setelah penerapan selesai. Hanya penerapan yang menggunakan platform EC2/On-Premises komputasi yang dapat menggunakan penerapan di tempat. Untuk informasi selengkapnya tentang penerapan di tempat, lihat. Ikhtisar penerapan di tempat
- Blue/green deployment: Perilaku penerapan Anda bergantung pada platform komputasi yang Anda gunakan:
  - Blue/green pada platform EC2/On-Premises komputasi: Instance dalam grup penerapan (lingkungan asli) digantikan oleh serangkaian instance yang berbeda (lingkungan pengganti) menggunakan langkah-langkah ini:
    - Instans disediakan untuk lingkungan pengganti.
    - Revisi aplikasi terbaru diinstal pada instance pengganti.
    - Waktu tunggu opsional terjadi untuk kegiatan seperti pengujian aplikasi dan verifikasi sistem.
    - Instans di lingkungan penggantian terdaftar dengan satu atau lebih penyeimbang beban Elastic Load Balancing, menyebabkan lalu lintas dialihkan ke sana. Contoh di lingkungan asli dideregistrasi dan dapat dihentikan atau terus berjalan untuk penggunaan lain.

> **Note**
>
> Jika Anda menggunakan platform EC2/On-Premises komputasi, ketahuilah bahwa blue/green penerapan hanya berfungsi dengan instans Amazon EC2.

- Blue/green pada platform komputasi Amazon ECS AWS Lambda atau Amazon: Lalu lintas digeser secara bertahap sesuai dengan konfigurasi penerapan kenari, linier, atau all-at-once.
- Blue/green penerapan melalui CloudFormation: Lalu lintas dialihkan dari sumber daya Anda saat ini ke sumber daya yang diperbarui sebagai bagian dari pembaruan CloudFormation tumpukan. Saat ini, hanya blue/green penerapan ECS yang didukung.

Untuk informasi selengkapnya tentang blue/green penerapan, lihat. Ikhtisar blue/green penerapan

Untuk informasi tentang penerapan otomatis dari Amazon S3, lihat Menerapkan secara otomatis dari Amazon S3 menggunakan. CodeDeploy

Topik

- Buat penerapan dengan CodeDeploy
- Lihat CodeDeploy detail penerapan
- Lihat data log untuk CodeDeploy EC2/On-Premises penerapan
- Hentikan penerapan dengan CodeDeploy
- Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy
- Menyebarkan aplikasi di tempat yang berbeda AWS akun
- Gunakan CodeDeploy agen untuk memvalidasi paket penerapan pada mesin lokal

Buat penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk membuat penerapan yang menginstal revisi aplikasi yang telah Anda dorong ke Amazon S3 atau, jika penerapan Anda ke platform EC2/On-Premises komputasi GitHub, pada instans dalam grup penerapan.

Proses untuk membuat penerapan tergantung pada platform komputasi yang digunakan oleh penerapan Anda.

Topik

- Prasyarat penyebaran
- Membuat penerapan Platform Komputasi Amazon ECS (konsol)
- Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol)
- Membuat penerapan Platform EC2/On-Premises Komputasi (konsol)

- Membuat penerapan Platform Komputasi Amazon ECS (CLI)
- Buat sebuah AWS Penerapan Platform Komputasi Lambda (CLI)
- Membuat penerapan Platform EC2/On-Premises Komputasi (CLI)
- Buat blue/green penyebaran Amazon ECS melalui CloudFormation

Prasyarat penyebaran

Pastikan langkah-langkah berikut selesai sebelum Anda memulai penerapan.

Prasyarat penerapan pada AWS Lambda platform komputasi

- Buat aplikasi yang mencakup setidaknya satu grup penyebaran. Untuk informasi selengkapnya, lihat Buat aplikasi dengan CodeDeploy dan Buat grup penerapan dengan CodeDeploy.
- Siapkan revisi aplikasi, juga dikenal sebagai AppSpec file, yang menentukan versi fungsi Lambda yang ingin Anda terapkan. AppSpec File ini juga dapat menentukan fungsi Lambda untuk memvalidasi penerapan Anda. Untuk mengetahui informasi selengkapnya, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
- Jika Anda ingin menggunakan konfigurasi penerapan khusus untuk penerapan Anda, buat sebelum memulai proses penerapan. Untuk informasi, lihat Create a Deployment Configuration.

Prasyarat penerapan pada platform komputasi EC2/on-premises

- Untuk penerapan di tempat, buat atau konfigurasikan instance yang ingin Anda gunakan. Untuk informasi, lihat Bekerja dengan instance untuk CodeDeploy. Untuk blue/green penerapan, Anda memiliki grup Penskalaan Otomatis Amazon EC2 yang sudah ada untuk digunakan sebagai templat untuk lingkungan pengganti, atau Anda memiliki satu atau beberapa instans atau grup Penskalaan Otomatis Amazon EC2 yang Anda tentukan sebagai lingkungan asli. Untuk informasi selengkapnya, lihat Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling dan Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling.
- Buat aplikasi yang mencakup setidaknya satu grup penyebaran. Untuk informasi selengkapnya, lihat Buat aplikasi dengan CodeDeploy dan Buat grup penerapan dengan CodeDeploy.
- Siapkan revisi aplikasi yang ingin Anda terapkan ke instance di grup penyebaran Anda. Untuk informasi, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
- Jika Anda ingin menggunakan konfigurasi penerapan khusus untuk penerapan Anda, buat sebelum memulai proses penerapan. Untuk informasi, lihat Create a Deployment Configuration.

- Jika Anda menerapkan revisi aplikasi dari bucket Amazon S3, bucket berada di Wilayah yang AWS sama dengan instance di grup penerapan Anda.
- Jika Anda menerapkan revisi aplikasi dari bucket Amazon S3, kebijakan bucket Amazon S3 telah diterapkan ke bucket. Kebijakan ini memberi instans Anda izin yang diperlukan untuk mengunduh revisi aplikasi.

  Misalnya, kebijakan bucket Amazon S3 berikut ini memungkinkan instans Amazon EC2 dengan profil instans IAM terlampir yang berisi `arn:aws:iam::444455556666:role/CodeDeployDemo` ARN untuk diunduh dari mana saja di bucket Amazon S3 bernama: `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"
                ]
            }
        }
    ]
}
```

  Kebijakan bucket Amazon S3 berikut memungkinkan instans lokal dengan pengguna IAM terkait yang berisi ARN `arn:aws:iam::444455556666:user/CodeDeployUser` dapat diunduh dari mana saja di bucket Amazon S3 yang diberi nama: `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"
                ]
            }
        }
    ]
}
```

Untuk informasi tentang cara membuat dan melampirkan kebijakan bucket Amazon S3, lihat contoh kebijakan Bucket.

- Jika Anda membuat blue/green penerapan, atau Anda telah menetapkan Classic Load Balancer opsional, Application Load Balancer, atau Network Load Balancer dalam grup penyebaran untuk penerapan di tempat, Anda telah membuat VPC menggunakan Amazon VPC yang berisi setidaknya dua subnet. (CodeDeploy menggunakan Elastic Load Balancing, yang mengharuskan semua instance dalam grup penyeimbang beban berada dalam satu VPC.)

  Jika Anda belum membuat VPC, lihat Panduan Memulai VPC Amazon.

- Jika Anda membuat blue/green penerapan, Anda telah mengonfigurasi setidaknya satu Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing dan menggunakannya untuk mendaftarkan instance yang membentuk lingkungan asli Anda.

  > **Note**
  >
  > Instans di lingkungan pengganti Anda akan didaftarkan ke penyeimbang beban nanti.

  Untuk informasi selengkapnya tentang mengonfigurasi penyeimbang beban, lihatSiapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2, dan. Siapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS

Prasyarat penerapan untuk penerapan melalui blue/green CloudFormation

- Template Anda tidak perlu memodelkan sumber daya untuk CodeDeploy aplikasi atau grup penyebaran.

- Template Anda harus menyertakan sumber daya untuk VPC menggunakan Amazon VPC yang berisi setidaknya dua subnet.
- Template Anda harus menyertakan sumber daya untuk satu atau lebih Classic Load Balancer, Application Load Balancer, atau Network Load Balancer di Elastic Load Balancing yang digunakan untuk mengarahkan lalu lintas ke grup target Anda.

Membuat penerapan Platform Komputasi Amazon ECS (konsol)

Topik ini menunjukkan cara menerapkan layanan Amazon ECS menggunakan konsol. Untuk informasi selengkapnya, lihat Tutorial: Menyebarkan aplikasi ke Amazon ECS dan Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Lakukan salah satu tindakan berikut:

   - Jika Anda ingin menyebarkan aplikasi, di panel navigasi, perluas Deploy, lalu pilih Aplikasi. Pilih nama aplikasi yang ingin Anda terapkan. Pastikan kolom Compute platform untuk aplikasi Anda adalah Amazon ECS.
   - Jika Anda ingin menerapkan ulang penerapan, di panel navigasi, perluas Deploy, lalu pilih Deployment. Pilih penyebaran yang ingin Anda gunakan kembali, dan di kolom Aplikasi, pilih nama aplikasinya. Pastikan kolom Compute platform untuk penerapan Anda adalah Amazon ECS.

3. Pada tab Deployment, pilih Create deployment.

   > **Note**
   >
   > Aplikasi Anda harus memiliki grup penyebaran sebelum dapat digunakan. Jika aplikasi Anda tidak memiliki grup penyebaran, pada tab Grup Deployment, pilih Buat grup

penyebaran. Untuk informasi selengkapnya, lihat Buat grup penerapan dengan CodeDeploy.

4. Di grup Deployment, pilih grup penyebaran yang akan digunakan untuk penerapan ini.
5. Di samping lokasi revisi, pilih lokasi revisi Anda:

   - Aplikasi saya disimpan di Amazon S3 — Untuk informasi, lihatMenentukan informasi tentang revisi yang disimpan dalam bucket Amazon S3, dan kemudian kembali ke langkah 6.
   - Gunakan AppSpec editor - Pilih JSON atau YAMAL, lalu masukkan AppSpec file Anda ke editor. Anda dapat menyimpan AppSpec file dengan memilih Simpan sebagai file teks. Ketika Anda memilih Deploy di akhir langkah-langkah ini, Anda menerima kesalahan jika JSON atau YAMAL Anda tidak valid. Untuk informasi selengkapnya tentang membuat AppSpec file, lihatTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.
6. (Opsional) Dalam deskripsi Deployment, masukkan deskripsi untuk penerapan ini.
7. (Opsional) Dalam penggantian konfigurasi Rollback, Anda dapat menentukan opsi rollback otomatis yang berbeda untuk penerapan ini daripada yang ditentukan untuk grup penerapan, jika ada.

   Untuk informasi tentang rollback di CodeDeploy, lihat Penerapan ulang dan rollback penerapan dan. Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy

   Pilih dari yang berikut ini:

   - Kembalikan saat penerapan gagal — CodeDeploy menyebarkan kembali revisi bagus terakhir yang diketahui sebagai penerapan baru.
   - Putar kembali saat ambang alarm terpenuhi - Jika alarm ditambahkan ke grup penerapan, pindahkan CodeDeploy revisi terakhir yang diketahui saat satu atau lebih alarm yang ditentukan diaktifkan.
   - Nonaktifkan rollback — Jangan lakukan rollback untuk penerapan ini.
8. Pilih Buat penerapan.

   Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol)

Topik ini menunjukkan cara menerapkan fungsi Lambda menggunakan konsol.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Lakukan salah satu tindakan berikut:

   - Jika Anda ingin menyebarkan aplikasi, di panel navigasi, perluas Deploy, lalu pilih Aplikasi. Pilih nama aplikasi yang ingin Anda terapkan. Pastikan kolom Compute platform untuk aplikasi Anda adalah AWS Lambda.
   - Jika Anda ingin menerapkan ulang penerapan, di panel navigasi, perluas Deploy, lalu pilih Deployment. Pilih penyebaran yang ingin Anda gunakan kembali, dan di kolom Aplikasi, pilih nama aplikasinya. Pastikan kolom Compute platform untuk penerapan Anda adalah Lambda AWS .

3. Pada tab Deployment, pilih Create deployment.

   > **Note**
   >
   > Aplikasi Anda harus memiliki grup penyebaran sebelum dapat digunakan. Jika aplikasi Anda tidak memiliki grup penyebaran, pada tab Grup Deployment, pilih Buat grup penyebaran. Untuk informasi selengkapnya, lihat Buat grup penerapan dengan CodeDeploy.

4. Di grup Deployment, pilih grup penyebaran yang akan digunakan untuk penerapan ini.
5. Di samping lokasi revisi, pilih lokasi revisi Anda:

   - Aplikasi saya disimpan di Amazon S3 — Untuk informasi, lihatMenentukan informasi tentang revisi yang disimpan dalam bucket Amazon S3, dan kemudian kembali ke langkah 6.
   - Gunakan AppSpec editor - Pilih JSON atau YAMAL, lalu masukkan AppSpec file Anda ke editor. Anda dapat menyimpan AppSpec file dengan memilih Simpan sebagai file teks. Ketika Anda memilih Deploy di akhir langkah-langkah ini, Anda menerima kesalahan jika JSON atau YAMAL Anda tidak valid. Untuk informasi selengkapnya tentang membuat AppSpec file, lihatTambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy.

6. (Opsional) Dalam deskripsi Deployment, masukkan deskripsi untuk penerapan ini.

7. (Opsional) Perluas penggantian grup Deployment untuk memilih konfigurasi penerapan untuk mengontrol bagaimana lalu lintas dialihkan ke versi fungsi Lambda yang berbeda dari yang ditentukan dalam grup penerapan.

   Untuk informasi selengkapnya, lihat Konfigurasi penerapan pada sebuah AWS Lambda platform komputasi.

8. (Opsional) Dalam penggantian konfigurasi Rollback, Anda dapat menentukan opsi rollback otomatis yang berbeda untuk penerapan ini daripada yang ditentukan untuk grup penerapan, jika ada.

   Untuk informasi tentang rollback di CodeDeploy, lihat Penerapan ulang dan rollback penerapan dan. Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy

   Pilih dari yang berikut ini:

   - Kembalikan saat penerapan gagal — CodeDeploy menyebarkan kembali revisi bagus terakhir yang diketahui sebagai penerapan baru.
   - Putar kembali saat ambang alarm terpenuhi - Jika alarm ditambahkan ke grup penerapan, pindahkan CodeDeploy revisi terakhir yang diketahui saat satu atau lebih alarm yang ditentukan diaktifkan.
   - Nonaktifkan rollback — Jangan lakukan rollback untuk penerapan ini.

9. Pilih Buat penerapan.

   Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

Membuat penerapan Platform EC2/On-Premises Komputasi (konsol)

Topik ini menunjukkan cara menerapkan aplikasi ke Amazon EC2 atau server lokal menggunakan konsol.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Lakukan salah satu tindakan berikut:

   - Jika Anda ingin menyebarkan aplikasi, di panel navigasi, perluas Deploy, lalu pilih Aplikasi. Pilih nama aplikasi yang ingin Anda terapkan. Pastikan kolom Compute platform untuk aplikasi Anda. EC2/On-Premises
   - Jika Anda ingin menerapkan ulang penerapan, di panel navigasi, perluas Deploy, lalu pilih Deployment. Temukan penyebaran yang ingin Anda gunakan kembali, lalu pilih nama aplikasinya di kolom Aplikasi. Pastikan kolom Compute platform untuk penerapan Anda. EC2/On-Premises

3. Pada tab Deployment, pilih Create deployment.

   > **Note**
   >
   > Aplikasi Anda harus memiliki grup penyebaran sebelum dapat digunakan. Jika aplikasi Anda tidak memiliki grup penyebaran, pada tab Grup Deployment, pilih Buat grup penyebaran. Untuk informasi selengkapnya, lihat Buat grup penerapan dengan CodeDeploy.

4. Di grup Deployment, pilih grup penyebaran yang akan digunakan untuk penerapan ini.
5. Di samping jenis Repositori, pilih jenis repositori yang disimpan dalam revisi Anda:

   - Aplikasi saya disimpan di Amazon S3 — Untuk informasi, lihatMenentukan informasi tentang revisi yang disimpan dalam bucket Amazon S3, dan kemudian kembali ke langkah 6.
   - Aplikasi saya disimpan di GitHub — Untuk informasi, lihatTentukan informasi tentang revisi yang disimpan dalam repositori GitHub, dan kemudian kembali ke langkah 6.

6. (Opsional) Dalam deskripsi Deployment, masukkan deskripsi untuk penerapan ini.
7. (Opsional) Perluas konfigurasi penerapan Override untuk memilih konfigurasi penerapan guna mengontrol cara lalu lintas dialihkan ke Amazon EC2 atau server lokal yang berbeda dari yang ditentukan dalam grup penerapan.

   Untuk informasi selengkapnya, lihat Bekerja dengan konfigurasi penerapan di CodeDeploy.

8. a. Pilih Jangan gagal penerapan jika peristiwa ApplicationStop siklus hidup gagal jika Anda ingin penerapan ke instance berhasil jika peristiwa siklus hidup gagal. `ApplicationStop`

   b. Perluas Setelan perilaku penerapan tambahan untuk menentukan cara CodeDeploy menangani file di lokasi target penerapan yang bukan bagian dari penerapan yang berhasil sebelumnya.

Pilih dari yang berikut ini:

- Gagal penerapan - Kesalahan dilaporkan dan status penerapan diubah menjadi. `Failed`
- Timpa konten — Jika file dengan nama yang sama ada di lokasi target, versi dari revisi aplikasi menggantikannya.
- Pertahankan konten — Jika file dengan nama yang sama ada di lokasi target, file disimpan dan versi dalam revisi aplikasi tidak disalin ke instance.

Untuk informasi selengkapnya, lihat Perilaku rollback dengan konten yang ada.

9. (Opsional) Dalam penggantian konfigurasi Rollback, Anda dapat menentukan opsi rollback otomatis yang berbeda untuk penerapan ini daripada yang ditentukan untuk grup penerapan, jika ada.

   Untuk informasi tentang rollback di CodeDeploy, lihat Penerapan ulang dan rollback penerapan dan. Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy

   Pilih dari yang berikut ini:

   - Kembalikan saat penerapan gagal — CodeDeploy menyebarkan kembali revisi bagus terakhir yang diketahui sebagai penerapan baru.
   - Putar kembali saat ambang alarm terpenuhi - Jika alarm ditambahkan ke grup CodeDeploy penyebaran, gunakan revisi terakhir yang diketahui saat satu atau lebih alarm yang ditentukan diaktifkan.
   - Nonaktifkan rollback — Jangan lakukan rollback untuk penerapan ini.

10. Pilih Mulai penerapan.

    Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

Topik

- Menentukan informasi tentang revisi yang disimpan dalam bucket Amazon S3
- Tentukan informasi tentang revisi yang disimpan dalam repositori GitHub

Menentukan informasi tentang revisi yang disimpan dalam bucket Amazon S3

Jika Anda mengikuti langkah-langkahnyaMembuat penerapan Platform EC2/On-Premises Komputasi (konsol), ikuti langkah-langkah berikut untuk menambahkan detail tentang revisi aplikasi yang disimpan dalam bucket Amazon S3.

1. Salin tautan Amazon S3 revisi Anda ke lokasi Revisi. Untuk menemukan nilai tautan:

   a. Di tab browser terpisah:

      Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

      Jelajahi dan pilih revisi Anda.

   b. Jika panel Properties tidak terlihat, pilih tombol Properties.

   c. Di panel Properti, salin nilai bidang Tautan ke kotak Lokasi revisi di CodeDeploy konsol.

   Untuk menentukan ETag (checksum file) sebagai bagian dari lokasi revisi:

   - Jika nilai bidang Tautan berakhir**?versionId=*versionId***, tambahkan **&etag=** dan ETag ke akhir nilai bidang Tautan.
   - Jika nilai bidang Tautan tidak menentukan ID versi, tambahkan **?etag=** dan ETag ke akhir nilai bidang Tautan.

   > **Note**
   >
   > Meskipun tidak semudah menyalin nilai bidang Tautan, Anda juga dapat mengetikkan lokasi revisi dalam salah satu format berikut:
   > **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. Jika pesan muncul di daftar Jenis file yang mengatakan jenis file tidak dapat dideteksi, pilih jenis file revisi. Jika tidak, terima jenis file yang terdeteksi.

Tentukan informasi tentang revisi yang disimpan dalam repositori GitHub

Jika Anda mengikuti langkah-langkahMembuat penerapan Platform EC2/On-Premises Komputasi (konsol), ikuti langkah-langkah ini untuk menambahkan detail tentang revisi aplikasi yang disimpan dalam repositori GitHub .

1. Di Connect to GitHub, lakukan salah satu hal berikut:

   - Untuk membuat koneksi untuk CodeDeploy aplikasi ke GitHub akun, di tab browser web yang berbeda, keluar dari GitHub. Di GitHub akun, masukkan nama untuk mengidentifikasi koneksi ini, lalu pilih Connect to GitHub. Halaman web meminta Anda untuk mengizinkan CodeDeploy untuk berinteraksi dengan GitHub aplikasi Anda. Lanjutkan ke langkah 2.
   - Untuk menggunakan koneksi yang telah Anda buat, di GitHubakun, pilih namanya, lalu pilih Connect to GitHub. Lanjutkan ke langkah 4.
   - Untuk membuat koneksi ke GitHub akun yang berbeda, di tab browser web yang berbeda, keluar dari GitHub. Pilih Connect to a different GitHub account, lalu pilih Connect to GitHub. Lanjutkan ke langkah 2.
2. Jika Anda diminta untuk masuk GitHub, ikuti petunjuk di halaman Masuk. Masuk dengan nama GitHub pengguna atau email dan kata sandi Anda.
3. Jika halaman aplikasi Otorisasi muncul, pilih Otorisasi aplikasi.
4. Pada halaman Create deployment, di kotak Nama repositori, masukkan nama GitHub pengguna atau organisasi yang berisi revisi, diikuti dengan garis miring (/), diikuti dengan nama repositori yang berisi revisi. Jika Anda tidak yakin dengan nilai yang akan diketik:

   a. Di tab browser web yang berbeda, buka GitHubdasbor Anda.

   b. Di repositori Anda, arahkan kursor mouse Anda ke nama repositori target. Tooltip muncul, menampilkan nama GitHub pengguna atau organisasi, diikuti dengan garis miring (/), diikuti dengan nama repositori. Masukkan nilai yang ditampilkan ini ke dalam kotak nama Repositori.

   > **Note**
   >
   > Jika nama repositori target tidak terlihat di repositori Anda, gunakan GitHub kotak Pencarian untuk menemukan nama repositori target dan nama pengguna atau organisasi. GitHub

5. Di Commit ID, masukkan ID komit yang mengacu pada revisi di repositori. Jika Anda tidak yakin dengan nilai yang akan diketik:

   a. Di tab browser web yang berbeda, buka GitHubdasbor Anda.

   b. Di repositori Anda, pilih nama repositori yang berisi komit target.

   c. Dalam daftar komit, temukan dan salin ID komit yang mengacu pada revisi di repositori. ID ini biasanya panjangnya 40 karakter dan terdiri dari huruf dan angka. (Jangan gunakan versi yang lebih pendek dari ID komit, yang biasanya merupakan 10 karakter pertama dari versi ID komit yang lebih panjang.)

   d. Rekatkan ID komit ke dalam kotak ID Komit.

Membuat penerapan Platform Komputasi Amazon ECS (CLI)

Setelah Anda membuat aplikasi dan revisi (dalam penyebaran Amazon ECS, ini adalah filenya): AppSpec

Panggil perintah create-deployment, dengan menentukan:

- Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.
- Informasi tentang revisi yang akan digunakan:

  Untuk revisi yang disimpan di Amazon S3:

  - Nama bucket Amazon S3 yang berisi revisi.
  - Nama revisi yang diunggah.
  - (Opsional) Pengidentifikasi versi Amazon S3 untuk revisi. (Jika pengenal versi tidak ditentukan, CodeDeploy gunakan versi terbaru.)
  - (Opsional) ETag untuk revisi. (Jika ETag tidak ditentukan, CodeDeploy lewati validasi objek.)

  Untuk revisi yang disimpan dalam file yang tidak ada di Amazon S3, Anda memerlukan nama file dan jalurnya. File revisi Anda ditulis menggunakan JSON atau YAMG, jadi kemungkinan besar memiliki ekstensi.json atau .yaml.
- (Opsional) Deskripsi untuk penerapan.

File revisi dapat ditentukan sebagai file yang diunggah ke bucket Amazon S3 atau sebagai string. Sintaks untuk masing-masing ketika digunakan sebagai bagian dari create-deployment perintah adalah:

- Ember Amazon S3:

  `versionDan eTag` bersifat opsional.

  ```
  --s3-location bucket=string,key=string,bundleType=JSON|
  YAML,version=string,eTag=string
  ```

- Tali:

  ```
  --revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
  ```

> **Note**
>
> create-deploymentPerintah dapat memuat revisi dari file. Untuk informasi selengkapnya, lihat Memuat parameter dari file.

Untuk templat revisi AWS Lambda penerapan, lihat. Menambahkan AppSpec file untuk AWS Penyebaran Lambda Untuk contoh revisi, lihatAppSpec Contoh file untuk AWS Penyebaran Lambda.

Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

Buat sebuah AWS Penerapan Platform Komputasi Lambda (CLI)

Setelah Anda membuat aplikasi dan revisi (dalam penerapan AWS Lambda, ini adalah file): AppSpec

Panggil perintah create-deployment, dengan menentukan:

- Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.
- Informasi tentang revisi yang akan digunakan:

  Untuk revisi yang disimpan di Amazon S3:

- Nama bucket Amazon S3 yang berisi revisi.
- Nama revisi yang diunggah.
- (Opsional) Pengidentifikasi versi Amazon S3 untuk revisi. (Jika pengenal versi tidak ditentukan, CodeDeploy gunakan versi terbaru.)
- (Opsional) ETag untuk revisi. (Jika ETag tidak ditentukan, CodeDeploy lewati validasi objek.)

  Untuk revisi yang disimpan dalam file yang tidak ada di Amazon S3, Anda memerlukan nama file dan jalurnya. File revisi Anda ditulis menggunakan JSON atau YAMG, jadi kemungkinan besar memiliki ekstensi.json atau .yaml.

- (Opsional) Nama konfigurasi penerapan yang akan digunakan. Untuk melihat daftar konfigurasi penerapan, panggil perintah list-deployment-configs. (Jika tidak ditentukan, CodeDeploy gunakan konfigurasi penerapan default tertentu.)
- (Opsional) Deskripsi untuk penerapan.

File revisi dapat ditentukan sebagai file yang diunggah ke bucket Amazon S3 atau sebagai string. Sintaks untuk masing-masing ketika digunakan sebagai bagian dari create-deployment perintah adalah:

- Ember Amazon S3:

  `version`Dan `eTag` bersifat opsional.

  ```
  --s3-location bucket=string,key=string,bundleType=JSON|
  YAML,version=string,eTag=string
  ```

- Tali:

  ```
  --revision '{"revisionType": "String", "string": {"content":"revision-as-string"}}'
  ```

> **Note**
>
> create-deploymentPerintah dapat memuat revisi dari file. Untuk informasi selengkapnya, lihat Memuat parameter dari file.

Untuk templat revisi AWS Lambda penerapan, lihat. Menambahkan AppSpec file untuk AWS Penyebaran Lambda Untuk contoh revisi, lihatAppSpec Contoh file untuk AWS Penyebaran Lambda.

Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

Membuat penerapan Platform EC2/On-Premises Komputasi (CLI)

Untuk menggunakan AWS CLI untuk menerapkan revisi ke platform EC2/On-Premises komputasi:

1. Setelah Anda menyiapkan instance, membuat aplikasi, dan mendorong revisi, lakukan salah satu hal berikut:
   - Jika Anda ingin menerapkan revisi dari bucket Amazon S3, lanjutkan ke langkah 2 sekarang.
   - Jika Anda ingin menerapkan revisi dari GitHub repositori, pertama-tama selesaikan langkah-langkahnya, lalu lanjutkan ke langkah 2. Connect CodeDeploy aplikasi ke GitHub repositori
2. Panggil perintah create-deployment, dengan menentukan:
   - `--application-name`: Nama aplikasi. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
   - `--deployment-group-name`: Nama grup penyebaran Amazon EC2. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.
   - `--revision`: Informasi tentang revisi yang akan digunakan:

     Untuk revisi yang disimpan di Amazon S3:
     - `s3Location`: Nama bucket Amazon S3 yang berisi revisi.
     - `s3Location --> key`: Nama revisi yang diunggah.
     - `s3Location --> bundleType`: Jenis file revisi yang diunggah.

       > **Note**
       >
       > Format file arsip tar dan tar terkompresi (.tar dan.tar.gz) tidak didukung untuk instance Windows Server.

     - `s3Location --> version`: (Opsional) Pengidentifikasi versi Amazon S3 untuk revisi. (Jika pengenal versi tidak ditentukan, CodeDeploy gunakan versi terbaru.)
     - `s3Location --> eTag`: (Opsional) ETag untuk revisi. (Jika ETag tidak ditentukan, CodeDeploy lewati validasi objek.)

     Untuk revisi yang disimpan di GitHub:

  - `gitHubLocation --> repository`: Nama GitHub pengguna atau grup yang ditetapkan ke repositori yang berisi revisi, diikuti dengan garis miring (`/`), diikuti dengan nama repositori.
  - `gitHubLocation --> commitId`: ID komit untuk revisi.
- `--deployment-config-name`: (Opsional) Nama konfigurasi penerapan yang akan digunakan. Untuk melihat daftar konfigurasi penerapan, panggil perintah list-deployment-configs. (Jika tidak ditentukan, CodeDeploy gunakan konfigurasi penerapan default tertentu.)
- `--ignore-application-stop-failures | --no-ignore-application-stop-failures`: (Opsional) Apakah Anda ingin penerapan ke instance melanjutkan ke peristiwa siklus hidup `BeforeInstall` penerapan jika peristiwa siklus hidup `ApplicationStop` penerapan gagal.
- `--description`: (Opsional) Deskripsi untuk penerapan.
- `--file-exists-behavior`: (Opsional) Sebagai bagian dari proses penerapan, CodeDeploy agen menghapus dari setiap instance semua file yang diinstal oleh penerapan terbaru. Pilih apa yang terjadi ketika file yang bukan bagian dari penerapan sebelumnya muncul di lokasi penyebaran target.
- `--target-instances`: Untuk blue/green penerapan, informasi tentang instans yang termasuk dalam lingkungan pengganti dalam blue/green penerapan, termasuk nama satu atau beberapa grup Penskalaan Otomatis Amazon EC2, atau kunci filter tag, jenis, dan nilai yang digunakan untuk mengidentifikasi instans Amazon EC2.

> **Note**
>
> Gunakan sintaks ini sebagai bagian dari create-deployment panggilan untuk menentukan informasi tentang revisi di Amazon S3 langsung pada baris perintah. (Itu `version` dan `eTag` opsional.)
>
> ```
> --s3-location bucket=string,key=string,bundleType=tar|tgz|
> zip,version=string,eTag=string
> ```
>
> Gunakan sintaks ini sebagai bagian dari create-deployment panggilan untuk menentukan informasi tentang revisi GitHub langsung pada baris perintah:
>
> ```
> --github-location repository=string,commitId=string
> ```

Untuk mendapatkan informasi tentang revisi yang sudah didorong, panggil perintah list-application-revisions.

Untuk melacak status penerapan Anda, lihatLihat CodeDeploy detail penerapan.

referensi perintah create-deployment

Struktur dan opsi perintah berikut tersedia untuk `create-deployment` perintah. Untuk informasi selengkapnya, lihat referensi create-deployment di Command Reference.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>]
```

Connect CodeDeploy aplikasi ke GitHub repositori

Sebelum Anda dapat menyebarkan aplikasi dari GitHub repositori untuk pertama kalinya menggunakan AWS CLI, Anda harus terlebih dahulu memberikan CodeDeploy izin untuk berinteraksi dengan GitHub atas nama akun Anda. GitHub Langkah ini harus diselesaikan sekali untuk setiap aplikasi menggunakan CodeDeploy konsol.

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

> **Note**
>
> Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Pilih Aplikasi.
3. Dari Aplikasi, pilih aplikasi yang ingin Anda tautkan ke akun GitHub pengguna Anda dan pilih Deploy aplikasi.

> **Note**
>
> Anda tidak membuat penerapan. Ini saat ini satu-satunya cara untuk memberikan CodeDeploy izin untuk berinteraksi dengan GitHub atas nama akun GitHub pengguna Anda.

4. Di samping Jenis repositori, pilih Revisi aplikasi saya disimpan di. GitHub
5. Pilih Connect to GitHub.

> **Note**
>
> Jika Anda melihat link Connect to GitHub account lain:
> Anda mungkin telah berwenang CodeDeploy untuk berinteraksi dengan GitHub atas nama GitHub akun lain untuk aplikasi tersebut.
> Anda mungkin telah mencabut otorisasi CodeDeploy untuk berinteraksi dengan GitHub atas nama GitHub akun yang masuk untuk semua aplikasi yang ditautkan ke dalam. CodeDeploy
> Untuk informasi selengkapnya, lihat GitHub otentikasi dengan aplikasi di CodeDeploy.

6. Jika Anda belum masuk GitHub, ikuti petunjuk di halaman Masuk.
7. Pada halaman Otorisasi aplikasi, pilih Otorisasi aplikasi.
8. Sekarang yang CodeDeploy memiliki izin, pilih Batal, dan lanjutkan dengan langkah-langkah masukMembuat penerapan Platform EC2/On-Premises Komputasi (CLI).

Buat blue/green penyebaran Amazon ECS melalui CloudFormation

Anda dapat menggunakan AWS CloudFormation untuk mengelola blue/green penyebaran Amazon ECS melalui. CodeDeploy Anda menghasilkan penyebaran Anda dengan mendefinisikan sumber daya hijau dan biru Anda dan menentukan pengaturan perutean dan stabilisasi lalu lintas yang akan digunakan. CloudFormation Topik ini mencakup perbedaan antara blue/green penerapan Amazon ECS yang dikelola oleh CodeDeploy dan penerapan yang dikelola oleh. CloudFormation

Untuk petunjuk langkah demi langkah tentang penggunaan CloudFormation untuk mengelola blue/green penerapan Amazon ECS, lihat Mengotomatiskan penerapan ECS melalui blue/green penggunaan di Panduan Pengguna. CodeDeploy AWS CloudFormationAWS CloudFormation

> **Note**
>
> Mengelola blue/green penerapan Amazon ECS tidak CloudFormation tersedia di Wilayah Asia Pasifik (Osaka).

Perbedaan antara blue/green penerapan Amazon ECS melalui dan CodeDeploy CloudFormation

Templat CloudFormation tumpukan memodelkan sumber daya dan infrastruktur terkait tugas Amazon ECS, dan juga opsi konfigurasi untuk penerapan. Jadi ada perbedaan antara penerapan Amazon ECS standar dan blue/green blue/green penerapan yang dibuat melalui. CloudFormation

Tidak seperti blue/green penerapan Amazon ECS standar, Anda tidak memodelkan atau membuat yang berikut secara manual:

- Anda tidak membuat AWS CodeDeploy aplikasi dengan menentukan nama yang secara unik mewakili apa yang ingin Anda terapkan.
- Anda tidak membuat grup AWS CodeDeploy penerapan.
- Anda tidak menentukan file spesifikasi aplikasi (AppSpec file). Informasi yang biasanya dikelola dengan AppSpec file, seperti opsi konfigurasi tertimbang atau peristiwa siklus hidup, dikelola oleh hook. `AWS::CodeDeploy::BlueGreen`

Tabel ini merangkum perbedaan alur kerja tingkat tinggi antara jenis penerapan.

Fungsi	blue/green Penerapan standar	Blue/green penyebaran melalui CloudFormation
Tentukan cluster Amazon ECS, layanan Amazon ECS, Application Load Balancer atau Network Load Balancer, Production listener, test listener, dan dua grup target.	Buat grup CodeDeploy penyebaran yang menentukan sumber daya ini.	Buat CloudFormation template untuk memodelkan sumber daya ini.
Tentukan perubahan yang akan diterapkan.	Buat CodeDeploy aplikasi.	Buat CloudFormation template yang menentukan gambar kontainer.
Tentukan definisi tugas Amazon ECS, nama kontainer , dan port kontainer.	Buat AppSpec file yang menentukan sumber daya ini.	Buat CloudFormation template untuk memodelkan sumber daya ini.
Tentukan opsi pemindahan lalu lintas penerapan dan kait peristiwa siklus hidup.	Buat AppSpec file yang menentukan opsi ini.	Buat CloudFormation template yang menggunakan parameter `AWS::CodeDeploy::BlueGreen` hook untuk menentukan opsi ini.
CloudWatch alarm.	Buat CloudWatch alarm yang memicu rollback.	Konfigurasikan CloudWatch alarm di tingkat CloudFormation tumpukan yang memicu rollback.
Rollback/redeployment.	Tentukan opsi rollback dan redeployment.	Batalkan pembaruan tumpukan di CloudFormation.

Memantau blue/green penyebaran Amazon ECS melalui CloudFormation

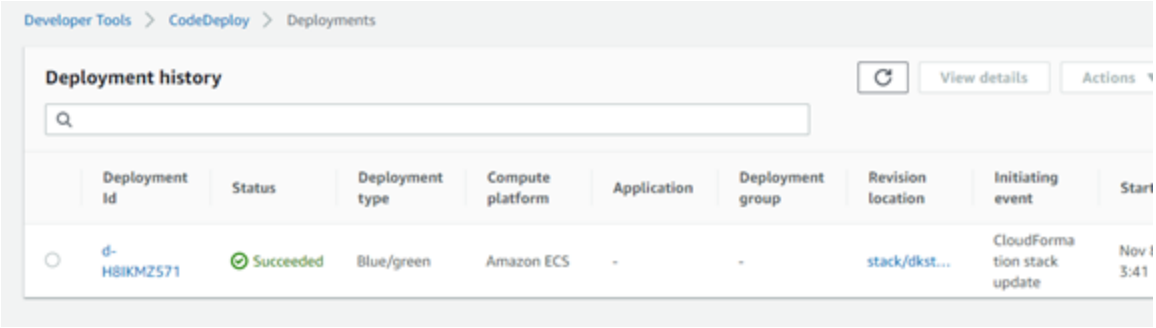
Anda dapat memantau blue/green penerapan melalui CloudFormation dan. CodeDeploy Untuk informasi tentang pemantauan melalui CloudFormation, lihat Memantau blue/green peristiwa CloudFormation di Panduan AWS CloudFormation Pengguna.

Untuk melihat status penerapan blue/green penerapan di CodeDeploy

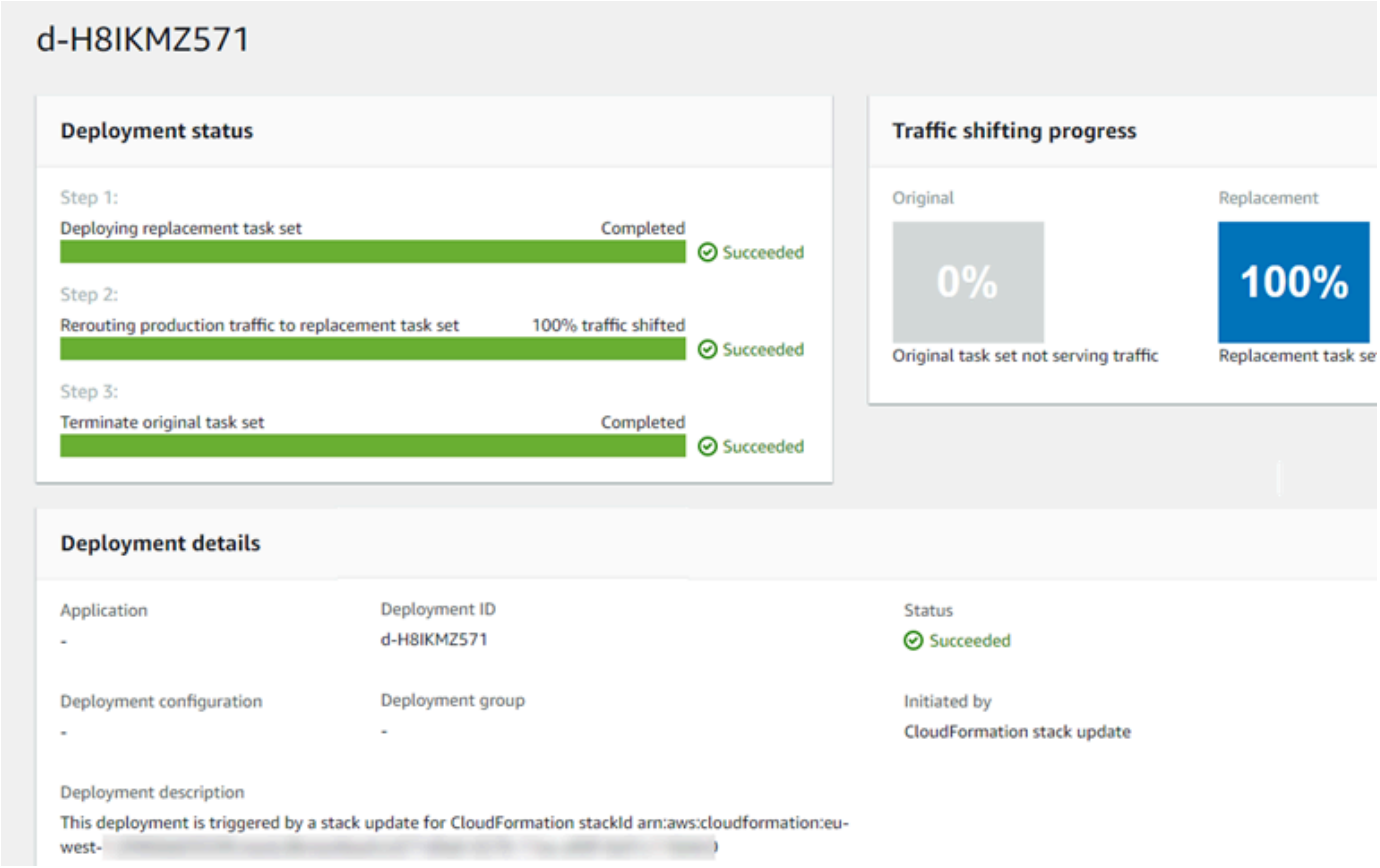
1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di Deployment, penerapan yang dipicu oleh pembaruan CloudFormation tumpukan muncul. Pilih penerapan untuk melihat riwayat Deployment.



3. Pilih penyebaran untuk melihat status pergeseran lalu lintas. Perhatikan bahwa grup aplikasi dan penyebaran tidak dibuat.



4. Berikut ini berlaku untuk memutar kembali atau menghentikan penerapan:

   - Penerapan yang berhasil muncul CodeDeploy dan menunjukkan bahwa penerapan diprakarsai oleh. CloudFormation
   - Jika Anda ingin menghentikan dan memutar kembali penerapan, Anda harus membatalkan pembaruan tumpukan di CloudFormation.

Lihat CodeDeploy detail penerapan

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk melihat detail tentang penerapan yang terkait dengan akun Anda AWS .

> **Note**
>
> Anda dapat melihat log EC2/On-Premises penerapan pada instans Anda di lokasi berikut:
>
> - Amazon Linux, RHEL, dan Server Ubuntu: `/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`

- Windows Server: C:\ProgramData\ Amazon\ CodeDeploy < DEPLOYMENT-GROUP-ID >< DEPLOYMENT-ID >\ log\ scripts.log

Untuk informasi selengkapnya, lihat Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance.

Topik

- Lihat detail penerapan (konsol)
- Lihat detail penerapan (CLI)

Lihat detail penerapan (konsol)

Untuk menggunakan CodeDeploy konsol untuk melihat detail penerapan:

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Deployments.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

3. Untuk melihat detail selengkapnya untuk satu penerapan, dalam riwayat Deployment, pilih ID penerapan atau pilih tombol di sebelah ID penerapan, lalu pilih Lihat.

Lihat detail penerapan (CLI)

Untuk menggunakan AWS CLI untuk melihat rincian penyebaran, panggil `get-deployment` perintah atau `batch-get-deployments` perintah. Anda dapat memanggil `list-deployments` perintah untuk mendapatkan daftar ID penyebaran unik untuk digunakan sebagai input ke `get-deployment` perintah dan perintah. `batch-get-deployments`

Untuk melihat detail tentang penerapan tunggal, panggil perintah get-deployment, dengan menentukan pengenal penerapan unik. Untuk mendapatkan ID penerapan, panggil perintah list-deployments.

Untuk melihat detail tentang beberapa penerapan, panggil perintah batch-get-deployments, dengan menentukan beberapa pengidentifikasi penerapan unik. Untuk mendapatkan ID penerapan, panggil perintah list-deployments.

Untuk melihat daftar ID penerapan, panggil perintah list-deployments, dengan menentukan:

- Nama aplikasi yang terkait dengan penyebaran. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran yang terkait dengan penyebaran. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.
- Secara opsional, apakah akan menyertakan detail tentang penerapan berdasarkan status penerapannya. (Jika tidak ditentukan, semua penerapan yang cocok akan dicantumkan, terlepas dari status penerapannya.)
- Secara opsional, apakah akan menyertakan detail tentang penerapan berdasarkan waktu mulai pembuatan penerapan atau waktu akhir, atau keduanya. (Jika tidak ditentukan, semua penerapan yang cocok akan dicantumkan, terlepas dari waktu pembuatannya.)

Lihat data log untuk CodeDeploy EC2/On-Premises penerapan

Anda dapat melihat data log yang dibuat oleh CodeDeploy penerapan dengan menyiapkan CloudWatch agen Amazon untuk melihat data agregat di CloudWatch konsol atau dengan masuk ke instance individual untuk meninjau file log.

> **Note**
>
> Log tidak didukung untuk penerapan AWS Lambda atau Amazon ECS. Mereka dapat dibuat hanya untuk EC2/On-Premises penerapan.

Topik

- Lihat data file log di CloudWatch konsol Amazon
- Lihat file log pada sebuah instance

Lihat data file log di CloudWatch konsol Amazon

Saat CloudWatch agen Amazon diinstal pada instance, data penerapan untuk semua penerapan ke instance tersebut akan tersedia untuk dilihat di konsol. CloudWatch Untuk mempermudah, sebaiknya gunakan CloudWatch untuk memonitor file log secara terpusat alih-alih melihatnya secara instance per instance. Untuk informasi selengkapnya, lihat Kirim log CodeDeploy agen ke CloudWatch.

Lihat file log pada sebuah instance

Untuk melihat data log penerapan untuk instance individual, Anda dapat masuk ke instans dan menelusuri informasi tentang kesalahan atau peristiwa penerapan lainnya.

Topik

- Untuk melihat file log penerapan di Amazon Linux, RHEL, dan instance Server Ubuntu
- Untuk melihat berkas log penyebaran pada instance Windows Server

Untuk melihat file log penerapan di Amazon Linux, RHEL, dan instance Server Ubuntu

Pada instans Amazon Linux, RHEL, dan Ubuntu Server, log penerapan disimpan di lokasi berikut:

`/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`

Untuk melihat atau menganalisis log penerapan di Amazon Linux, RHEL, dan instance Server Ubuntu, masuk ke instance, lalu ketik perintah berikut untuk membuka file log CodeDeploy agen:

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Untuk versi 2.0.x dan yang lebih baru, log agen berputar setiap hari atau ketika file mencapai 64 MB (dapat dikonfigurasi). Arsip yang diputar diberi nama `codedeploy-agent.YYYYMMDD.log` (dengan akhiran numerik untuk rotasi hari yang sama) dan dipangkas setelah tujuh hari. Script log (`scripts.log`) diputar ukuran pada 64 MB dengan hingga 8 file yang diarsipkan (dapat dikonfigurasi).

Ketik perintah berikut untuk menelusuri file log untuk pesan kesalahan:

Perintah	Hasil
& ERROR	Tampilkan hanya pesan kesalahan dalam file log. Gunakan satu spasi sebelum dan sesudah kata**ERROR**.
/ ERROR	Cari pesan kesalahan berikutnya.[1]
? ERROR	Cari pesan kesalahan sebelumnya.[2] Gunakan satu spasi sebelum dan sesudah kata. **ERROR**
G	Pergi ke akhir file log.
g	Pergi ke awal file log.
q	Keluar dari file log.
h	Pelajari tentang perintah tambahan.

[1] Setelah Anda mengetik**/ ERROR** , ketik **n** untuk pesan kesalahan berikutnya. Ketik **N** untuk pesan kesalahan sebelumnya.

[2] Setelah Anda mengetik**? ERROR** , ketik **n** untuk pesan kesalahan berikutnya, atau ketik **N** untuk pesan kesalahan sebelumnya.

Anda juga dapat mengetik perintah berikut untuk membuka file log CodeDeploy skrip:

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/
scripts.log
```

Ketik perintah berikut untuk menelusuri file log untuk pesan kesalahan:

Perintah	Hasil
&stderr	Tampilkan hanya pesan kesalahan dalam file log.
/stderr	Cari pesan kesalahan berikutnya.[1]
?stderr	Cari pesan kesalahan sebelumnya.[2]
G	Pergi ke akhir file log.
g	Pergi ke awal file log.
q	Keluar dari file log.
h	Pelajari tentang perintah tambahan.

[1]Setelah Anda mengetik**/stderr**, ketik **n** untuk meneruskan pesan kesalahan berikutnya. Ketik **N** pesan kesalahan sebelumnya ke belakang.

[2] Setelah Anda mengetik**?stderr**, **n** ketik pesan kesalahan berikutnya ke belakang. Ketik **N** untuk meneruskan pesan kesalahan sebelumnya.

Untuk melihat berkas log penyebaran pada instance Windows Server

CodeDeploy file log agen: Pada instance Windows Server, file log CodeDeploy agen disimpan di lokasi berikut:

Untuk versi 2.0.x dan yang lebih baru:

```
C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Untuk versi 1.8.x dan sebelumnya:

```
C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Untuk melihat atau menganalisis file log CodeDeploy agen pada instance Windows Server, masuk ke instance, lalu ketik perintah berikut untuk membuka file:

Untuk versi 2.0.x dan yang lebih baru:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent.log
```

Untuk versi 1.8.x dan sebelumnya:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Untuk menelusuri file log untuk pesan kesalahan, tekan CTRL+F, ketik**ERROR**, lalu tekan Enter untuk menemukan kesalahan pertama.

CodeDeploy file log skrip: Pada instance Windows Server, log penyebaran disimpan di lokasi berikut:

`C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\logs\scripts.log`

Di mana:

- *`deployment-group-id`*adalah string seperti `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *`deployment-id`*adalah pengenal seperti `d-12EXAMPLE`

Ketik perintah berikut untuk membuka file log CodeDeploy skrip:

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs
\scripts.log
```

Untuk versi 2.0.x dan yang lebih baru, log agen berputar setiap hari atau ketika file mencapai 64 MB (dapat dikonfigurasi). Log skrip diputar ukuran pada 64 MB dengan hingga 8 file yang diarsipkan (dapat dikonfigurasi).

Untuk menelusuri file log untuk pesan kesalahan, tekan CTRL+F, ketik**stderr**, lalu tekan Enter untuk menemukan kesalahan pertama.

Hentikan penerapan dengan CodeDeploy

Anda dapat menggunakan CodeDeploy konsol, API AWS CLI, atau CodeDeploy API untuk menghentikan penerapan yang terkait dengan akun Anda AWS .

> **Warning**
>
> Menghentikan EC2/On-Premises penerapan dapat meninggalkan beberapa atau semua instance di grup penerapan Anda dalam status penerapan tak tentu. Untuk informasi selengkapnya, lihat Penerapan yang dihentikan dan gagal.

Anda dapat menghentikan penerapan atau menghentikan dan memutar kembali penerapan.

- Hentikan penerapan (konsol)
- Hentikan penerapan (CLI)

> **Note**
>
> Jika penerapan Anda adalah blue/green penerapan CloudFormation, Anda tidak dapat melakukan tugas ini di konsol. CodeDeploy Pergi ke CloudFormation konsol untuk melakukan tugas ini.

Hentikan penerapan (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Deployments.

   > **Note**
   >
   > Jika tidak ada entri yang ditampilkan, pastikan wilayah yang benar dipilih. Pada bilah navigasi, di pemilih wilayah, pilih salah satu wilayah yang tercantum di Wilayah dan Titik Akhir di. Referensi Umum AWS CodeDeploy hanya didukung di wilayah ini.

3. Pilih penerapan yang ingin Anda hentikan, lakukan salah satu hal berikut:

1. Pilih Stop deployment untuk menghentikan penerapan tanpa rollback.
2. Pilih Berhenti dan putar kembali penerapan untuk menghentikan dan memutar kembali penerapan

Untuk informasi selengkapnya, lihat Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.

> **Note**
>
> Jika Stop deployment dan Stop and roll back deployment tidak tersedia, penerapan telah berkembang ke titik di mana tidak dapat dihentikan.

Hentikan penerapan (CLI)

Panggil perintah stop-deployment, dengan menentukan ID deployment. Untuk melihat daftar ID penerapan, panggil perintah list-deployments.

Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy

CodeDeploy memutar kembali penerapan dengan menerapkan kembali revisi aplikasi yang sebelumnya diterapkan sebagai penerapan baru. Penerapan rolled-back ini secara teknis merupakan penerapan baru, dengan ID penerapan baru, bukan versi yang dipulihkan dari penerapan sebelumnya.

Penerapan dapat diputar kembali secara otomatis atau manual.

Topik

- Rollback otomatis
- Rollback manual
- Alur kerja rollback dan redeployment
- Perilaku rollback dengan konten yang ada

Rollback otomatis

Anda dapat mengonfigurasi grup penerapan atau penerapan untuk memutar kembali secara otomatis saat penerapan gagal atau saat ambang batas pemantauan yang Anda tentukan terpenuhi. Dalam hal ini, versi baik terakhir yang diketahui dari revisi aplikasi digunakan. Anda mengonfigurasi rollback otomatis saat membuat aplikasi atau membuat atau memperbarui grup penyebaran.

Saat membuat penerapan baru, Anda juga dapat memilih untuk mengganti konfigurasi rollback otomatis yang ditentukan untuk grup penerapan.

> **Note**
>
> Anda dapat menggunakan Amazon Simple Notification Service untuk menerima notifikasi setiap kali penerapan digulirkan kembali secara otomatis. Untuk informasi, lihat Monitoring Deployments with Amazon SNS Event Notifications.

Untuk informasi selengkapnya tentang mengonfigurasi rollback otomatis, lihat. Mengonfigurasi opsi lanjutan untuk grup deployment

Rollback manual

Jika Anda belum menyiapkan rollback otomatis, Anda dapat memutar kembali penerapan secara manual dengan membuat penerapan baru yang menggunakan revisi aplikasi yang digunakan sebelumnya dan mengikuti langkah-langkah untuk menerapkan revisi. Anda dapat melakukan ini jika aplikasi telah masuk ke keadaan yang tidak diketahui. Daripada menghabiskan banyak waktu pemecahan masalah, Anda dapat menerapkan ulang aplikasi ke status kerja yang diketahui. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy.

> **Note**
>
> Jika Anda menghapus instance dari grup penyebaran, CodeDeploy tidak menghapus instalan apa pun yang mungkin telah diinstal pada instance itu.

Alur kerja rollback dan redeployment

Ketika rollback otomatis dimulai, atau ketika Anda secara manual memulai redeployment atau manual rollback, CodeDeploy pertama-tama mencoba untuk menghapus dari setiap instance

yang berpartisipasi semua file yang terakhir berhasil diinstal. CodeDeploy melakukan ini dengan memeriksa file pembersihan:

`/opt/codedeploy-agent/deployment-root/deployment-instructions/`*`deployment-group-ID`*`-cleanup`file (untuk Amazon Linux, Ubuntu Server, dan instans RHEL)

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`*`deployment-group-ID`*`-cleanup`berkas (untuk instance Windows Server)

Jika ada, CodeDeploy gunakan file pembersihan untuk menghapus dari instance semua file yang terdaftar sebelum memulai penerapan baru.

Misalnya, dua file teks pertama dan dua file skrip sudah digunakan ke instans Amazon EC2 yang menjalankan Windows Server, dan skrip membuat dua file teks lagi selama peristiwa siklus hidup penerapan:

```
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)
```

File pembersihan hanya akan mencantumkan dua file teks pertama dan dua file skrip:

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
c:\temp\d.bat
```

Sebelum penyebaran baru, CodeDeploy akan menghapus hanya dua file teks pertama dan dua file skrip, meninggalkan dua file teks terakhir yang tidak tersentuh:

```
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
```

Sebagai bagian dari proses ini, tidak CodeDeploy akan mencoba mengembalikan atau mendamaikan tindakan apa pun yang diambil oleh skrip apa pun dalam penerapan sebelumnya selama pemindahan berikutnya, baik secara manual maupun otomatis. Misalnya, jika `d.bat` file `c.bat` dan berisi logika untuk tidak membuat ulang `f.txt` file `e.txt` dan jika sudah ada, maka versi lama `e.txt` dan `f.txt` akan tetap tidak tersentuh setiap kali CodeDeploy berjalan `c.bat` dan `d.bat` dalam penerapan berikutnya. Anda dapat menambahkan logika `d.bat` ke `c.bat` dan untuk selalu memeriksa dan menghapus versi lama `e.txt` dan `f.txt` sebelum membuat yang baru.

Perilaku rollback dengan konten yang ada

Sebagai bagian dari proses penyebaran, CodeDeploy agen menghapus dari setiap instance semua file yang diinstal oleh penerapan terbaru. Jika file yang bukan bagian dari penerapan sebelumnya muncul di lokasi penyebaran target, Anda dapat memilih apa yang CodeDeploy dilakukannya selama penerapan berikutnya:

- Gagal penerapan — Kesalahan dilaporkan dan status penerapan diubah menjadi Gagal.
- Timpa konten — Versi file dari revisi aplikasi menggantikan versi yang sudah ada pada instance.
- Mempertahankan konten — File di lokasi target disimpan dan versi dalam revisi aplikasi tidak disalin ke instance.

Anda dapat memilih perilaku ini saat membuat penerapan. Jika membuat penerapan di konsol, lihatMembuat penerapan Platform EC2/On-Premises Komputasi (konsol). Jika membuat penerapan dengan AWS CLI, lihatMembuat penerapan Platform EC2/On-Premises Komputasi (CLI).

Anda dapat memilih untuk menyimpan file yang ingin Anda menjadi bagian dari penerapan berikutnya tanpa harus menambahkannya ke paket revisi aplikasi. Misalnya, Anda dapat mengunggah file langsung ke instance yang diperlukan untuk penerapan tetapi tidak ditambahkan ke bundel revisi aplikasi. Atau Anda dapat mengunggah file ke instance jika aplikasi Anda sudah berada di lingkungan produksi Anda tetapi Anda ingin menggunakannya CodeDeploy untuk pertama kalinya untuk menerapkannya.

Dalam kasus rollback, di mana revisi aplikasi terbaru yang berhasil diterapkan diterapkan kembali karena kegagalan penerapan, opsi penanganan konten untuk penerapan terakhir yang berhasil diterapkan pada penerapan rollback.

Namun, jika penerapan yang gagal dikonfigurasi untuk menimpa, alih-alih mempertahankan file, hasil yang tidak terduga dapat terjadi selama rollback. Secara khusus, file yang Anda harapkan

untuk disimpan mungkin akan dihapus oleh penerapan yang gagal. File tidak ada pada instance saat penerapan rollback berjalan.

Dalam contoh berikut, ada tiga penerapan. File apa pun yang ditimpa (dihapus) selama penerapan kedua yang gagal tidak lagi tersedia (tidak dapat dipertahankan) saat revisi aplikasi 1 diterapkan lagi selama penerapan 3:

Deployment	Revisi aplikasi	Opsi menimpa konten	Status deployment	Perilaku dan hasil
penyebaran 1	revisi aplikasi 1	MELESTARIKAN	Berhasil	CodeDeploy mendeteksi file di lokasi target yang tidak digunakan oleh penerapan sebelumnya. File-file ini mungkin ditempatkan di sana dengan sengaja untuk menjadi bagian dari penyebaran saat ini. Mereka disimpan dan dicatat sebagai bagian dari paket penyebaran saat ini.
penyebaran 2	revisi aplikasi 2	MENIMPA	Gagal	Selama proses penyebaran, CodeDeploy menghapus semua file yang merupakan

Deployment	Revisi aplikasi	Opsi menimpa konten	Status deployment	Perilaku dan hasil
				bagian dari penerapan sukses sebelumnya. Ini termasuk file yang disimpan selama penerapan 1. Namun, penerapan gagal karena alasan yang tidak terkait.

Deployment	Revisi aplikasi	Opsi menimpa konten	Status deployment	Perilaku dan hasil
penyebaran 3	revisi aplikasi 1	MELESTARI KAN		Karena auto rollback diaktifka n untuk grup penyebaran atau penyebara n, CodeDeplo y menyebarkan revisi aplikasi bagus terakhir yang diketahui, revisi aplikasi 1. Namun, file yang ingin Anda simpan dalam penerapan 1 telah dihapus sebelum penerapan 2 gagal dan tidak dapat diambil oleh. AWS CodeDeploy Anda dapat menambahk annya ke instance sendiri jika diperluka n untuk revisi aplikasi 1, atau Anda dapat membuat revisi aplikasi baru.

Menyebarkan aplikasi di tempat yang berbeda AWS akun

Organizations umumnya memiliki beberapa AWS akun yang mereka gunakan untuk tujuan yang berbeda (misalnya, satu untuk tugas administrasi sistem dan satu lagi untuk pengembangan, pengujian, dan tugas produksi atau yang terkait dengan lingkungan pengembangan dan pengujian dan lainnya terkait dengan lingkungan produksi).

Meskipun Anda mungkin melakukan pekerjaan terkait di akun yang berbeda, grup CodeDeploy penerapan, dan instans Amazon EC2 yang mereka gunakan terkait erat dengan akun tempat mereka dibuat. Anda tidak dapat, misalnya, menambahkan instance yang Anda luncurkan di satu akun ke grup penyebaran di akun lain.

Asumsikan Anda memiliki dua AWS akun: akun pengembangan dan akun produksi Anda. Anda bekerja terutama di akun pengembangan, tetapi Anda ingin dapat memulai penerapan di akun produksi Anda tanpa set lengkap kredensi di sana atau tanpa harus keluar dari akun pengembangan dan masuk ke akun produksi.

Setelah mengikuti langkah-langkah konfigurasi lintas-akun, Anda dapat memulai penerapan yang dimiliki oleh akun organisasi Anda yang lain tanpa memerlukan set lengkap kredensional untuk akun lain tersebut. Anda melakukan ini, sebagian, dengan menggunakan kemampuan yang disediakan oleh AWS Security Token Service (AWS STS) yang memberi Anda akses sementara ke akun itu.

Langkah 1: Buat bucket S3 di salah satu akun

Baik dalam akun pengembangan atau akun produksi:

- Jika Anda belum melakukannya, buat bucket Amazon S3 tempat revisi aplikasi untuk akun produksi akan disimpan. Untuk selengkapnya, lihat Membuat Bucket di Amazon S3. Anda bahkan dapat menggunakan bucket dan revisi aplikasi yang sama untuk kedua akun, menerapkan file yang sama ke lingkungan produksi yang Anda uji dan verifikasi di akun pengembangan Anda.

Langkah 2: Berikan izin bucket Amazon S3 ke profil instans IAM akun produksi

Jika bucket Amazon S3 yang Anda buat di langkah 1 ada di akun produksi Anda, langkah ini tidak diperlukan. Peran yang Anda asumsikan nanti sudah memiliki akses ke bucket ini karena juga ada di akun produksi.

Jika Anda membuat bucket Amazon S3 di akun pengembangan, lakukan hal berikut:

- Di akun produksi, buat profil instans IAM. Untuk informasi, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.

  > **Note**
  >
  > Catat ARN untuk profil instans IAM ini. Anda harus menambahkannya ke kebijakan cross-bucket yang Anda buat selanjutnya.

- Di akun pengembangan, berikan akses ke bucket Amazon S3 yang Anda buat di akun pengembangan ke profil instans IAM yang baru saja Anda buat di akun produksi Anda. Untuk selengkapnya, lihat Contoh 2: Pemilik bucket yang memberikan izin bucket lintas akun.

  Perhatikan hal-hal berikut saat Anda menyelesaikan proses pemberian izin bucket lintas akun:

  - Dalam contoh panduan, Akun A mewakili akun pengembangan Anda dan Akun B mewakili akun produksi Anda.
  - Saat Anda melakukan tugas Akun A (akun pengembangan), ubah kebijakan bucket berikut untuk memberikan izin lintas akun alih-alih menggunakan kebijakan sampel yang disediakan dalam panduan.

    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 mewakili nomor akun dari akun produksi tempat Anda baru saja membuat profil instans IAM.

role-name mewakili nama profil instans IAM yang baru saja Anda buat.

bucket-name mewakili nama bucket yang Anda buat di langkah 1. Pastikan untuk menyertakan `/*` nama bucket Anda untuk memberikan akses ke setiap file di dalam bucket.

Langkah 3: Buat sumber daya dan peran lintas akun di akun produksi

Di akun produksi Anda:

- Buat CodeDeploy sumber daya Anda — aplikasi, grup penerapan, konfigurasi penerapan, instans Amazon EC2, profil instans Amazon EC2, peran layanan, dan sebagainya — menggunakan petunjuk dalam panduan ini.
- Buat peran tambahan, peran IAM lintas akun, yang dapat diasumsikan oleh pengguna di akun pengembangan Anda untuk melakukan CodeDeploy operasi di akun produksi ini.

  Gunakan Walkthrough: Delegasikan akses di seluruh AWS akun menggunakan peran IAM sebagai panduan untuk membantu Anda membuat peran lintas akun. Alih-alih menambahkan izin sampel dalam panduan ke dokumen kebijakan Anda, Anda harus melampirkan, setidaknya, dua kebijakan yang AWS diberikan berikut ke peran tersebut:

  - `AmazonS3FullAccess`: Diperlukan hanya jika bucket S3 ada di akun pengembangan. Menyediakan peran akun produksi yang diasumsikan dengan akses penuh ke layanan dan sumber daya Amazon S3 di akun pengembangan, tempat revisi disimpan.
  - `AWSCodeDeployDeployerAccess`: Memungkinkan pengguna untuk mendaftar dan menyebarkan revisi.

  Jika Anda ingin membuat dan mengelola grup penerapan dan tidak hanya memulai penerapan, tambahkan `AWSCodeDeployFullAccess` kebijakan sebagai ganti kebijakan. `AWSCodeDeployDeployerAccess` Untuk informasi selengkapnya tentang menggunakan kebijakan terkelola IAM guna memberikan izin CodeDeploy tugas, lihat. AWS kebijakan terkelola (standar) untuk CodeDeploy

  Anda dapat melampirkan kebijakan tambahan jika ingin melakukan tugas di AWS layanan lain saat menggunakan peran lintas akun ini.

Important

Saat Anda membuat peran IAM lintas akun, catat detail yang Anda perlukan untuk mendapatkan akses ke akun produksi.
Untuk menggunakan Konsol Manajemen AWS untuk beralih peran, Anda harus menyediakan salah satu dari berikut ini:

- URL untuk mengakses akun produksi dengan kredenal peran yang diasumsikan. Anda akan menemukan URL di halaman Ulasan, yang ditampilkan di akhir proses pembuatan peran lintas akun.
- Nama peran lintas akun dan nomor ID akun atau alias.

Untuk menggunakan AWS CLI untuk beralih peran, Anda harus menyediakan yang berikut:

- ARN dari peran lintas akun yang akan Anda asumsikan.

Langkah 4: Unggah revisi aplikasi ke bucket Amazon S3

Di akun tempat Anda membuat bucket Amazon S3:

- Unggah revisi aplikasi Anda ke bucket Amazon S3. Untuk informasi, lihat Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan).

Langkah 5: Asumsikan peran lintas akun dan gunakan aplikasi

Di akun pengembangan, Anda dapat menggunakan AWS CLI atau Konsol Manajemen AWS untuk mengambil peran lintas akun dan memulai penyebaran di akun produksi.

Untuk petunjuk tentang cara menggunakan Konsol Manajemen AWS untuk beralih peran dan memulai penerapan, lihat Beralih ke role ()Konsol Manajemen AWS dan. Membuat penerapan Platform EC2/On-Premises Komputasi (konsol)

Untuk petunjuk tentang cara menggunakan peran lintas-akun dan memulai penerapan, lihat Beralih ke peran IAM () dan. AWS CLI AWS Command Line InterfaceMembuat penerapan Platform EC2/On-Premises Komputasi (CLI)

Untuk informasi selengkapnya tentang mengasumsikan peran AWS STS, lihat AssumeRoledi Panduan AWS Security Token Service Pengguna dan peran asumsi di Referensi Perintah.AWS CLI

Topik terkait:

- CodeDeploy: Menyebarkan dari akun pengembangan ke akun produksi

Gunakan CodeDeploy agen untuk memvalidasi paket penerapan pada mesin lokal

Dengan menggunakan CodeDeploy agen, Anda dapat menyebarkan konten pada instance yang Anda masuki. Ini memungkinkan Anda untuk menguji integritas file spesifikasi aplikasi (AppSpec file) yang ingin Anda gunakan dalam penyebaran dan konten yang ingin Anda gunakan.

Anda tidak perlu membuat grup aplikasi dan penyebaran. Jika Anda ingin menyebarkan konten yang disimpan pada instance lokal, Anda bahkan tidak memerlukan AWS akun. Untuk pengujian yang paling sederhana, Anda dapat menjalankan codedeploy-agent deploy-local perintah, tanpa menentukan opsi apa pun, di direktori yang berisi AppSpec file dan konten yang akan digunakan. Ada opsi untuk kasus uji lain di alat ini.

Dengan memvalidasi paket penerapan pada mesin lokal, Anda dapat:

- Uji integritas revisi aplikasi.
- Uji isi AppSpec file.
- Cobalah CodeDeploy untuk pertama kalinya dengan kode aplikasi yang ada.
- Terapkan konten dengan cepat saat Anda sudah masuk ke sebuah instans.

Anda dapat menggunakan konten deploy yang disimpan di instans lokal atau dalam jenis repositori jarak jauh yang didukung (bucket Amazon S3 atau repositori publik). GitHub

Prasyarat

Sebelum memulai penerapan lokal, selesaikan langkah-langkah berikut:

- Buat atau gunakan jenis instance yang didukung oleh CodeDeploy agen. Untuk informasi, lihat Sistem operasi yang didukung oleh CodeDeploy agen.

- Instal versi 2.0.0 atau yang lebih baru dari agen. CodeDeploy `deploy-local`Subperintah membutuhkan versi 2.0.0 atau yang lebih baru. Untuk versi 1.8.x dan sebelumnya, gunakan perintah mandiri sebagai gantinya`codedeploy-local`. Untuk informasi versi, lihat Riwayat versi CodeDeploy agen. Untuk informasi instalasi, lihatInstal CodeDeploy agen.
- Jika Anda menerapkan konten dari bucket GitHub atau repositori Amazon S3, sediakan pengguna untuk digunakan. CodeDeploy Untuk informasi, lihat Langkah 1: Menyiapkan.
- Jika Anda menerapkan revisi aplikasi dari bucket Amazon S3, buat bucket Amazon S3 di Wilayah tempat Anda bekerja dan terapkan kebijakan bucket Amazon S3 ke bucket. Kebijakan ini memberi instans Anda izin yang diperlukan untuk mengunduh revisi aplikasi.

  Misalnya, kebijakan bucket Amazon S3 berikut memungkinkan instans Amazon EC2 dengan profil instans IAM terlampir yang berisi `arn:aws:iam::444455556666:role/CodeDeployDemo` ARN untuk diunduh dari mana saja di bucket Amazon S3 bernama: `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"
                ]
            }
        }
    ]
}
```

  Kebijakan bucket Amazon S3 berikut memungkinkan instans lokal dengan pengguna IAM terkait yang berisi ARN `arn:aws:iam::444455556666:user/CodeDeployUser` dapat diunduh dari mana saja di bucket Amazon S3 bernama: `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"
                ]
            }
        }
    ]
}
```

Untuk informasi tentang cara membuat dan melampirkan kebijakan bucket Amazon S3, lihat contoh kebijakan Bucket.

- Jika Anda menerapkan revisi aplikasi dari bucket GitHub atau repositori Amazon S3, siapkan profil instans IAM dan lampirkan ke instance. Untuk informasi, lihatLangkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda,Buat instans Amazon EC2 untuk (CodeDeploy AWS CLI atau konsol Amazon EC2), danBuat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat).
- Jika Anda menyebarkan konten Anda dari GitHub, buat GitHub akun dan repositori publik. Untuk membuat GitHub akun, lihat Bergabung GitHub. Untuk membuat GitHub repositori, lihat Membuat repo.

> **Note**
>
> Repositori pribadi saat ini tidak didukung. Jika konten Anda disimpan dalam GitHub repositori pribadi, Anda dapat mengunduhnya ke instance dan menggunakan `--bundle-location` opsi untuk menentukan jalur lokalnya.

- Siapkan konten (termasuk AppSpec file) yang ingin Anda terapkan ke instance dan letakkan di instance lokal, di bucket Amazon S3, atau di GitHub repositori Anda. Untuk informasi, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.
- Jika Anda ingin menggunakan nilai selain default untuk opsi konfigurasi lainnya, buat file konfigurasi dan letakkan di instance (untuk instance `/etc/codedeploy-agent/conf/codedeployagent.yml` Amazon Linux, RHEL, atau Ubuntu Server atau `C:\ProgramData\Amazon\CodeDeploy\conf.yml` untuk instance Windows Server). Untuk informasi, lihat CodeDeploy referensi konfigurasi agen.

> **Note**
>
> Jika Anda menggunakan file konfigurasi di Amazon Linux, RHEL, atau instance Server Ubuntu, Anda harus:
>
> - Gunakan `:log_dir:` variabel `:root_dir:` dan untuk menentukan lokasi selain default untuk folder direktori root dan log deployment.
> - Gunakan `sudo` untuk menjalankan perintah CodeDeploy agen.

Buat penerapan lokal

Pada contoh di mana Anda ingin membuat penyebaran lokal, buka sesi terminal (Amazon Linux, RHEL, atau contoh Server Ubuntu) atau prompt perintah (Windows Server) untuk menjalankan perintah alat.

> **Note**
>
> codedeploy-agent deploy-localPerintah menggunakan `codedeploy-agent` biner di lokasi berikut:
>
> - Di Amazon Linux, RHEL, atau Server Ubuntu:`/opt/codedeploy-agent/bin/codedeploy-agent`.
> - Di Windows Server:`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-agent.exe`.
>
> Untuk kompatibilitas mundur, codedeploy-local perintah ini juga tersedia di lokasi yang sama:
>
> - Di Amazon Linux, RHEL, atau Ubuntu Server, perintahnya terletak di`/opt/codedeploy-agent/bin/codedeploy-local`.
> - Di Windows Server, perintahnya terletak di`C:\ProgramData\Amazon\CodeDeploy\bin\codedeploy-local.exe`.
>
> codedeploy-local [options]Sintaks terus bekerja dan menerima opsi yang sama seperti. codedeploy-agent deploy-local

Sintaks Perintah Dasar

```
codedeploy-agent deploy-local [options]
```

Sinopsis

```
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>]
```

Pilihan

-l, --bundle-lokasi

Lokasi bundel revisi aplikasi. Jika Anda tidak menentukan lokasi, alat menggunakan direktori yang sedang Anda kerjakan secara default. Jika Anda menentukan `--bundle-location`, maka Anda juga harus menentukan nilai `--type`.

Contoh format lokasi bundel:

- Instans Amazon Linux, RHEL, atau Server Ubuntu lokal: `/path/to/local/bundle.tgz`
- Contoh Windows Server lokal: `C:/path/to/local/bundle`
- Ember Amazon S3: `s3://amzn-s3-demo-bucket/bundle.tar`
- GitHub repositori: `https://github.com/`*`account-name`*`/`*`repository-name`*`/`

-t, --tipe

Format bundel revisi aplikasi. Jenis yang didukung meliputi`itgz`,`tar`,`zip`, dan`directory`. Jika Anda tidak menentukan jenis, alat ini menggunakan secara `directory` default. Jika Anda menentukan `--type`, maka Anda juga harus menentukan nilai `--bundle-location`.

-b, --file-exists-behavior

Menunjukkan bagaimana file ditangani yang sudah ada di lokasi target penerapan tetapi bukan bagian dari penerapan yang berhasil sebelumnya. Opsi termasuk DISALLOW, OVERWRITE, RESTAIN. Untuk informasi selengkapnya, lihat berkas ExistsBehavior di Referensi AWS CodeDeploy API.

-g, --deployment-group

Jalur ke folder yang merupakan lokasi target untuk konten yang akan digunakan. Jika Anda tidak menentukan folder, alat akan membuat folder bernama default-local-deployment-group di dalam direktori root deployment Anda. Untuk setiap penyebaran lokal yang Anda buat, alat ini membuat subdirektori di dalam folder ini dengan nama seperti local-1234 (di mana nomornya adalah ID proses).

-d, --deployment-group-name

Nama grup penyebaran diekspos ke skrip lifecycle-hook sebagai variabel lingkungan. `DEPLOYMENT_GROUP_NAME` Default ke `LocalFleet`.

-a, --nama aplikasi

Nama aplikasi terkena skrip lifecycle-hook sebagai variabel lingkungan. `APPLICATION_NAME` Default ke lokasi bundel.

-e, --event

Satu set kait peristiwa siklus hidup timpa yang ingin Anda jalankan, secara berurutan, bukan peristiwa yang Anda cantumkan dalam file. AppSpec Beberapa kait dapat ditentukan, dipisahkan dengan koma. Anda dapat menggunakan opsi ini jika:

- Anda ingin menjalankan serangkaian acara yang berbeda tanpa harus memperbarui AppSpec file.
- Anda ingin menjalankan satu event hook sebagai pengecualian untuk apa yang ada di AppSpec file, sepertiApplicationStop.

Jika Anda tidak menentukan DownloadBundledan Menginstal peristiwa dalam daftar penggantian, mereka akan berjalan sebelum semua kait acara yang Anda tentukan. Jika Anda menyertakan DownloadBundledan Menginstal dalam daftar `--events` opsi, mereka harus didahului hanya oleh peristiwa yang biasanya berjalan sebelum mereka dalam CodeDeploy penerapan. Untuk informasi, lihat AppSpec Bagian 'kait'.

-c, --agent-configuration-file

Lokasi file konfigurasi yang akan digunakan untuk penyebaran, jika Anda menyimpannya di lokasi selain default. File konfigurasi menentukan alternatif untuk nilai dan perilaku default lainnya untuk penerapan.

Secara default, file konfigurasi disimpan di `/etc/codedeploy-agent/conf/codedeployagent.yml` (Amazon Linux, RHEL, atau contoh Server Ubuntu) atau `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server). Untuk informasi selengkapnya, lihat CodeDeploy referensi konfigurasi agen.

-A, --appspec-nama file

Nama AppSpec file. Untuk penerapan lokal, nilai yang diterima adalah `appspec.yml` dan. `appspec.yaml` Secara default, AppSpec file tersebut dipanggil`appspec.yml`.

-h, --membantu

Menampilkan ringkasan konten bantuan.

-v, --versi

Menampilkan nomor versi alat.

Contoh

Berikut ini adalah contoh format perintah yang valid.

```
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
```

Terapkan bundel dari 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
```

Menyebarkan bundel dari GitHub repositori publik:

```
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
```

Terapkan bundel yang menentukan beberapa peristiwa siklus hidup:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/
bundle.tar --type tar --deployment-group my-deployment-group --events
 DownloadBundle,Install,ApplicationStart,HealthCheck
```

Menghentikan aplikasi yang digunakan sebelumnya menggunakan peristiwa ApplicationStop siklus hidup:

```
codedeploy-agent deploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --
deployment-group my-deployment-group --events ApplicationStop
```

Terapkan menggunakan ID grup penerapan tertentu:

```
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
```

Memantau penyebaran di CodeDeploy

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja CodeDeploy dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. Namun CodeDeploy, sebelum Anda mulai memantau, Anda harus membuat rencana pemantauan yang mencakup jawaban atas pertanyaan-pertanyaan berikut:

- Apa saja sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Langkah selanjutnya adalah menetapkan dasar untuk CodeDeploy kinerja normal di lingkungan Anda, dengan mengukur kinerja pada berbagai waktu dan dalam kondisi beban yang berbeda. Saat Anda memantau CodeDeploy, simpan data pemantauan historis sehingga Anda dapat membandingkannya dengan data kinerja saat ini, mengidentifikasi pola kinerja normal dan anomali kinerja, dan merancang metode untuk mengatasi masalah.

Misalnya, jika Anda menggunakan CodeDeploy, Anda dapat memantau status penerapan dan instance target. Ketika penerapan atau instance gagal, Anda mungkin perlu mengkonfigurasi ulang file spesifikasi aplikasi, menginstal ulang atau memperbarui CodeDeploy agen, memperbarui pengaturan dalam aplikasi atau grup penyebaran, atau membuat perubahan pada pengaturan instans atau file. AppSpec

Untuk menetapkan baseline, Anda harus, setidaknya, memantau item-item berikut:

- Acara dan status penyebaran
- Peristiwa dan status instance

Alat pemantauan otomatis

AWS menyediakan berbagai alat yang dapat Anda gunakan untuk memantau CodeDeploy. Anda dapat mengonfigurasi beberapa alat ini untuk melakukan pemantauan untuk Anda, sementara

beberapa alat memerlukan intervensi manual. Kami menyarankan agar Anda mengautomasi tugas pemantauan sebanyak mungkin.

Anda dapat menggunakan alat pemantauan otomatis berikut untuk menonton CodeDeploy dan melaporkan ketika ada sesuatu yang salah:

- CloudWatch Alarm Amazon — Tonton satu metrik selama periode waktu yang Anda tentukan, dan lakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Tindakannya adalah pemberitahuan yang dikirim ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Amazon EC2 Auto Scaling. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu; negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu. Untuk informasi selengkapnya, lihat Monitoring Deployments with Amazon CloudWatch Tools.

  Untuk informasi tentang memperbarui peran layanan agar berfungsi dengan pemantauan CloudWatch alarm, lihatBerikan CloudWatch izin untuk peran CodeDeploy layanan. Untuk informasi tentang menambahkan pemantauan CloudWatch alarm ke CodeDeploy operasi Anda, lihat Buat aplikasi dengan CodeDeployBuat grup penerapan dengan CodeDeploy, atauUbah pengaturan grup penerapan dengan CodeDeploy.

- Amazon CloudWatch Logs — Pantau, simpan, dan akses file log Anda dari AWS CloudTrail atau sumber lain. Untuk informasi selengkapnya, lihat Memantau File Log di Panduan CloudWatch Pengguna Amazon.

  Untuk informasi tentang menggunakan CloudWatch konsol untuk melihat CodeDeploy log, lihat Melihat CodeDeploy CloudWatch log di konsol Log.

- CloudWatch Acara Amazon — Cocokkan peristiwa dan arahkan ke satu atau beberapa fungsi atau aliran target untuk membuat perubahan, menangkap informasi status, dan mengambil tindakan korektif. Untuk informasi selengkapnya, lihat Apa itu CloudWatch Acara Amazon di Panduan CloudWatch Pengguna Amazon.

  Untuk informasi tentang menggunakan CloudWatch Acara dalam CodeDeploy operasi Anda, lihatMemantau penerapan dengan Amazon Events CloudWatch.

- AWS CloudTrail Pemantauan Log - Bagikan file log antar akun, pantau file CloudTrail log secara real time dengan mengirimkannya ke CloudWatch Log, menulis aplikasi pemrosesan log di Java, dan validasi bahwa file log Anda tidak berubah setelah pengiriman oleh CloudTrail. Untuk informasi selengkapnya, lihat Bekerja dengan File CloudTrail Log di Panduan AWS CloudTrail Pengguna.

Untuk informasi tentang menggunakan CloudTrail dengan CodeDeploy, lihatMonitoring Deployments.

- Amazon Simple Notification Service — Konfigurasikan pemicu yang digerakkan oleh peristiwa untuk menerima pemberitahuan SMS atau email tentang penerapan dan kejadian instance, seperti keberhasilan atau kegagalan. Untuk informasi selengkapnya, lihat Membuat topik dan Apa itu Layanan Pemberitahuan Sederhana Amazon.

  Untuk informasi tentang mengatur notifikasi Amazon SNS CodeDeploy, lihat. Monitoring Deployments with Amazon SNS Event Notifications

Alat pemantauan manual

Bagian penting lainnya dari pemantauan CodeDeploy melibatkan pemantauan secara manual item-item yang tidak tercakup oleh CloudWatch alarm. Dasbor AWS konsol CodeDeploy CloudWatch,, dan lainnya memberikan tampilan sekilas tentang keadaan lingkungan Anda. AWS Kami menyarankan Anda juga memeriksa file log pada CodeDeploy penerapan.

- CodeDeploy konsol menunjukkan:
  - Status penyebaran
  - Tanggal dan waktu setiap upaya terakhir dan penerapan revisi terakhir yang berhasil
  - Jumlah instance yang berhasil, gagal, dilewati, atau sedang berlangsung dalam penerapan
  - Status instans lokal
  - Tanggal dan waktu saat instans lokal didaftarkan atau dideregistrasi
- CloudWatch halaman rumah menunjukkan:
  - Alarm dan status saat ini
  - Grafik alarm dan sumber daya
  - Status kesehatan layanan

  Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:
  - Membuat dasbor yang disesuaikan untuk memantau layanan yang penting bagi Anda
  - Data metrik grafik untuk memecahkan masalah dan mengungkap tren
  - Cari dan telusuri semua metrik AWS sumber daya Anda
  - Membuat dan mengedit alarm untuk menerima notifikasi terkait masalah

Topik

- Monitoring Deployments with Amazon CloudWatch Tools
- Monitoring Deployments
- Monitoring Deployments with Amazon SNS Event Notifications

Memantau penerapan dengan alat Amazon CloudWatch

Anda dapat memantau CodeDeploy penerapan menggunakan CloudWatch alat berikut: Amazon CloudWatch Events, CloudWatch alarm, dan Amazon Logs. CloudWatch

Meninjau log yang dibuat oleh CodeDeploy agen dan penerapan dapat membantu Anda memecahkan masalah penyebab kegagalan penerapan. Sebagai alternatif untuk meninjau CodeDeploy log pada satu instance pada satu waktu, Anda dapat menggunakan CloudWatch Log untuk memantau semua log di lokasi pusat.

Untuk informasi tentang menggunakan CloudWatch alarm dan CloudWatch Acara untuk memantau CodeDeploy penerapan Anda, lihat topik berikut.

Topik

- Memantau penyebaran dengan CloudWatch alarm di CodeDeploy
- Memantau penerapan dengan Amazon Events CloudWatch

Memantau penyebaran dengan CloudWatch alarm di CodeDeploy

Anda dapat membuat CloudWatch alarm untuk instans atau grup Amazon EC2 Auto Scaling yang Anda gunakan CodeDeploy dalam operasi Anda. Alarm mengawasi satu metrik selama periode waktu yang Anda tentukan dan melakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. CloudWatch alarm memanggil tindakan ketika statusnya berubah (misalnya, dari `OK` ke`ALARM`).

Dengan menggunakan fungsionalitas CloudWatch alarm asli, Anda dapat menentukan tindakan apa pun yang didukung CloudWatch saat instance yang Anda gunakan dalam penerapan gagal, seperti mengirim notifikasi Amazon SNS atau menghentikan, menghentikan, me-reboot, atau memulihkan instance. Untuk CodeDeploy operasi Anda, Anda dapat mengonfigurasi grup penyebaran untuk menghentikan penerapan setiap kali CloudWatch alarm yang Anda kaitkan dengan grup penerapan diaktifkan.

Anda dapat mengaitkan hingga sepuluh CloudWatch alarm dengan grup CodeDeploy penyebaran. Jika salah satu alarm yang ditentukan diaktifkan, penerapan berhenti, dan status diperbarui ke Berhenti. Untuk menggunakan opsi ini, Anda harus memberikan CloudWatch izin untuk peran CodeDeploy layanan Anda.

Untuk informasi tentang mengatur CloudWatch alarm di CloudWatch konsol, lihat Membuat CloudWatch alarm Amazon di CloudWatch Panduan Pengguna Amazon.

Untuk informasi tentang mengaitkan CloudWatch alarm dengan grup penyebaran di CodeDeploy, lihat Buat grup penerapan dengan CodeDeploy dan. Ubah pengaturan grup penerapan dengan CodeDeploy

Topik

- Berikan CloudWatch izin untuk peran CodeDeploy layanan

Berikan CloudWatch izin untuk peran CodeDeploy layanan

Sebelum Anda dapat menggunakan pemantauan CloudWatch alarm dengan penerapan Anda, peran layanan yang Anda gunakan dalam CodeDeploy operasi Anda harus diberikan izin untuk mengakses sumber daya. CloudWatch

Untuk memberikan CloudWatch izin ke peran layanan

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di konsol IAM, di panel navigasi, pilih Peran.
3. Pilih nama peran layanan yang Anda gunakan dalam AWS CodeDeploy operasi Anda.
4. Pada tab Izin, di area Kebijakan Sebaris, pilih Buat Kebijakan Peran.

   —atau—

   Jika tombol Buat Kebijakan Peran tidak tersedia, perluas area Kebijakan Sebaris, lalu pilih klik di sini.
5. Pada halaman Setel Izin, pilih Kebijakan Kustom, lalu pilih Pilih.
6. Pada halaman Kebijakan Tinjauan, di bidang Nama Kebijakan, ketikkan nama untuk mengidentifikasi kebijakan ini, misalnya`CWAlarms`.
7. Tempelkan yang berikut ini ke bidang Dokumen Kebijakan:

JSON

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

8. Pilih Terapkan Kebijakan.

Memantau penerapan dengan Amazon Events CloudWatch

Anda dapat menggunakan CloudWatch Acara Amazon untuk mendeteksi dan bereaksi terhadap perubahan status instans atau penerapan (“peristiwa”) dalam CodeDeploy operasi Anda. Kemudian, berdasarkan aturan yang Anda buat, CloudWatch Acara akan memanggil satu atau beberapa tindakan target saat penerapan atau instance memasuki status yang Anda tentukan dalam aturan. Bergantung pada jenis perubahan status, Anda mungkin ingin mengirim pemberitahuan, menangkap informasi status, mengambil tindakan korektif, memulai peristiwa, atau mengambil tindakan lain. Anda dapat memilih jenis target berikut saat menggunakan CloudWatch Acara sebagai bagian dari CodeDeploy operasi Anda:

- AWS Lambda fungsi
- Aliran Kinesis
- Antrean Amazon SQS
- Built-in target (`EC2 CreateSnapshot API call`,`EC2 RebootInstances API call`,`EC2 StopInstances API call`, dan`EC2 TerminateInstances API call`)
- Topik Amazon SNS

Berikut ini adalah beberapa kasus penggunaan:

- Gunakan fungsi Lambda untuk meneruskan notifikasi ke saluran Slack setiap kali penerapan gagal.

- Dorong data tentang penerapan atau instans ke aliran Kinesis untuk mendukung pemantauan status real-time yang komprehensif.
- Gunakan tindakan CloudWatch alarm untuk menghentikan, menghentikan, me-reboot, atau memulihkan instans Amazon EC2 secara otomatis saat peristiwa penerapan atau instans yang Anda tentukan terjadi.

Sisa topik ini menjelaskan prosedur dasar untuk membuat aturan CloudWatch Acara untuk CodeDeploy. Namun, sebelum Anda membuat aturan acara untuk digunakan dalam CodeDeploy operasi Anda, Anda harus melakukan hal berikut:

- Lengkapi prasyarat CloudWatch Acara. Untuk selengkapnya, lihat Prasyarat CloudWatch Acara Amazon.
- Biasakan diri Anda dengan acara, aturan, dan target dalam CloudWatch Acara. Untuk informasi selengkapnya, lihat Apa itu CloudWatch Acara Amazon? dan CloudWatch Acara Baru - Lacak dan tanggapi perubahan pada AWS sumber daya Anda.
- Buat target yang akan Anda gunakan dalam aturan kejadian Anda.

Untuk membuat aturan CloudWatch Acara untuk CodeDeploy:

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa.
3. Pilih Buat aturan, lalu di bawah Pemilih acara, pilih AWS CodeDeploy.
4. Tentukan jenis detail:
   - Untuk membuat aturan yang berlaku untuk semua perubahan status dari instance dan penerapan, pilih Jenis detail apa pun, lalu lewati ke langkah 6.
   - Untuk membuat aturan yang hanya berlaku untuk instance, pilih Jenis detail spesifik, lalu pilih State-changePemberitahuan CodeDeploy Instance.
   - Untuk membuat aturan yang hanya berlaku untuk penerapan, pilih Jenis detail spesifik, lalu pilih Pemberitahuan CodeDeploy Penerapan State-change.
5. Tentukan perubahan status yang berlaku untuk aturan:
   - Untuk membuat aturan yang berlaku untuk semua perubahan status, pilih Status apa pun.

- Untuk membuat aturan yang hanya berlaku untuk beberapa perubahan status, pilih Status tertentu, lalu pilih satu atau beberapa nilai status dari daftar. Tabel berikut mencantumkan nilai status yang dapat Anda pilih:

Nilai status penerapan	Nilai status instans
GAGAL	GAGAL
MULAI	MULAI
BERHENTI	SEDIA
DIANTREKAN	BERHASIL
SEDIA	
SUKSES	

6. Tentukan CodeDeploy aplikasi mana aturan berlaku untuk:

   - Untuk membuat aturan yang berlaku untuk semua aplikasi, pilih Aplikasi apa saja, lalu lewati ke langkah 8.
   - Untuk membuat aturan yang berlaku untuk satu aplikasi saja, pilih Aplikasi spesifik, lalu pilih nama aplikasi dari daftar.

7. Tentukan grup penyebaran mana aturan berlaku untuk:

   - Untuk membuat aturan yang berlaku untuk semua grup penyebaran yang terkait dengan aplikasi yang dipilih, pilih Grup penyebaran apa pun.
   - Untuk membuat aturan yang hanya berlaku untuk salah satu grup penyebaran yang terkait dengan aplikasi yang dipilih, pilih Grup penyebaran khusus, lalu pilih nama grup penyebaran dari daftar.

8. Tinjau pengaturan aturan Anda untuk memastikannya memenuhi persyaratan pemantauan kejadian Anda.
9. Di area Targets (Target), pilih Add target* (Tambahkan target*).
10. Dalam daftar Pilih jenis target, pilih jenis target yang telah Anda siapkan untuk digunakan dengan aturan ini, lalu konfigurasikan opsi tambahan apa pun yang diperlukan oleh jenis tersebut.
11. Pilih Konfigurasikan detail.

12. Pada halaman Konfigurasi detail aturan, ketikkan nama dan deskripsi untuk aturan, lalu pilih kotak Negara untuk mengaktifkan aturan sekarang.
13. Jika Anda puas dengan aturan, pilih Create rule (Buat aturan).

Memantau penyebaran dengan AWS CloudTrail

CodeDeploy terintegrasi dengan CloudTrail, layanan yang menangkap panggilan API yang dilakukan oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. CodeDeploy CloudTrailmenangkap panggilan API dari CodeDeploy konsol, dari CodeDeploy perintah melalui AWS CLI, atau dari CodeDeploy API secara langsung. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan mana yang dibuat CodeDeploy, alamat IP sumber dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan sebagainya. Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat Panduan AWS CloudTrail Pengguna.

CodeDeploy informasi di CloudTrail

Saat CloudTrail logging diaktifkan di AWS akun Anda, panggilan API yang dilakukan untuk CodeDeploy tindakan dilacak dalam file log. CodeDeploy catatan ditulis bersama dengan catatan AWS layanan lainnya dalam file log. CloudTrail menentukan kapan harus membuat dan menulis ke file baru berdasarkan periode waktu dan ukuran file.

Semua CodeDeploy tindakan dicatat dan didokumentasikan dalam Referensi Baris AWS CodeDeploy Perintah dan Referensi AWS CodeDeploy API. Misalnya, panggilan untuk membuat penerapan, menghapus aplikasi, dan mendaftarkan revisi aplikasi menghasilkan entri dalam file log. CloudTrail

Setiap entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas pengguna di log membantu Anda menentukan apakah permintaan dibuat dengan kredensi root atau pengguna, dengan kredensi keamanan sementara untuk peran atau pengguna gabungan, atau oleh layanan lain. AWS Untuk informasi selengkapnya, lihat bidang UserIdentity di referensi CloudTrail acara.

Anda dapat menyimpan file log dalam bucket selama yang diinginkan, tetapi Anda juga dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Secara default, enkripsi sisi server Amazon S3 (SSE) digunakan untuk mengenkripsi file log Anda.

Anda dapat CloudTrail mempublikasikan notifikasi Amazon SNS saat file log baru dikirimkan. Untuk informasi lebih lanjut, lihat Mengonfigurasi pemberitahuan Amazon SNS untuk CloudTrail.

Anda juga dapat menggabungkan file CodeDeploy log dari beberapa AWS Wilayah dan beberapa AWS akun ke dalam satu bucket Amazon S3. Untuk informasi selengkapnya, lihat Menerima file CloudTrail log dari beberapa Wilayah.

Memahami entri file CodeDeploy log

CloudTrail file log dapat berisi satu atau lebih entri log di mana setiap entri terdiri dari beberapa JSON-formatted peristiwa. Entri log mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, parameter apa pun, tanggal dan waktu tindakan, dan lain-lain. Entri log tidak dijamin berada dalam urutan tertentu. Artinya, entri tersebut bukan merupakan jejak tumpukan yang disusun dari panggilan API publik.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan tindakan grup CodeDeploy create deployment:

```
{
 "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 ...
    ]
}
```

Memantau penerapan dengan notifikasi acara Amazon SNS

Anda dapat menambahkan pemicu ke grup CodeDeploy penerapan untuk menerima pemberitahuan tentang peristiwa yang terkait dengan penerapan atau instance dalam grup penerapan tersebut. Pemberitahuan ini dikirim ke penerima yang berlangganan topik Amazon SNS yang telah Anda jadikan bagian dari tindakan pemicu.

Anda dapat menerima pemberitahuan untuk CodeDeploy acara dalam pesan SMS atau pesan email. Anda juga dapat menggunakan data JSON yang dibuat ketika peristiwa tertentu terjadi dengan cara lain, seperti mengirim pesan ke antrian Amazon SQS atau menjalankan fungsi di. AWS Lambda Untuk melihat struktur data JSON yang disediakan untuk penerapan dan pemicu instance, lihat. [Format data JSON untuk CodeDeploy pemicu]

Anda dapat memilih untuk menggunakan pemicu untuk menerima pemberitahuan jika:

- Anda adalah pengembang yang perlu tahu kapan penerapan gagal atau berhenti sehingga Anda dapat memecahkan masalah itu.
- Anda adalah administrator sistem yang perlu mengetahui berapa banyak instance yang gagal untuk memantau kesehatan armada Amazon EC2 Anda.
- Anda adalah manajer yang menginginkan jumlah penyebaran dan kejadian instance sekilas, yang dapat Anda dapatkan melalui aturan pemfilteran yang merutekan berbagai jenis notifikasi ke dalam folder di klien email desktop Anda.

Anda dapat membuat hingga 10 pemicu untuk setiap grup CodeDeploy penyebaran, untuk salah satu jenis peristiwa berikut.

Acara penyebaran	Peristiwa contoh
• Berhasil • Kegagalan • Dimulai • Dihentikan • Rollback • Siap[1] • Semua acara penerapan	• Berhasil • Kegagalan • Dimulai • Siap[1] • Semua kejadian contoh

[1]Hanya berlaku untuk blue/green penerapan. Menunjukkan bahwa revisi aplikasi terbaru telah diinstal pada instance di lingkungan pengganti dan lalu lintas dari lingkungan asli sekarang dapat dialihkan di belakang penyeimbang beban. Untuk mengetahui informasi selengkapnya, lihat Bekerja dengan penerapan di CodeDeploy.

Topik

- Berikan izin Amazon SNS ke peran layanan CodeDeploy
- Buat pemicu untuk suatu CodeDeploy acara
- Mengedit pemicu dalam CodeDeploy grup penerapan
- Menghapus pemicu dari CodeDeploy grup penerapan
- Format data JSON untuk CodeDeploy pemicu

Berikan izin Amazon SNS ke peran layanan CodeDeploy

Sebelum pemicu Anda dapat menghasilkan notifikasi, peran layanan yang Anda gunakan dalam CodeDeploy operasi Anda harus diberikan izin untuk mengakses sumber daya Amazon SNS.

Untuk memberikan izin Amazon SNS ke peran layanan

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di konsol IAM, di panel navigasi, pilih Peran.
3. Pilih nama peran layanan yang Anda gunakan dalam AWS CodeDeploy operasi Anda.
4. Pada tab Izin, di area Kebijakan Sebaris, pilih Buat Kebijakan Peran.

   —atau—

   Jika tombol Buat Kebijakan Peran tidak tersedia, perluas area Kebijakan Sebaris, lalu pilih klik di sini.
5. Pada halaman Setel Izin, pilih Kebijakan Kustom, lalu pilih Pilih.
6. Pada halaman Kebijakan Tinjauan, di bidang Nama Kebijakan, masukkan nama untuk mengidentifikasi kebijakan ini, misalnya`SNSPublish`.
7. Tempelkan yang berikut ini ke bidang Dokumen Kebijakan:

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "sns:Publish",
               "Resource": "*"
           }
       ]
   }
   ```
8. Pilih Terapkan Kebijakan.

Buat pemicu untuk suatu CodeDeploy acara

Anda dapat membuat pemicu yang memublikasikan topik Amazon Simple Notification Service (Amazon SNS) untuk penyebaran atau kejadian instans AWS CodeDeploy . Kemudian, ketika peristiwa itu terjadi, semua pelanggan ke topik terkait menerima pemberitahuan melalui titik akhir yang ditentukan dalam topik, seperti pesan SMS atau pesan email. Amazon SNS menawarkan beberapa metode untuk berlangganan topik.

Sebelum membuat pemicu, Anda harus mengatur topik Amazon SNS agar pemicu dapat ditunjuk. Untuk selengkapnya, lihat Membuat topik. Saat Anda membuat topik, kami sarankan Anda memberinya nama yang mengidentifikasi tujuannya, dalam format seperti `Topic-group-us-west-3-deploy-fail` atau`Topic-group-project-2-instance-stop`.

Anda juga harus memberikan izin Amazon SNS ke peran CodeDeploy layanan sebelum pemberitahuan dapat dikirim untuk pemicu Anda. Untuk informasi, lihat Berikan izin Amazon SNS ke peran layanan CodeDeploy.

Setelah Anda membuat topik, Anda dapat menambahkan pelanggan. Untuk informasi tentang membuat, mengelola, dan berlangganan topik, lihat Apa itu Layanan Pemberitahuan Sederhana Amazon.

Buat pemicu untuk mengirim pemberitahuan untuk CodeDeploy acara (konsol)

Anda dapat menggunakan CodeDeploy konsol untuk membuat pemicu untuk suatu CodeDeploy peristiwa. Di akhir proses penyiapan, pesan pemberitahuan pengujian dikirim untuk memastikan bahwa izin dan detail pemicu diatur dengan benar.

Untuk membuat pemicu untuk suatu CodeDeploy peristiwa

1. Di Konsol Manajemen AWS, buka AWS CodeDeploy konsol.
2. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

3. Di panel navigasi, perluas Deploy, lalu pilih Applications.

4. Pada halaman Aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran tempat Anda ingin menambahkan pemicu.
5. Pada halaman Detail aplikasi, pilih grup penyebaran tempat Anda ingin menambahkan pemicu.
6. Pilih Edit.
7. Perluas Lanjutan - opsional.
8. Di area Pemicu, pilih Buat pemicu.
9. Di panel Pemicu Create deployment, lakukan hal berikut:

    a. Di nama Pemicu, masukkan nama untuk pemicu yang membuatnya mudah untuk mengidentifikasi tujuannya. Kami merekomendasikan format seperti `Trigger-group-us-west-3-deploy-fail` atau`Trigger-group-eu-central-instance-stop`.

    b. Di Acara, pilih jenis atau jenis acara untuk memicu topik Amazon SNS untuk mengirim notifikasi.

    c. Di topik Amazon SNS, pilih nama topik yang Anda buat untuk mengirim notifikasi untuk pemicu ini.

    d. Pilih Buat pemicu. CodeDeploy mengirimkan pemberitahuan pengujian untuk mengonfirmasi bahwa Anda telah mengonfigurasi akses dengan benar antara CodeDeploy dan topik Amazon SNS. Bergantung pada jenis titik akhir yang Anda pilih untuk topik tersebut, dan jika Anda berlangganan topik, Anda menerima konfirmasi dalam pesan SMS atau pesan email.
10. Pilih Simpan perubahan.

Buat pemicu untuk mengirim notifikasi untuk CodeDeploy acara (CLI)

Anda dapat menggunakan CLI untuk menyertakan pemicu saat Anda membuat grup penerapan, atau Anda dapat menambahkan pemicu ke grup penerapan yang ada.

Untuk membuat pemicu untuk mengirim pemberitahuan untuk grup penyebaran baru

Buat file JSON untuk mengonfigurasi grup penyebaran, lalu jalankan perintah create-deployment-group menggunakan opsi. `--cli-input-json`

Cara termudah untuk membuat file JSON adalah dengan menggunakan `--generate-cli-skeleton` opsi untuk mendapatkan salinan format JSON, dan kemudian memberikan nilai yang diperlukan dalam editor teks biasa.

1. Jalankan perintah berikut, lalu salin hasilnya ke editor teks biasa.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Tambahkan nama CodeDeploy aplikasi yang ada ke output:

```
{
    "applicationName": "TestApp-us-east-2",
    "deploymentGroupName": "",
    "deploymentConfigName": "",
    "ec2TagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "onPremisesInstanceTagFilters": [
        {
            "Key": "",
            "Value": "",
            "Type": ""
        }
    ],
    "autoScalingGroups": [
        ""
    ],
    "serviceRoleArn": "",
    "triggerConfigurations": [
        {
            "triggerName": "",
            "triggerTargetArn": "",
            "triggerEvents": [
                ""
            ]
        }
    ]
}
```

3. Berikan nilai untuk parameter yang ingin Anda konfigurasikan.

   Saat Anda menggunakan perintah create-deployment-group, Anda harus memberikan, minimal, nilai untuk parameter berikut:

- `applicationName`: Nama aplikasi yang sudah dibuat di akun Anda.
- `deploymentGroupName`: Nama untuk grup penyebaran yang Anda buat.
- `serviceRoleArn`: ARN dari peran layanan yang ada disiapkan CodeDeploy di akun Anda. Untuk informasi, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Di `triggerConfigurations` bagian ini, berikan nilai untuk parameter berikut:

- `triggerName`: Nama yang ingin Anda berikan pelatuknya sehingga Anda dapat dengan mudah mengidentifikasinya. Kami merekomendasikan format seperti `Trigger-group-us-west-3-deploy-fail` atau`Trigger-group-eu-central-instance-stop`.
- `triggerTargetArn`: ARN dari topik Amazon SNS yang Anda buat untuk dikaitkan dengan pemicu Anda, dalam format ini:. `arn:aws:sns:us-east-2:444455556666:NewTestTopic`
- `triggerEvents`: Jenis acara atau acara yang ingin Anda picu notifikasi. Anda dapat menentukan satu atau beberapa jenis acara, memisahkan beberapa nama jenis acara dengan koma (misalnya,`"triggerEvents": ["DeploymentSuccess","DeploymentFailure","InstanceFailure"]`). Saat Anda menambahkan lebih dari satu jenis acara, notifikasi untuk semua jenis tersebut dikirim ke topik yang Anda tentukan, bukan ke topik yang berbeda untuk masing-masing. Anda dapat memilih dari jenis acara berikut:
  - DeploymentStart
  - DeploymentSuccess
  - DeploymentFailure
  - DeploymentStop
  - DeploymentRollback
  - DeploymentReady (Hanya berlaku untuk instance pengganti dalam blue/green penerapan)
  - InstanceStart
  - InstanceSuccess
  - InstanceFailure
  - InstanceReady (Hanya berlaku untuk instance pengganti dalam blue/green penerapan)

Contoh konfigurasi berikut membuat grup penyebaran bernama `dep-group-ghi-789-2` untuk aplikasi bernama `TestApp-us-east-2` dan pemicu yang meminta pengiriman pemberitahuan setiap kali penerapan dimulai, berhasil, atau gagal:

```
{
    "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. Simpan pembaruan Anda sebagai file JSON, lalu panggil file itu menggunakan `--cli-input-json` opsi saat Anda menjalankan create-deployment-group perintah:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

Di akhir proses pembuatan, Anda menerima pesan pemberitahuan pengujian yang menunjukkan izin dan detail pemicu diatur dengan benar.

Untuk membuat pemicu untuk mengirim pemberitahuan untuk grup penyebaran yang ada

Untuk menggunakan AWS CLI untuk menambahkan pemicu CodeDeploy peristiwa ke grup penyebaran yang ada, buat file JSON untuk memperbarui grup penerapan, lalu jalankan perintah update-deployment-group menggunakan opsi. `--cli-input-json`

Cara termudah untuk membuat file JSON adalah dengan menjalankan get-deployment-group perintah untuk mendapatkan salinan konfigurasi grup penyebaran, dalam format JSON, dan kemudian memperbarui nilai parameter dalam editor teks biasa.

1. Jalankan perintah berikut, lalu salin hasilnya ke editor teks biasa.

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```

2. Hapus yang berikut dari output:

   - Di awal output, hapus`{ "deploymentGroupInfo":`.
   - Di akhir output, hapus`}`.
   - Hapus baris yang berisi`deploymentGroupId`.
   - Hapus baris yang berisi`deploymentGroupName`.

   Isi file teks Anda sekarang akan terlihat mirip dengan yang berikut ini:

```
{
    "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. Di `triggerConfigurations` bagian ini, tambahkan data untuk`triggerEvents`,`triggerTargetArn`, dan `triggerName` parameter. Untuk informasi tentang parameter konfigurasi pemicu, lihat TriggerConfig.

   Isi file teks Anda sekarang harus terlihat mirip dengan yang berikut ini. Kode ini meminta pemberitahuan untuk dikirim setiap kali penerapan dimulai, berhasil, atau gagal.

```
{
    "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. Simpan pembaruan Anda sebagai file JSON, lalu jalankan perintah update-deployment-group menggunakan opsi. `--cli-input-json` Pastikan untuk menyertakan `--current-deployment-group-name` opsi dan mengganti nama file JSON Anda untuk*`filename`*:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
```

Di akhir proses pembuatan, Anda menerima pesan pemberitahuan pengujian yang menunjukkan izin dan detail pemicu diatur dengan benar.

Mengedit pemicu dalam CodeDeploy grup penerapan

Jika persyaratan notifikasi berubah, Anda dapat memodifikasi pemicu daripada membuat yang baru.

Memodifikasi CodeDeploy pemicu (CLI)

Untuk menggunakan AWS CLI untuk mengubah detail pemicu untuk CodeDeploy peristiwa saat Anda memperbarui grup penerapan, buat file JSON untuk menentukan perubahan pada properti grup penyebaran, lalu jalankan perintah update-deployment-group dengan opsi. `--cli-input-json`

Cara termudah untuk membuat file JSON adalah dengan menjalankan get-deployment-group perintah untuk mendapatkan detail grup penyebaran saat ini dalam format JSON, dan kemudian mengedit nilai yang diperlukan dalam editor teks biasa.

1. Jalankan perintah berikut, ganti nama aplikasi dan grup penyebaran Anda dengan dan: *application deployment-group*

   ```
   aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
   ```

2. Salin hasil perintah ke editor teks biasa dan kemudian hapus yang berikut ini:

   - Di awal output, hapus`{ "deploymentGroupInfo":`.
   - Di akhir output, hapus`}`.
   - Hapus baris yang berisi`deploymentGroupId`.
   - Hapus baris yang berisi`deploymentGroupName`.

Isi file teks Anda sekarang akan terlihat mirip dengan yang berikut ini:

```
{
    "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. Ubah parameter apa pun, seperlunya. Untuk informasi tentang parameter konfigurasi pemicu, lihat TriggerConfig.
4. Simpan pembaruan Anda sebagai file JSON, lalu jalankan perintah update-deployment-group menggunakan opsi. `--cli-input-json` Pastikan untuk menyertakan `--current-deployment-group-name` opsi dan mengganti nama file JSON Anda untuk*`filename`*:

> **Important**
>
> Pastikan untuk menyertakan `file://` sebelum nama file. Diperlukan dalam perintah ini.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-
group-name --cli-input-json file://filename.json
```

Di akhir proses pembuatan, Anda menerima pesan pemberitahuan pengujian yang menunjukkan izin dan detail pemicu diatur dengan benar.

Menghapus pemicu dari CodeDeploy grup penerapan

Karena ada batas 10 pemicu per grup penerapan, Anda mungkin ingin menghapus pemicu jika tidak lagi digunakan. Anda tidak dapat membatalkan penghapusan pemicu, tetapi Anda dapat membuatnya kembali.

Menghapus pemicu dari grup penerapan (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy.

   > **Note**
   >
   > Masuk dengan pengguna yang sama dengan yang Anda aturMemulai dengan CodeDeploy.

2. Di panel navigasi, perluas Deploy, lalu pilih Applications.
3. Pada halaman Aplikasi, pilih nama aplikasi yang terkait dengan grup penyebaran tempat Anda ingin menghapus pemicu.
4. Pada halaman Detail aplikasi, pilih grup penyebaran tempat Anda ingin menghapus pemicu.
5. Pilih Edit.
6. Perluas Lanjutan - opsional.
7. Di area Pemicu, pilih pemicu yang ingin Anda hapus, lalu pilih Hapus pemicu.
8. Pilih Simpan perubahan.

Hapus pemicu dari grup penerapan (CLI)

Untuk menggunakan CLI untuk menghapus pemicu, panggil perintah update-deployment-group, dengan parameter konfigurasi pemicu kosong, dengan menentukan:

- Nama aplikasi yang terkait dengan grup penyebaran. Untuk melihat daftar nama aplikasi, panggil perintah daftar-aplikasi.
- Nama grup penyebaran yang terkait dengan aplikasi. Untuk melihat daftar nama grup penyebaran, panggil perintah list-deployment-groups.

Contoh:

```
aws deploy update-deployment-group --application-name application-name --current-
deployment-group-name deployment-group-name --trigger-configurations
```

Format data JSON untuk CodeDeploy pemicu

Anda dapat menggunakan keluaran JSON yang dibuat saat pemicu penerapan atau instans diaktifkan dalam alur kerja notifikasi kustom, seperti mengirim pesan ke antrian Amazon SQS atau menjalankan fungsi. AWS Lambda

> **Note**
>
> Panduan ini tidak membahas cara mengonfigurasi notifikasi menggunakan JSON. Untuk informasi tentang menggunakan Amazon SNS untuk mengirim pesan ke antrian Amazon SQS, lihat Mengirim pesan Amazon SNS ke antrian Amazon SQS. Untuk informasi tentang menggunakan Amazon SNS untuk menjalankan fungsi Lambda, lihat Memanggil fungsi Lambda menggunakan notifikasi Amazon SNS.

Contoh berikut menunjukkan struktur output JSON yang tersedia dengan CodeDeploy pemicu.

Contoh Output JSON untuk Pemicu 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
    ]
}
```

Contoh Output JSON untuk Pemicu 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"
    }
}
```

Keamanan di AWS CodeDeploy

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Third-party Auditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari program AWS kepatuhan. Untuk mempelajari tentang program kepatuhan yang berlaku AWS CodeDeploy, lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan CodeDeploy. Topik berikut menunjukkan cara mengonfigurasi CodeDeploy untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan CodeDeploy sumber daya Anda.

Topik

- Perlindungan data di AWS CodeDeploy
- Identitas dan manajemen akses untuk AWS CodeDeploy
- Penebangan dan pemantauan di CodeDeploy
- Validasi kepatuhan untuk AWS CodeDeploy
- Ketahanan di AWS CodeDeploy
- Keamanan infrastruktur di AWS CodeDeploy

Perlindungan data di AWS CodeDeploy

Model tanggung jawab bersama AWS berlaku untuk perlindungan data di AWS CodeDeploy. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat FAQ Privasi Data AWS . Untuk informasi tentang perlindungan data di Eropa, lihat Pusat Peraturan Umum Perlindungan Data (GDPR).

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan CodeDeploy atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan

supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Privasi lalu lintas antarjaringan

CodeDeploy adalah layanan penyebaran terkelola penuh yang mendukung instans EC2, fungsi Lambda, Amazon ECS, dan server lokal. Untuk instans EC2 dan server lokal, agen berbasis host berkomunikasi dengan menggunakan TLS. CodeDeploy

Saat ini, komunikasi dari agen ke layanan memerlukan koneksi internet keluar sehingga agen dapat berkomunikasi dengan publik CodeDeploy dan titik akhir layanan Amazon S3. Dalam cloud pribadi virtual, ini dapat dicapai dengan gateway internet, koneksi VPN situs ke situs ke jaringan perusahaan Anda, atau koneksi langsung.

CodeDeploy Agen mendukung proxy HTTP.

Titik akhir VPC Amazon, yang didukung oleh AWS PrivateLink, tersedia untuk CodeDeploy di Wilayah tertentu. Lihat perinciannya di Gunakan CodeDeploy dengan Amazon Virtual Private Cloud.

> **Note**
>
> CodeDeploy Agen hanya diperlukan jika Anda menyebarkan ke platform EC2/On-premises komputasi Amazon. Agen tidak diperlukan untuk penerapan yang menggunakan Amazon ECS atau AWS Lambda platform komputasi.

Enkripsi Saat Diam

Kode pelanggan tidak disimpan di CodeDeploy. Sebagai layanan penerapan, CodeDeploy mengirimkan perintah ke CodeDeploy agen yang berjalan pada instans EC2 atau server lokal. CodeDeploy Agen kemudian mengeksekusi perintah menggunakan TLS. Data model layanan untuk penerapan, konfigurasi penerapan, grup penyebaran, aplikasi, dan revisi aplikasi disimpan di Amazon DynamoDB dan dienkripsi saat istirahat menggunakan, dimiliki, dan dikelola oleh. Kunci milik AWS CodeDeploy Untuk informasi lebih lanjut, lihat Kunci milik AWS s.

Enkripsi Saat Bergerak

CodeDeploy Agen memulai semua komunikasi dengan CodeDeploy lebih dari port 443. Agen melakukan polling CodeDeploy dan mendengarkan perintah. CodeDeploy Agen adalah open

source. Semua komunikasi layanan-ke-layanan dan klien-ke-layanan dienkripsi dalam perjalanan menggunakan TLS. Ini melindungi data pelanggan dalam perjalanan antara CodeDeploy dan layanan lain seperti Amazon S3.

Manajemen kunci enkripsi

Tidak ada kunci enkripsi yang perlu Anda kelola. Data model CodeDeploy layanan dienkripsi menggunakan Kunci milik AWS, dimiliki dan dikelola oleh. CodeDeploy Untuk informasi lebih lanjut, lihat Kunci milik AWS s.

Identitas dan manajemen akses untuk AWS CodeDeploy

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. CodeDeploy IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan Identitas
- Mengelola Akses menggunakan Kebijakan
- Bagaimana AWS CodeDeploy bekerja dengan IAM
- AWS kebijakan terkelola (standar) untuk CodeDeploy
- CodeDeploy update ke AWS kebijakan terkelola
- AWS CodeDeploy contoh kebijakan berbasis identitas
- Pemecahan Masalah AWS CodeDeploy identitas dan akses
- CodeDeploy referensi izin
- Cross-service pencegahan wakil bingung

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Pemecahan Masalah AWS CodeDeploy identitas dan akses)

- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Bagaimana AWS CodeDeploy bekerja dengan IAM)
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat AWS CodeDeploy contoh kebijakan berbasis identitas)

Mengautentikasi dengan Identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensil dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensional. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Pengguna dan grup

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola Akses menggunakan Kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Identity-based kebijakan

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Identity-based kebijakan dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada beberapa identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Jenis-jenis Kebijakan Lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – Menentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCP) – Menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCP) dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai Jenis Kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS CodeDeploy bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses CodeDeploy, Anda harus memahami fitur IAM mana yang tersedia untuk digunakan. CodeDeploy Untuk informasi selengkapnya, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- CodeDeploy kebijakan berbasis identitas
- CodeDeploy kebijakan berbasis sumber daya
- Otorisasi berdasarkan tag CodeDeploy
- CodeDeploy Peran IAM

CodeDeploy kebijakan berbasis identitas

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. CodeDeploy mendukung tindakan, sumber daya, dan kunci kondisi. Untuk informasi tentang elemen yang Anda gunakan dalam kebijakan JSON, lihat referensi elemen kebijakan IAM JSON di Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan CodeDeploy menggunakan `codedeploy:` awalan sebelum tindakan. Misalnya, `codedeploy:GetApplication` izin memberikan izin pengguna untuk melakukan operasi. `GetApplication` Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. CodeDeploy mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
      "codedeploy:action1",
      "codedeploy:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, sertakan tindakan berikut untuk menentukan semua tindakan yang dimulai dengan kata`Describe`:

```
"Action": "ec2:Describe*"
```

Untuk daftar tindakan, lihat CodeDeploy Tindakan yang Ditentukan oleh AWS CodeDeploy dalam Panduan Pengguna IAM.

Untuk tabel yang mencantumkan semua tindakan CodeDeploy API dan sumber daya yang diterapkan, lihatCodeDeploy referensi izin.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Misalnya, Anda dapat menunjukkan grup penyebaran (*myDeploymentGroup*) dalam pernyataan Anda menggunakan ARN sebagai berikut:

```
"Resource": "arn:aws:codedeploy:us-
west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

Anda juga dapat menentukan semua grup penyebaran milik akun dengan menggunakan karakter wildcard (*) sebagai berikut:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Untuk menentukan semua sumber daya, atau jika tindakan API tidak mendukung ARN, gunakan karakter wildcard (*) dalam `Resource` elemen sebagai berikut:

```
"Resource": "*"
```

Beberapa tindakan CodeDeploy API menerima beberapa sumber daya (misalnya,`BatchGetDeploymentGroups`). Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN mereka dengan koma, sebagai berikut:

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy menyediakan satu set operasi untuk bekerja dengan CodeDeploy sumber daya. Untuk daftar operasi yang tersedia, lihat CodeDeploy referensi izin.

Untuk daftar jenis CodeDeploy sumber daya dan ARNnya, lihat Sumber Daya yang Ditentukan oleh AWS CodeDeploy dalam Panduan Pengguna IAM. Untuk informasi tentang tindakan di mana Anda dapat menentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh. AWS CodeDeploy

CodeDeploy sumber daya dan operasi

Di CodeDeploy, sumber daya utama adalah grup penyebaran. Dalam kebijakan, Anda menggunakan Nama Sumber Daya Amazon (ARN) untuk mengidentifikasi sumber daya yang berlaku untuk kebijakan tersebut. CodeDeploy mendukung sumber daya lain yang dapat digunakan dengan grup penyebaran, termasuk aplikasi, konfigurasi penerapan, dan instance. Ini dirujuk sebagai sub-sumber daya. Sumber daya dan subsumber daya ini memiliki ARN unik yang terkait dengannya. Untuk informasi selengkapnya, lihat nama sumber daya Amazon (ARN) di. Referensi Umum Amazon Web Services

Jenis Sumber Daya	Format ARN
Grup deployment	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentgroup:` *`application-name`* `/`*`deployment-group-name`*
Aplikasi	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:application:` *`application-name`*
Konfigurasi deployment	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:deploymentconfig:` *`deployment-configuration-name`*
Instans	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:instance/` *`instance-ID`*
Semua CodeDeploy sumber daya	`arn:aws:codedeploy:*`
Semua CodeDeploy sumber daya yang dimiliki oleh akun yang ditentukan di Wilayah yang ditentukan	`arn:aws:codedeploy:` *`region`*`:`*`account-id`* `:*`

> **Note**
>
> Sebagian besar layanan dalam AWS memperlakukan titik dua (:) atau garis miring (/) sebagai karakter yang sama di ARN. Namun, CodeDeploy menggunakan kecocokan yang tepat dalam pola dan aturan sumber daya. Pastikan untuk menggunakan karakter ARN yang benar saat Anda membuat pola acara sehingga cocok dengan sintaks ARN di sumber daya.

Kunci syarat

CodeDeploy tidak menyediakan kunci kondisi khusus layanan apa pun, tetapi mendukung penggunaan beberapa kunci kondisi global. Untuk informasi selengkapnya, lihat Kunci konteks kondisi global AWS dalam Panduan Pengguna IAM.

Contoh

Untuk melihat contoh kebijakan CodeDeploy berbasis identitas, lihat. AWS CodeDeploy contoh kebijakan berbasis identitas

CodeDeploy kebijakan berbasis sumber daya

CodeDeploy tidak mendukung kebijakan berbasis sumber daya. Untuk melihat contoh halaman kebijakan berbasis sumber daya terperinci, lihat Menggunakan kebijakan berbasis sumber daya untuk. AWS Lambda

Otorisasi berdasarkan tag CodeDeploy

CodeDeploy tidak mendukung penandaan sumber daya atau mengendalikan akses berdasarkan tag.

CodeDeploy Peran IAM

Peran IAM adalah entitas di AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensyal sementara dengan CodeDeploy

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensil keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

CodeDeploy mendukung penggunaan kredensyal sementara.

Service-linked peran

CodeDeploy tidak mendukung peran terkait layanan.

Peran Layanan

Fitur ini memungkinkan layanan untuk menerima peran layanan atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di AWS akun Anda dan dimiliki oleh akun. Ini berarti bahwa pengguna dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

CodeDeploy mendukung peran layanan.

Memilih peran IAM di CodeDeploy

Saat membuat sumber daya grup penyebaran CodeDeploy, Anda harus memilih peran yang CodeDeploy memungkinkan mengakses Amazon EC2 atas nama Anda. Jika sebelumnya Anda telah membuat peran layanan atau peran terkait layanan, CodeDeploy memberi Anda daftar peran yang dapat dipilih. Penting untuk memilih peran yang memungkinkan akses untuk memulai dan menghentikan instans EC2.

AWS kebijakan terkelola (standar) untuk CodeDeploy

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan yang AWS dikelola ini memberikan izin untuk kasus penggunaan umum sehingga Anda dapat menghindari keharusan menyelidiki izin mana yang diperlukan. Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

Topik

- Daftar AWS kebijakan terkelola untuk CodeDeploy
- CodeDeploy kebijakan dan pemberitahuan terkelola

Daftar AWS kebijakan terkelola untuk CodeDeploy

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk CodeDeploy:

- `AWSCodeDeployFullAccess`: Memberikan akses penuh ke CodeDeploy.

> **Note**
>
> AWSCodeDeployFullAccess tidak memberikan izin untuk operasi di layanan lain yang diperlukan untuk menyebarkan aplikasi Anda, seperti Amazon EC2 dan Amazon S3, hanya untuk operasi khusus untuk. CodeDeploy

- `AWSCodeDeployDeployerAccess`: Memberikan izin untuk mendaftar dan menyebarkan revisi.

- `AWSCodeDeployReadOnlyAccess`: Memberikan akses hanya-baca ke. CodeDeploy

- `AWSCodeDeployRole`: Memungkinkan CodeDeploy untuk:
  - baca tag pada instans Anda atau identifikasi instans Amazon EC2 Anda dengan nama grup Amazon EC2 Auto Scaling
  - membaca, membuat, memperbarui, dan menghapus grup Amazon EC2 Auto Scaling, kait siklus hidup, kebijakan penskalaan, dan fitur kolam hangat
  - mempublikasikan informasi ke topik Amazon SNS
  - mengambil informasi tentang alarm Amazon CloudWatch
  - membaca dan memperbarui sumber daya di layanan Elastic Load Balancing

  Kebijakan tersebut berisi kode berikut:

  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`: Memberikan CodeDeploy izin untuk mengakses AWS Lambda dan sumber daya lain yang diperlukan untuk penerapan.

- `AWSCodeDeployRoleForECS`: Memberikan CodeDeploy izin untuk mengakses Amazon ECS dan sumber daya lain yang diperlukan untuk penerapan.

- `AWSCodeDeployRoleForECSLimited`: Memberikan CodeDeploy izin untuk mengakses Amazon ECS dan sumber daya lain yang diperlukan untuk penerapan dengan pengecualian berikut:
  - Di `hooks` bagian AppSpec file, hanya fungsi Lambda dengan nama yang dimulai dengan yang `CodeDeployHook_` dapat digunakan. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan Amazon ECS.
  - Akses bucket S3 terbatas pada bucket S3 dengan tag registrasi,`UseWithCodeDeploy`, yang memiliki nilai. `true` Untuk informasi selengkapnya, lihat Penandaan objek.
- `AmazonEC2RoleforAWSCodeDeployLimited`: Memberikan CodeDeploy izin untuk mendapatkan dan mencantumkan objek di bucket CodeDeploy Amazon S3. Kebijakan tersebut berisi kode berikut:

  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"
                }
            }
        }
    ]
}
```

Izin untuk beberapa aspek proses penerapan diberikan kepada dua tipe peran lain yang bertindak atas nama: CodeDeploy

- Profil instans IAM adalah peran IAM yang Anda lampirkan ke instans Amazon EC2 Anda. Profil ini mencakup izin yang diperlukan untuk mengakses bucket Amazon S3 GitHub atau repositori tempat aplikasi disimpan. Untuk informasi selengkapnya, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.
- Peran layanan adalah peran IAM yang memberikan izin ke AWS layanan sehingga dapat mengakses sumber daya. AWS Kebijakan yang Anda lampirkan ke peran layanan menentukan AWS sumber daya yang dapat diakses oleh layanan dan tindakan yang dapat dilakukan dengan sumber daya tersebut. Untuk CodeDeploy, peran layanan digunakan untuk hal berikut:
  - Untuk membaca tag yang diterapkan pada instans atau nama grup Amazon EC2 Auto Scaling yang terkait dengan instans. Ini memungkinkan CodeDeploy untuk mengidentifikasi instance yang dapat digunakan aplikasi.
  - Untuk melakukan operasi pada instans, Amazon EC2 Auto Scaling mengelompokkan, dan penyeimbang beban Elastic Load Balancing.
  - Untuk mempublikasikan informasi ke topik Amazon SNS sehingga notifikasi dapat dikirim saat penerapan atau kejadian instans tertentu terjadi.
  - Untuk mengambil informasi tentang CloudWatch alarm untuk mengatur pemantauan alarm untuk penyebaran.

  Untuk informasi selengkapnya, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.

Anda juga dapat membuat kebijakan IAM khusus untuk memberikan izin untuk CodeDeploy tindakan dan sumber daya. Anda melampirkan kebijakan khusus ini ke peran IAM, lalu Anda menetapkan peran tersebut ke pengguna atau grup yang memerlukan izin.

CodeDeploy kebijakan dan pemberitahuan terkelola

CodeDeploy mendukung pemberitahuan untuk membuat pengguna sadar akan perubahan penting pada penerapan. Kebijakan terkelola untuk CodeDeploy menyertakan pernyataan kebijakan untuk fungsionalitas notifikasi. Untuk informasi selengkapnya, lihat Apa itu notifikasi?.

Izin untuk pemberitahuan dalam kebijakan terkelola akses penuh

Kebijakan terkelola `AWSCodeDeployFullAccess` mencakup pernyataan berikut untuk mengizinkan akses penuh ke notifikasi. Pengguna yang menerapkan kebijakan terkelola ini juga dapat membuat dan mengelola topik Amazon SNS untuk notifikasi, berlangganan dan berhenti berlangganan pengguna ke topik, serta mencantumkan topik yang akan dipilih sebagai target aturan notifikasi.

```
    {
        "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": "*"
    }
```

Izin untuk notifikasi dalam kebijakan terkelola hanya-baca

Kebijakan terkelola `AWSCodeDeployReadOnlyAccess` mencakup pernyataan berikut untuk mengizinkan akses penuh ke notifikasi. Pengguna yang menerapkan kebijakan terkelola ini dapat melihat pemberitahuan untuk sumber daya, tetapi tidak dapat membuat, mengelola, atau berlangganan ke sumber daya tersebut.

```
   {
        "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": "*"
    }
```

Izin untuk pemberitahuan dalam kebijakan terkelola lainnya

Kebijakan `AWSCodeDeployDeployerAccess` terkelola mencakup pernyataan berikut untuk memungkinkan pengguna membuat, memperbarui, berlangganan, dan melihat pemberitahuan untuk sumber daya, tetapi tidak dapat menghapusnya. Pengguna yang menerapkan kebijakan terkelola ini juga dapat membuat dan mengelola topik Amazon SNS untuk notifikasi.

Kebijakan ini mencakup izin untuk melakukan hal berikut:

- `codestar-notifications:CreateNotificationRule`— Memungkinkan kepala sekolah untuk membuat notifikasi.
- `codestar-notifications:DescribeNotificationRule`— Memungkinkan kepala sekolah untuk mengambil informasi tentang pemberitahuan.
- `codestar-notifications:UpdateNotificationRule`— Memungkinkan kepala sekolah untuk memperbarui pemberitahuan.
- `codestar-notifications:Subscribe`— Memungkinkan kepala sekolah untuk berlangganan pembaruan pemberitahuan.
- `codestar-notifications:Unsubscribe`— Memungkinkan kepala sekolah untuk berhenti berlangganan pembaruan pemberitahuan.
- `codestar-notifications:ListNotificationRules`— Memungkinkan kepala sekolah untuk mengambil daftar aturan pemberitahuan.
- `codestar-notifications:ListTargets`— Memungkinkan kepala sekolah untuk mengambil daftar target.
- `codestar-notifications:ListTagsforResource`— Memungkinkan prinsipal untuk mengambil daftar tag.
- `codestar-notifications:ListEventTypes`— Memungkinkan kepala sekolah untuk mengambil daftar jenis acara.
- `chatbot:DescribeSlackChannelConfiguration`— Memungkinkan kepala sekolah untuk mengambil informasi tentang konfigurasi saluran Slack.

- `sns:ListTopics`— Memungkinkan kepala sekolah untuk mengambil daftar topik Amazon SNS untuk pemberitahuan.

```
  {
     "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" : "*"
    }
```

Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk CodeStar Pemberitahuan AWS.

CodeDeploy update ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola CodeDeploy sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan ke umpan RSS pada halaman CodeDeploy Riwayat dokumen.

Ubah	Deskripsi	Date
`AWSCodeDeployDeployerAccess` kebijakan terkelola - Pembaruan kebijakan yang ada	Memperbarui `codestar-notifications:NotificationsForResource` tindakan untuk mendukung perubahan validasi kebijakan IAM. Sumber daya asli `arn:aws:codedeploy:*` telah diperbarui kearn:aws:codedeploy:*:*:application:* . Untuk informasi lebih lanjut tentang kebijakan ini, lihatIzin untuk pemberitahuan dalam kebijakan terkelola lainnya.	Desember 16, 2024
`AWSCodeDeployReadOnlyAccess` kebijakan terkelola - Pembaruan kebijakan yang ada	Memperbarui `codestar-notifications:NotificationsForResource` tindakan untuk	Desember 16, 2024

Ubah	Deskripsi	Date
	mendukung perubahan validasi kebijakan IAM. Sumber daya asli `arn:aws:codedeploy:*` telah diperbarui ke `arn:aws:codedeploy:*:*:application:*` . Untuk informasi lebih lanjut tentang kebijakan ini, lihat Izin untuk notifikasi dalam kebijakan terkelola hanya-baca.	
`AWSCodeDeployFullAccess` kebijakan terkelola - Pembaruan kebijakan yang ada	Memperbarui `codestar-notifications:NotificationsForResource` tindakan untuk mendukung perubahan validasi kebijakan IAM. Sumber daya asli `arn:aws:codedeploy:*` telah diperbarui ke `arn:aws:codedeploy:*:*:application:*` . Untuk informasi lebih lanjut tentang kebijakan ini, lihat Izin untuk pemberitahuan dalam kebijakan terkelola akses penuh.	Desember 16, 2024

Ubah	Deskripsi	Date
`AWSCodeDeployRole` kebijakan terkelola - Pembaruan kebijakan yang ada	Menambahkan `elasticloadbalancing:DescribeLoadBalancerAttributes` dan `elasticloadbalancing:DescribeTargetGroupAttributes` tindakan ke pernyataan kebijakan untuk mendukung perubahan Elastic Load Balancing. Untuk informasi lebih lanjut tentang kebijakan ini, lihatAWSCodeDeployRole.	16 Agustus 2023
`AWSCodeDeployFullAccess` kebijakan terkelola - Pembaruan kebijakan yang ada	Menambahkan `chatbot:ListMicrosoftTeamsChannelConfigurations` tindakan ke pernyataan kebijakan untuk mendukung perubahan pemberitahuan. Untuk informasi lebih lanjut tentang kebijakan ini, lihatAWSCodeDeployRole.	11 Mei 2023
`AWSCodeDeployRole` kebijakan terkelola - Pembaruan kebijakan yang ada	Menambahkan `autoscaling:CreateOrUpdateTags` tindakan ke pernyataan kebijakan untuk mendukung perubahan otorisasi Amazon EC2 Auto Scaling. Untuk informasi lebih lanjut tentang kebijakan ini, lihatAWSCodeDeployRole.	3 Februari 2023

Ubah	Deskripsi	Date
`AmazonEC2RoleforAWSCodeDeployLimited` kebijakan terkelola - Pembaruan kebijakan yang ada	Menghapus `s3:ListBucket` tindakan dari pernyataan kebijakan yang mencakup `s3:ExistingObjectTag/UseWithCodeDeploy` kondisi. Untuk informasi lebih lanjut tentang kebijakan ini, lihatAmazonEC2RoleforAWSCodeDeployLimited.	22 November 2021
`AWSCodeDeployRole` kebijakan terkelola - Pembaruan kebijakan yang ada	Menambahkan `autoscaling:PutWarmPool` tindakan untuk mendukung penambahan kolam hangat ke grup Amazon EC2 Auto Scaling blue/green untuk penerapan. Menghapus tindakan duplikat yang tidak perlu.	18 Mei 2021
CodeDeploy mulai melacak perubahan	CodeDeploy mulai melacak perubahan untuk kebijakan yang AWS dikelola.	18 Mei 2021

AWS CodeDeploy contoh kebijakan berbasis identitas

Secara default, pengguna tidak memiliki izin untuk membuat atau memodifikasi CodeDeploy sumber daya. Mereka juga tidak dapat melakukan tugas menggunakan Konsol Manajemen AWS, AWS CLI, atau AWS API. Anda harus membuat kebijakan IAM yang memberikan izin peran IAM untuk melakukan operasi API pada sumber daya tertentu yang mereka butuhkan. Anda kemudian harus melampirkan peran IAM tersebut ke pengguna atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan pada tab JSON dalam Panduan Pengguna IAM.

Dalam CodeDeploy, kebijakan berbasis identitas digunakan untuk mengelola izin ke berbagai sumber daya yang terkait dengan proses penyebaran. Anda dapat mengontrol akses ke jenis sumber daya berikut:

- Aplikasi dan revisi aplikasi.
- Penerapan.
- Konfigurasi penerapan.
- Instans dan instans lokal.

Kemampuan yang dikendalikan oleh kebijakan sumber daya bervariasi tergantung pada jenis sumber daya, seperti yang diuraikan dalam tabel berikut:

Jenis sumber daya	Kemampuan
Semua	Lihat dan daftar detail tentang sumber daya
Aplikasi Konfigurasi deployment Grup penyebaran	Buat sumber daya Hapus sumber daya
Deployment	Buat deployment Hentikan penerapan
Revisi aplikasi	Daftarkan revisi aplikasi
Aplikasi Grup penyebaran	Perbarui sumber daya
On-premises contoh	Tambahkan tag ke instance Hapus tag dari instance Daftar contoh Contoh deregister

Contoh berikut menunjukkan kebijakan izin yang memungkinkan pengguna untuk menghapus grup penyebaran bernama **WordPress_DepGroup** terkait dengan aplikasi bernama **WordPress_App** di **us-west-2** Wilayah.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:DeleteDeploymentGroup"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/
WordPress_DepGroup"
      ]
    }
  ]
}
```

Topik

- Customer-managed contoh kebijakan
- Praktik Terbaik Kebijakan
- Menggunakan CodeDeploy konsol
- Mengizinkan Pengguna Melihat Izin Mereka Sendiri

Customer-managed contoh kebijakan

Di bagian ini, Anda dapat menemukan contoh kebijakan yang memberikan izin untuk berbagai CodeDeploy tindakan. Kebijakan ini berfungsi saat Anda menggunakan CodeDeploy API, AWS SDK, atau. AWS CLI Anda harus memberikan izin tambahan untuk tindakan yang Anda lakukan di konsol. Untuk mempelajari lebih lanjut tentang pemberian izin konsol, lihat. Menggunakan CodeDeploy konsol

> **Note**
>
> Semua contoh menggunakan Region US West (Oregon) (`us-west-2`) dan berisi ID akun fiktif.

Contoh

- Contoh 1: Izinkan izin untuk melakukan CodeDeploy operasi di satu Wilayah
- Contoh 2: Izinkan izin untuk mendaftarkan revisi untuk satu aplikasi
- Contoh 3: Izinkan izin untuk membuat penerapan untuk satu grup penyebaran

Contoh 1: Izinkan izin untuk melakukan CodeDeploy operasi di satu Wilayah

Contoh berikut memberikan izin untuk melakukan CodeDeploy operasi di **`us-west-2`** Wilayah saja:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:*"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:*"
      ]
    }
  ]
}
```

Contoh 2: Izinkan izin untuk mendaftarkan revisi untuk satu aplikasi

Contoh berikut memberikan izin untuk mendaftarkan revisi aplikasi untuk semua aplikasi yang dimulai dengan **`Test`** di Wilayah: **`us-west-2`**

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codedeploy:RegisterApplicationRevision"
      ],
      "Resource" : [
        "arn:aws:codedeploy:us-west-2:111122223333:application:Test*"
      ]
    }
  ]
}
```

Contoh 3: Izinkan izin untuk membuat penerapan untuk satu grup penyebaran

Contoh berikut memungkinkan izin untuk membuat penerapan untuk grup penyebaran bernama **`WordPress_DepGroup`** terkait dengan aplikasi bernama, konfigurasi penerapan kustom bernama **`WordPress_AppThreeQuartersHealthy`**, dan revisi aplikasi apa pun yang terkait dengan aplikasi bernama. **`WordPress_App`** Semua sumber daya ini ada di **`us-west-2`** Wilayah.

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"
      ]
    }
  ]
}
```

Praktik Terbaik Kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus CodeDeploy sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi

selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan CodeDeploy konsol

Jika Anda menggunakan CodeDeploy konsol, Anda harus memiliki set izin minimum yang memungkinkan Anda menjelaskan AWS sumber daya lain untuk AWS akun Anda. Untuk digunakan CodeDeploy di CodeDeploy konsol, Anda harus memiliki izin dari layanan berikut:

- Amazon EC2 Auto Scaling
- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- AWS Identity and Access Management

- Amazon Simple Storage Service
- Layanan Notifikasi Sederhana Amazon
- Amazon CloudWatch

Jika Anda membuat kebijakan IAM yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan untuk pengguna yang memiliki peran dengan kebijakan IAM tersebut. Untuk memastikan bahwa pengguna tersebut masih dapat menggunakan CodeDeploy konsol, lampirkan juga kebijakan `AWSCodeDeployReadOnlyAccess` terkelola ke peran yang ditetapkan kepada pengguna, seperti yang dijelaskan dalamAWS kebijakan terkelola (standar) untuk CodeDeploy.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau CodeDeploy API.

Mengizinkan Pengguna Melihat Izin Mereka Sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI 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": "*"
        }
    ]
}
```

Pemecahan Masalah AWS CodeDeploy identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan CodeDeploy dan IAM.

Topik

- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya AWS akun untuk mengakses CodeDeploy sumber daya saya

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran CodeDeploy.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di CodeDeploy. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya AWS akun untuk mengakses CodeDeploy sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah CodeDeploy mendukung fitur-fitur ini, lihatBagaimana AWS CodeDeploy bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

CodeDeploy referensi izin

Gunakan tabel berikut saat menyiapkan kebijakan akses dan menulis izin yang dapat dilampirkan ke identitas IAM (kebijakan berbasis identitas). Tabel mencantumkan setiap operasi CodeDeploy API, tindakan yang dapat Anda berikan izin untuk melakukan tindakan, dan format ARN sumber daya yang akan digunakan untuk memberikan izin. Anda menentukan tindakan di bidang `Action`

kebijakan. Anda menentukan ARN, dengan atau tanpa karakter wildcard (*), sebagai nilai sumber daya dalam bidang `Resource` kebijakan.

Anda dapat menggunakan kunci kondisi AWS-wide dalam CodeDeploy kebijakan Anda untuk menyatakan kondisi. Untuk daftar lengkap tombol AWS-wide, lihat Kunci yang tersedia di Panduan Pengguna IAM.

Untuk menentukan tindakan, gunakan prefiks `codedeploy:` diikuti dengan nama operasi API (misalnya `codedeploy:GetApplication` dan `codedeploy:CreateApplication`). Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma (misalnya, `"Action": ["codedeploy:action1", "codedeploy:action2"]`).

Menggunakan Karakter Wildcard

Anda dapat menggunakan karakter wildcard (*) di ARN untuk menentukan beberapa tindakan atau sumber daya. Misalnya, `codedeploy:*` menentukan semua CodeDeploy tindakan dan `codedeploy:Get*` menentukan semua CodeDeploy tindakan yang dimulai dengan kata. `Get` Contoh berikut memberikan akses ke semua grup penyebaran dengan nama yang dimulai dengan `West` dan dikaitkan dengan aplikasi yang memiliki nama yang dimulai dengan. `Test`

```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

Anda dapat menggunakan wildcard dengan sumber daya berikut yang tercantum dalam tabel:

- *application-name*
- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

Wildcard tidak dapat digunakan dengan *region* atau*account-id*. Untuk informasi lebih lanjut tentang wildcard, lihat Pengidentifikasi IAM dalam Panduan Pengguna IAM.

> **Note**
>
> Dalam ARN untuk setiap tindakan, titik dua (:) mengikuti sumber daya. Anda juga dapat mengikuti sumber daya dengan garis miring (/). Untuk informasi selengkapnya, lihat CodeDeploycontoh ARN.

CodeDeploy Operasi API dan izin yang diperlukan untuk tindakan

AddTagsToOnPremisesInstances

Tindakan: `codedeploy:AddTagsToOnPremisesInstances`

Diperlukan untuk menambahkan tag ke satu atau beberapa instance lokal.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

BatchGetApplicationRevisions

Tindakan: `codedeploy:BatchGetApplicationRevisions`

Diperlukan untuk mendapatkan informasi tentang beberapa revisi aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

BatchGetApplications

Tindakan: `codedeploy:BatchGetApplications`

Diperlukan untuk mendapatkan informasi tentang beberapa aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:*`

BatchGetDeploymentGroups

Tindakan: `codedeploy:BatchGetDeploymentGroups`

Diperlukan untuk mendapatkan informasi tentang beberapa grup penyebaran yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

BatchGetDeploymentInstances

Tindakan: `codedeploy:BatchGetDeploymentInstances`

Diperlukan untuk mendapatkan informasi tentang satu atau lebih instance yang merupakan bagian dari grup penyebaran.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

BatchGetDeployments

Tindakan: `codedeploy:BatchGetDeployments`

Diperlukan untuk mendapatkan informasi tentang beberapa penerapan yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

BatchGetOnPremisesInstances

Tindakan: `codedeploy:BatchGetOnPremisesInstances`

Diperlukan untuk mendapatkan informasi tentang satu atau beberapa instans lokal.

Sumber daya: `arn:aws:codedeploy:region:account-id:*`

ContinueDeployment

Tindakan: `codedeploy:ContinueDeployment`

Diperlukan selama blue/green penerapan untuk mulai mendaftarkan instans di lingkungan pengganti dengan penyeimbang beban Elastic Load Balancing.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

CreateApplication

Tindakan: `codedeploy:CreateApplication`

Diperlukan untuk membuat aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:application:application-name`

CreateDeployment[1]

Tindakan: `codedeploy:CreateDeployment`

Diperlukan untuk membuat penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

CreateDeploymentConfig

Tindakan: `codedeploy:CreateDeploymentConfig`

Diperlukan untuk membuat konfigurasi penerapan kustom yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

CreateDeploymentGroup

Tindakan: `codedeploy:CreateDeploymentGroup`

Diperlukan untuk membuat grup penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeleteApplication

Tindakan: `codedeploy:DeleteApplication`

Diperlukan untuk menghapus aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:`*`application-name`*

DeleteDeploymentConfig

Tindakan: `codedeploy:DeleteDeploymentConfig`

Diperlukan untuk menghapus konfigurasi penerapan kustom yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentconfig/`*`deployment-configuration-name`*

DeleteDeploymentGroup

Tindakan: `codedeploy:DeleteDeploymentGroup`

Diperlukan untuk menghapus grup penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

DeregisterOnPremisesInstance

Tindakan: `codedeploy:DeregisterOnPremisesInstance`

Diperlukan untuk membatalkan pendaftaran instans lokal.

Sumber daya: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

GetApplication

Tindakan: `codedeploy:GetApplication`

Diperlukan untuk mendapatkan informasi tentang satu aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:application:application-name`

GetApplicationRevision

Tindakan: `codedeploy:GetApplicationRevision`

Diperlukan untuk mendapatkan informasi tentang revisi aplikasi tunggal untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:application:application-name`

GetDeployment

Tindakan: `codedeploy:GetDeployment`

Diperlukan untuk mendapatkan informasi tentang penyebaran tunggal ke grup penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

GetDeploymentConfig

Tindakan: `codedeploy:GetDeploymentConfig`

Diperlukan untuk mendapatkan informasi tentang konfigurasi penerapan tunggal yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

GetDeploymentGroup

Tindakan: `codedeploy:GetDeploymentGroup`

Diperlukan untuk mendapatkan informasi tentang grup penyebaran tunggal untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

GetDeploymentInstance

Tindakan: `codedeploy:GetDeploymentInstance`

Diperlukan untuk mendapatkan informasi tentang satu instance dalam penerapan yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:deploymentgroup:`*`application-name`*`/`*`deployment-group-name`*

GetOnPremisesInstance

Tindakan: `codedeploy:GetOnPremisesInstance`

Diperlukan untuk mendapatkan informasi tentang satu instans lokal.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:instance/`*`instance-ID`*

ListApplicationRevisions

Tindakan: `codedeploy:ListApplicationRevisions`

Diperlukan untuk mendapatkan informasi tentang semua revisi aplikasi untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:*`

ListApplications

Tindakan: `codedeploy:ListApplications`

Diperlukan untuk mendapatkan informasi tentang semua aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:`*`region`*`:`*`account-id`*`:application:*`

ListDeploymentConfigs

Tindakan: `codedeploy:ListDeploymentConfigs`

Diperlukan untuk mendapatkan informasi tentang semua konfigurasi penerapan yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

ListDeploymentGroups

Tindakan: `codedeploy:ListDeploymentGroups`

Diperlukan untuk mendapatkan informasi tentang semua grup penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

ListDeploymentInstances

Tindakan: `codedeploy:ListDeploymentInstances`

Diperlukan untuk mendapatkan informasi tentang semua instance dalam penerapan yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListDeployments

Tindakan: `codedeploy:ListDeployments`

Diperlukan untuk mendapatkan informasi tentang semua penerapan ke grup penyebaran yang terkait dengan pengguna, atau untuk mendapatkan semua penerapan yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

ListGitHubAccountTokenNames

Tindakan: `codedeploy:ListGitHubAccountTokenNames`

Diperlukan untuk mendapatkan daftarnama-nama koneksi yang disimpan ke GitHub akun.

Sumber daya: `arn:aws:codedeploy:region:account-id:*`

ListOnPremisesInstances

Tindakan: `codedeploy:ListOnPremisesInstances`

Diperlukan untuk mendapatkan daftar satu atau beberapa nama instans lokal.

Sumber daya: `arn:aws:codedeploy:region:account-id:*`

RegisterApplicationRevision

Tindakan: `codedeploy:RegisterApplicationRevision`

Diperlukan untuk mendaftarkan informasi tentang revisi aplikasi untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:application:application-name`

RegisterOnPremisesInstance

Tindakan: `codedeploy:RegisterOnPremisesInstance`

Diperlukan untuk mendaftarkan instans lokal dengan CodeDeploy.

Sumber daya: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

RemoveTagsFromOnPremisesInstances

Tindakan: `codedeploy:RemoveTagsFromOnPremisesInstances`

Diperlukan untuk menghapus tag dari satu atau beberapa instance lokal.

Sumber daya: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

SkipWaitTimeForInstanceTermination

Tindakan: `codedeploy:SkipWaitTimeForInstanceTermination`

Diperlukan untuk mengganti waktu tunggu yang ditentukan dan mulai menghentikan instance di lingkungan asli segera setelah lalu lintas berhasil dirutekan dalam penerapan. blue/green

Sumber daya: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

StopDeployment

Tindakan: `codedeploy:StopDeployment`

Diperlukan untuk menghentikan penyebaran yang sedang berlangsung ke grup penyebaran untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

UpdateApplication[3]

Tindakan: `codedeploy:UpdateApplication`

Diperlukan untuk mengubah informasi tentang aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:application:application-name`

UpdateDeploymentGroup[3]

Tindakan: `codedeploy:UpdateDeploymentGroup`

Diperlukan untuk mengubah informasi tentang grup penyebaran tunggal untuk aplikasi yang terkait dengan pengguna.

Sumber daya: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

[1] Saat Anda menentukan `CreateDeployment` izin, Anda juga harus menentukan `GetDeploymentConfig` izin untuk konfigurasi penerapan dan `GetApplicationRevision` atau `RegisterApplicationRevision` izin untuk revisi aplikasi.

[2] Berlaku untuk `ListDeployments` saat Anda menyediakan grup penerapan, tetapi tidak saat Anda mencantumkan semua penerapan yang terkait dengan pengguna.

[3] Untuk`UpdateApplication`, Anda harus memiliki `UpdateApplication` izin untuk nama aplikasi lama dan baru. Untuk `UpdateDeploymentGroup` tindakan yang melibatkan perubahan nama grup penerapan, Anda harus memiliki `UpdateDeploymentGroup` izin untuk nama grup penerapan lama dan baru.

Cross-service pencegahan wakil bingung

Masalah "confused deputy" adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memengaruhi entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Cross-service peniruan identitas dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara

yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS sediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Kami merekomendasikan untuk menggunakan kunci konteks kondisi SourceAccount global aws: SourceArn dan aws: dalam kebijakan sumber daya untuk membatasi izin yang CodeDeploy memberikan layanan lain ke sumber daya. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin sumber daya apa pun di akun itu dikaitkan dengan penggunaan lintas layanan.

Untuk EC2/On-Premises, AWS Lambda, dan penerapan Amazon ECS reguler, nilai `aws:SourceArn` harus mencakup ARN CodeDeploy grup penyebaran yang CodeDeploy diizinkan untuk mengambil peran IAM.

Untuk blue/greenpenerapan Amazon ECS yang dibuat melalui CloudFormation, nilai `aws:SourceArn` harus menyertakan tumpukan CloudFormation ARN yang diizinkan untuk mengambil peran CodeDeploy IAM.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan `aws:SourceArn` kunci dengan ARN penuh sumber daya. Jika Anda tidak mengetahui ARN lengkap atau jika Anda menentukan beberapa sumber daya, gunakan karakter wildcard (*) untuk bagian yang tidak diketahui.

Misalnya, Anda dapat menggunakan kebijakan kepercayaan berikut dengan penerapan EC2/On-Premises, AWS Lambda, atau Amazon ECS reguler:

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:deployment-group:myApplication/*"
                }
            }
        }
    ]
}
```

Untuk blue/greenpenerapan Amazon ECS yang dibuat melalui CloudFormation, Anda dapat menggunakan:

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/*"
                }
            }
        }
    ]
}
```

Penebangan dan pemantauan di CodeDeploy

Bagian ini memberikan gambaran umum tentang pemantauan, pencatatan, dan respons insiden di CodeDeploy.

Audit semua interaksi dengan CodeDeploy

CodeDeploy terintegrasi dengan AWS CloudTrail, layanan yang menangkap panggilan API yang dilakukan oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket S3 yang Anda tentukan. CodeDeploy CloudTrail menangkap panggilan API dari CodeDeploy konsol, dari CodeDeploy perintah melalui AWS CLI, atau dari CodeDeploy API secara langsung. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan mana yang dibuat CodeDeploy, alamat IP sumber dari mana permintaan itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan sebagainya. Untuk mempelajari selengkapnya CloudTrail, lihat Bekerja dengan file CloudTrail log di Panduan AWS CloudTrail Pengguna.

Anda dapat melihat data log yang dibuat oleh CodeDeploy penerapan dengan menyiapkan CloudWatch agen Amazon untuk melihat data agregat di CloudWatch konsol atau dengan masuk ke instance untuk meninjau file log. Untuk informasi selengkapnya, lihat Kirim log CodeDeploy agen ke CloudWatch.

Peringatan dan manajemen insiden

Anda dapat menggunakan Amazon CloudWatch Events untuk mendeteksi dan bereaksi terhadap perubahan status instans atau penerapan (peristiwa) dalam CodeDeploy operasi Anda. Kemudian, berdasarkan aturan yang Anda buat, CloudWatch Acara memanggil satu atau beberapa tindakan target saat penerapan atau instance memasuki status yang Anda tentukan dalam aturan. Bergantung pada jenis perubahan status, Anda mungkin ingin mengirim pemberitahuan, menangkap informasi status, mengambil tindakan korektif, memulai peristiwa, atau mengambil tindakan lain. Anda dapat memilih jenis target berikut ketika Anda menggunakan CloudWatch Acara sebagai bagian dari CodeDeploy operasi Anda:

- AWS Lambda fungsi
- Aliran Kinesis
- Antrian Amazon SQS SQS
- Built-in target (tindakan CloudWatch alarm)
- Topik Amazon SNS

Berikut ini adalah beberapa kasus penggunaan:

- Gunakan fungsi Lambda untuk meneruskan notifikasi ke saluran Slack setiap kali penerapan gagal.
- Dorong data tentang penerapan atau instans ke aliran Kinesis untuk mendukung pemantauan status real-time yang komprehensif.
- Gunakan tindakan CloudWatch alarm untuk menghentikan, menghentikan, me-reboot, atau memulihkan instans EC2 secara otomatis saat peristiwa penerapan atau instans yang Anda tentukan terjadi.

Untuk informasi selengkapnya, lihat Apa itu CloudWatch Acara Amazon di Panduan CloudWatch Pengguna Amazon.

Validasi kepatuhan untuk AWS CodeDeploy

Pengalaman AWS pendaftaran baru kami tidak dirancang untuk beban kerja yang diatur. Jika Anda menggunakan pengalaman AWS pendaftaran baru kami, tetapi Anda ingin menggunakannya AWS untuk beban kerja yang diatur, Anda dapat mendaftar untuk AWS (lanjutan) atau mengaktifkan fitur lanjutan untuk lingkungan Anda AWS .

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat Dokumentasi AWS Keamanan.

Ketahanan di AWS CodeDeploy

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis

mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat infrastruktur AWS global.

Keamanan infrastruktur di AWS CodeDeploy

Sebagai layanan terkelola, AWS CodeDeploy dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam Amazon Web Services: Ringkasan proses keamanan whitepaper.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses CodeDeploy melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2 atau versi yang lebih baru. Kami merekomendasikan TLS 1.3 atau versi yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS), seperti Ephemeral (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan prinsipal IAM. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Referensi

Referensi.

Topik

- CodeDeploy AppSpec referensi file
- CodeDeploy referensi konfigurasi agen
- CloudFormation template untuk CodeDeploy referensi
- Gunakan CodeDeploy dengan Amazon Virtual Private Cloud
- CodeDeploy referensi kit sumber daya
- CodeDeploy kuota

CodeDeploy AppSpec referensi file

Bagian ini hanya referensi. Untuk ikhtisar konseptual AppSpec file, lihatApplication Specification Files.

File spesifikasi aplikasi (AppSpec file) adalah format YAML-atau JSON-formatted file yang digunakan oleh CodeDeploy untuk mengelola penyebaran.

> **Note**
>
> AppSpec File untuk EC2/On-Premises penerapan harus diberi nama`appspec.yml`, kecuali jika Anda melakukan penerapan lokal. Untuk informasi selengkapnya, lihat Buat penerapan lokal.

Topik

- AppSpec file pada platform komputasi Amazon ECS
- AppSpec berkas pada sebuah AWS Lambda platform komputasi
- AppSpec file pada platform EC2/on-premises komputasi
- AppSpec Struktur berkas
- AppSpec Contoh berkas
- AppSpec Spasi file

- Validasi AppSpec File dan lokasi file Anda

AppSpec file pada platform komputasi Amazon ECS

Untuk aplikasi platform komputasi Amazon ECS, AppSpec file tersebut digunakan CodeDeploy untuk menentukan:

- File definisi tugas Amazon ECS Anda. Ini ditentukan dengan ARN dalam `TaskDefinition` instruksi dalam file. AppSpec
- Container dan port dalam tugas pengganti Anda ditetapkan di mana Application Load Balancer atau Network Load Balancer mengalihkan lalu lintas selama penerapan. Ini ditentukan dengan `LoadBalancerInfo` instruksi dalam AppSpec file.
- Informasi opsional tentang layanan Amazon ECS Anda, seperti versi platform yang dijalankannya, subnetnya, dan grup keamanannya.
- Fungsi Lambda opsional untuk dijalankan selama pengait yang sesuai dengan peristiwa siklus hidup selama penerapan Amazon ECS. Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait' untuk penerapan Amazon ECS.

AppSpec berkas pada sebuah AWS Lambda platform komputasi

Untuk aplikasi platform komputasi AWS Lambda, AppSpec file digunakan oleh CodeDeploy untuk menentukan:

- Versi fungsi Lambda mana yang akan diterapkan.
- Lambda mana yang berfungsi untuk digunakan sebagai tes validasi.

AppSpec File dapat berupa YAML-formatted atau JSON-formatted. Anda juga dapat memasukkan konten AppSpec file langsung ke CodeDeploy konsol saat Anda membuat penerapan.

AppSpec file pada platform EC2/on-premises komputasi

Jika aplikasi Anda menggunakan platform EC2/On-Premises komputasi, AppSpec file harus berupa YAML-formatted file bernama `appspec.yml` dan harus ditempatkan di root struktur direktori kode sumber aplikasi. Jika tidak, deployment gagal. Ini digunakan oleh CodeDeploy untuk menentukan:

- Apa yang harus diinstal ke instance Anda dari revisi aplikasi Anda di Amazon S3 atau. GitHub

- Peristiwa siklus hidup mana yang akan dijalankan sebagai respons terhadap peristiwa siklus hidup penerapan.

Setelah Anda memiliki AppSpec file yang lengkap, Anda bundel, bersama dengan konten yang akan digunakan, ke dalam file arsip (zip, tar, atau tar terkompresi). Untuk informasi selengkapnya, lihat Bekerja dengan revisi aplikasi untuk CodeDeploy.

> **Note**
>
> Format file arsip tar dan tar terkompresi (.tar dan.tar.gz) tidak didukung untuk instance Windows Server.

Setelah Anda memiliki file arsip yang dibundel (dikenal CodeDeploy sebagai revisi), Anda mengunggahnya ke bucket Amazon S3 atau repositori Git. Kemudian Anda gunakan CodeDeploy untuk menyebarkan revisi. Untuk petunjuk, lihat Buat penerapan dengan CodeDeploy.

Appspec.yl untuk penerapan platform EC2/On-Premises komputasi disimpan di direktori root revisi Anda. Untuk informasi selengkapnya, lihat Menambahkan AppSpec file untuk EC2/On-Premises penerapan dan Rencanakan revisi untuk CodeDeploy.

AppSpec Struktur berkas

Berikut ini adalah struktur tingkat tinggi untuk AppSpec file yang digunakan untuk penyebaran ke AWS Lambda dan platform komputasi. EC2/On-Premises

Nilai dalam YAML-formatted AppSpec file yang merupakan string tidak boleh dibungkus dengan tanda kutip (“”) kecuali ditentukan lain.

AppSpec struktur file untuk penerapan Amazon ECS

> **Note**
>
> AppSpec File ini ditulis dalam YAMG, tetapi Anda dapat menggunakan struktur yang sama untuk menulis satu di JSON. String dalam JSON-formatted AppSpec file selalu dibungkus dengan tanda kutip (“”).

```
version: 0.0
```

```
resources:
  ecs-service-specifications
hooks:
  deployment-lifecycle-event-mappings
```

Dalam struktur ini:

versi

Bagian ini menentukan versi AppSpec file. Jangan ubah nilai ini. Hal ini wajib diisi. Saat ini, satu-satunya nilai yang diizinkan adalah**0.0**. Ini dicadangkan oleh CodeDeploy untuk penggunaan masa depan.

Tentukan versi dengan string.

sumber daya

Bagian ini menentukan informasi tentang aplikasi Amazon ECS untuk digunakan.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

kait

Bagian ini menentukan fungsi Lambda untuk dijalankan pada kait peristiwa siklus hidup penerapan tertentu untuk memvalidasi penerapan.

Untuk informasi selengkapnya, lihat Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS.

AppSpec struktur file untuk AWS Penerapan Lambda

> **Note**
>
> AppSpec File ini ditulis dalam YAMB, tetapi Anda dapat menggunakan struktur yang sama untuk menulis AppSpec file untuk penyebaran Lambda di JSON. String dalam JSON-formatted AppSpec file selalu dibungkus dengan tanda kutip (“”).

```
version: 0.0
```

```
resources:
  lambda-function-specifications
hooks:
  deployment-lifecycle-event-mappings
```

Dalam struktur ini:

versi

Bagian ini menentukan versi AppSpec file. Jangan ubah nilai ini. Hal ini wajib diisi. Saat ini, satu-satunya nilai yang diizinkan adalah**`0.0`**. Ini dicadangkan oleh CodeDeploy untuk penggunaan masa depan.

Tentukan versi dengan string.

sumber daya

Bagian ini menentukan informasi tentang fungsi Lambda untuk menyebarkan.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'sumber daya' (Amazon ECS dan AWS Lambda hanya penerapan).

kait

Bagian ini menentukan fungsi Lambda untuk dijalankan pada peristiwa siklus hidup penerapan tertentu untuk memvalidasi penerapan.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait'.

AppSpec struktur file untuk EC2/On-Premises penerapan

```
version: 0.0
os: operating-system-name
files:
  source-destination-files-mappings
permissions:
  permissions-specifications
hooks:
  deployment-lifecycle-event-mappings
```

Dalam struktur ini:

versi

Bagian ini menentukan versi AppSpec file. Jangan ubah nilai ini. Hal ini wajib diisi. Saat ini, satu-satunya nilai yang diizinkan adalah**`0.0`**. Ini dicadangkan oleh CodeDeploy untuk penggunaan masa depan.

Tentukan versi dengan string.

os

Bagian ini menentukan nilai sistem operasi dari instance yang Anda gunakan. Hal ini wajib diisi. Nilai-nilai berikut dapat ditentukan:

- linux — Instans ini adalah Amazon Linux, Ubuntu Server, RHEL, atau macOS instance.
- windows — Instans adalah instance Windows Server.

Tentukan os dengan string.

berkas

Bagian ini menentukan nama-nama file yang harus disalin ke instance selama acara Install deployment.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan).

izin

Bagian ini menentukan bagaimana izin khusus, jika ada, harus diterapkan ke file di `files` bagian saat mereka sedang disalin ke instance. Bagian ini hanya berlaku untuk Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL), dan instance macOS.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'izin' (hanya EC2/On-Premises penerapan).

kait

Bagian ini menentukan skrip untuk dijalankan pada peristiwa siklus hidup penerapan tertentu selama penerapan.

Untuk informasi selengkapnya, lihat AppSpec Bagian 'kait'.

Topik

- AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan)
- AppSpec Bagian 'sumber daya' (Amazon ECS dan AWS Lambda hanya penerapan)

- AppSpec Bagian 'izin' (hanya EC2/On-Premises penerapan)
- AppSpec Bagian 'kait'

AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan)

Memberikan informasi CodeDeploy tentang file mana dari revisi aplikasi Anda yang harus diinstal pada instance selama peristiwa Install penerapan. Bagian ini diperlukan hanya jika Anda menyalin file dari revisi ke lokasi pada instance selama penerapan.

Bagian ini memiliki struktur sebagai berikut:

```
files:
  - source: source-file-location-1
    destination: destination-file-location-1
file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

Beberapa `source` dan `destination` pasangan dapat diatur.

`source`Instruksi mengidentifikasi file atau direktori dari revisi Anda untuk menyalin ke instance:

- Jika `source` mengacu pada file, hanya file yang ditentukan yang disalin ke instance.
- Jika `source` mengacu pada direktori, maka semua file dalam direktori disalin ke instance.
- Jika `source` merupakan garis miring tunggal (“/” untuk instance Amazon Linux, RHEL, dan Ubuntu Server, atau “\” untuk instance Windows Server), maka semua file dari revisi Anda disalin ke instance.

Jalur yang `source` digunakan relatif terhadap `appspec.yml` file, yang seharusnya berada di akar revisi Anda. Untuk detail tentang struktur file revisi, lihatRencanakan revisi untuk CodeDeploy.

`destination`Instruksi mengidentifikasi lokasi pada contoh di mana file harus disalin. Ini harus menjadi jalur yang sepenuhnya memenuhi syarat seperti `/root/destination/directory` (di Linux, RHEL, dan Ubuntu) atau `c:\destination\folder` (di Windows).

`source`dan `destination` masing-masing ditentukan dengan string.

`file_exists_behavior`Instruksi bersifat opsional, dan menentukan cara CodeDeploy menangani file yang sudah ada di lokasi target penerapan tetapi bukan bagian dari penerapan yang berhasil sebelumnya. Pengaturan ini dapat mengambil salah satu nilai berikut:

- DILARANG: Penerapan gagal. Ini juga merupakan perilaku default jika tidak ada opsi yang ditentukan.
- OVERWRITE: Versi file dari revisi aplikasi yang saat ini sedang digunakan menggantikan versi yang sudah ada di instance.
- MEMPERTAHANKAN: Versi file yang sudah ada di instance disimpan dan digunakan sebagai bagian dari penerapan baru.

Saat menggunakan `file_exists_behavior` pengaturan, pahami bahwa pengaturan ini:

- hanya dapat ditentukan sekali, dan berlaku untuk semua file dan direktori yang tercantum di bawah`files:`.
- lebih diutamakan daripada `--file-exists-behavior` AWS CLI opsi dan opsi `fileExistsBehavior` API (keduanya juga opsional).

Berikut adalah `files` bagian contoh untuk Amazon Linux, Ubuntu Server, atau RHEL instance.

```
files:
  - source: Config/config.txt
    destination: /webapps/Config
  - source: source
    destination: /webapps/myApp
```

Dalam contoh ini, dua operasi berikut dilakukan selama acara Install:

1. Salin `Config/config.txt` file dalam revisi Anda ke `/webapps/Config/config.txt` jalur pada instance.
2. Salin secara rekursif semua file di direktori revisi Anda ke `source` `/webapps/myApp` direktori pada instance.

Contoh bagian 'File'

Contoh berikut menunjukkan cara menentukan `files` bagian. Meskipun contoh-contoh ini menggambarkan struktur file dan direktori (folder) Windows Server, mereka dapat dengan mudah diadaptasi untuk Amazon Linux, Ubuntu Server, dan instance RHEL.

> **Note**
>
> Hanya EC2/On-Premises penerapan yang menggunakan bagian ini. `files` Itu tidak berlaku untuk penyebaran AWS Lambda.

Untuk contoh berikut, kami berasumsi file-file ini muncul di bundel di root`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
```

Untuk contoh berikut, kita asumsikan `appspec.yml` muncul dalam bundel di root `source` bersama dengan folder bernama `my-folder` yang berisi tiga file:

- `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 Bagian 'sumber daya' (Amazon ECS dan AWS Lambda hanya penerapan)

Konten di `'resources'` bagian AppSpec file bervariasi, tergantung pada platform komputasi penerapan Anda. `'resources'`Bagian untuk penyebaran Amazon ECS berisi definisi tugas Amazon ECS, wadah, dan port untuk merutekan lalu lintas ke set tugas Amazon ECS yang diperbarui, dan informasi opsional lainnya. `'resources'`Bagian untuk AWS Lambda penerapan berisi nama, alias, versi saat ini, dan versi target dari fungsi Lambda.

Topik

- AppSpec Bagian 'sumber daya' untuk AWS Penerapan Lambda
- AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS

AppSpec Bagian 'sumber daya' untuk AWS Penerapan Lambda

`'resources'`Bagian ini menentukan fungsi Lambda untuk menyebarkan dan memiliki struktur berikut:

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"
            }
        }
    }
]
```

Setiap properti ditentukan dengan string.

- `name` – Wajib. Ini adalah nama fungsi Lambda yang akan digunakan.
- `alias` – Wajib. Ini adalah nama alias untuk fungsi Lambda.
- `currentversion` – Wajib. Ini adalah versi lalu lintas fungsi Lambda yang saat ini menunjuk ke. Nilai ini harus berupa bilangan bulat positif yang valid.
- `targetversion` – Wajib. Ini adalah versi lalu lintas fungsi Lambda yang digeser ke. Nilai ini harus berupa bilangan bulat positif yang valid.

AppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS

`'resources'` Bagian ini menentukan layanan Amazon ECS untuk diterapkan dan memiliki struktur berikut:

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
                }
            ]
        }
    }
]
```

Setiap properti ditentukan dengan string kecuali untuk`ContainerPort`, yang merupakan angka.

- `TaskDefinition` – Wajib. Ini adalah definisi tugas untuk layanan Amazon ECS untuk diterapkan. Ini ditentukan dengan ARN dari definisi tugas. Format ARN adalah. `arn:aws:ecs:`*`aws-region`*`:`*`account-id`*`:task-definition/`*`task-definition-family`*`:`*`task-definition-revision`* Untuk informasi selengkapnya, lihat Nama Sumber Daya Amazon (ARN) dan ruang nama AWS layanan.

  > **Note**
  >
  > `:`*`task-definition-revision`*Bagian dari ARN adalah opsional. Jika dihilangkan, Amazon ECS menggunakan revisi ACTIVE terbaru dari definisi tugas.

- `ContainerName` – Wajib. Ini adalah nama wadah Amazon ECS yang berisi aplikasi Amazon ECS Anda. Itu harus berupa wadah yang ditentukan dalam definisi tugas Amazon ECS Anda.
- `ContainerPort` – Wajib. Ini adalah port pada kontainer tempat lalu lintas akan diarahkan ke.

- `PlatformVersion`: Opsional. Versi platform tugas Fargate dalam layanan Amazon ECS yang dikerahkan. Untuk informasi selengkapnya, lihat Versi platform AWS Fargate. Jika tidak ditentukan, `LATEST` digunakan secara default.
- `NetworkConfiguration`: Opsional. Di bawah`AwsvpcConfiguration`, Anda dapat menentukan yang berikut ini. Untuk informasi selengkapnya, lihat AwsVpcConfigurationdi Referensi API Amazon ECS Container Service.
  - `Subnets`: Opsional. Daftar terpisah koma dari satu atau beberapa subnet di layanan Amazon ECS Anda.
  - `SecurityGroups`: Opsional. Daftar terpisah koma dari satu atau beberapa grup keamanan di Amazon Elastic Container Service Anda.
  - `AssignPublicIp`: Opsional. String yang menentukan apakah elastic network interface layanan Amazon ECS Anda menerima alamat IP publik. Nilai yang valid adalah `ENABLED` dan `DISABLED`.

  > **Note**
  >
  > Semua atau tidak ada pengaturan di bawah `NetworkConfiguration` harus ditentukan. Misalnya, jika Anda ingin menentukan`Subnets`, Anda juga harus menentukan `SecurityGroups` dan`AssignPublicIp`. Jika tidak ada yang ditentukan, CodeDeploy gunakan pengaturan Amazon ECS jaringan saat ini.

- `CapacityProviderStrategy`: Opsional. Daftar penyedia kapasitas Amazon ECS yang ingin Anda gunakan untuk penyebaran Anda. Untuk informasi selengkapnya, lihat Penyedia kapasitas Amazon ECS di Panduan Pengembang Layanan Kontainer Elastis Amazon. Untuk setiap penyedia kapasitas, Anda dapat menentukan pengaturan berikut. Untuk detail tentang pengaturan ini, lihat AWS: :ECS:: ServiceCapacityProviderStrategyItem di Panduan Pengguna AWS CloudFormation
  - `Base`: Opsional. Nilai dasar menandakan berapa banyak tugas, dengan angka minimum, yang berjalan pada penyedia kapasitas tertentu. Hanya satu penyedia kapasitas di strategi penyedia kapasitas yang dapat menentukan nilai dasar. Jika tidak ada nilai yang ditentukan, nilai default 0 digunakan.
  - `CapacityProvider`: Opsional. Nama singkat dari penyedia kapasitas. Contoh: CapacityProvidera
  - `Weight`: Opsional.

Nilai bobot menandakan persentase relatif dari jumlah total tugas yang diluncurkan yang harus menggunakan penyedia kapasitas tertentu. Nilai `weight` dipertimbangkan setelah nilai `base`, jika ditentukan, dianggap memuaskan.

Jika tidak ada nilai `weight` yang ditentukan, nilai default `0` digunakan. Ketika beberapa penyedia kapasitas ditentukan dalam strategi penyedia kapasitas, setidaknya satu dari penyedia kapasitas harus memiliki nilai bobot lebih besar dari nol dan penyedia kapasitas apa pun dengan bobot `0` tidak akan digunakan untuk menempatkan tugas. Jika Anda menentukan beberapa penyedia kapasitas dalam strategi di mana semua memiliki bobot `0`, tindakan `RunTask` atau `CreateService` apa pun yang menggunakan strategi penyedia kapasitas akan gagal.

Contoh skenario untuk penggunaan bobot yang menentukan strategi yang berisi dua penyedia kapasitas dan keduanya memiliki bobot 1, jadi ketika `base` dianggap memuaskan, tugas akan dibagi secara merata ke dua penyedia kapasitas. Dengan menggunakan logika yang sama, jika Anda menentukan bobot 1 untuk capacityProviderA dan bobot 4 untuk capacityProviderB, maka untuk setiap satu tugas yang berjalan menggunakan capacityProviderA, empat tugas akan menggunakan capacityProviderB.

AppSpec Bagian 'izin' (hanya EC2/On-Premises penerapan)

`'permissions'`Bagian ini menentukan bagaimana izin khusus, jika ada, harus diterapkan ke file dan directories/folders di `'files'` bagian setelah mereka disalin ke instance. Anda dapat menentukan beberapa `object` instruksi. Bagian ini opsional. Ini hanya berlaku untuk Amazon Linux, Ubuntu Server, RHEL, dan macOS.

> **Note**
>
> Pada instans macOS, `context` instruksi `acls` dan tidak didukung. Penerapan yang AppSpec filenya menggunakan salah satu dari mereka di macOS gagal sebelum file apa pun diinstal. Instruksi lainnya berfungsi seperti yang dijelaskan di bagian ini.

> **Note**
>
> `'permissions'`Bagian ini hanya digunakan untuk EC2/On-Premises penerapan. Ini tidak digunakan untuk penyebaran AWS Lambda atau Amazon ECS.

Bagian ini memiliki struktur sebagai berikut:

```
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
```

Instruksi adalah sebagai berikut:

- `object` – Wajib. Ini adalah satu set objek sistem file (file atau directories/folders) yang izin yang ditentukan diterapkan setelah objek sistem file disalin ke instance.

  Tentukan `object` dengan string.
- `pattern` – Opsional. Menentukan pola untuk menerapkan izin. Jika tidak ditentukan atau ditentukan dengan karakter khusus**"**"**, izin diterapkan ke semua file atau direktori yang cocok, tergantung pada file. `type`

  Tentukan `pattern` dengan string dengan tanda kutip (“”).
- `except` – Opsional. Menentukan setiap file atau direktori yang pengecualian untuk. `pattern`

  Tentukan `except` dengan daftar string yang dipisahkan koma di dalam tanda kurung siku.
- `owner` – Opsional. Nama pemilik`object`. Jika tidak ditentukan, semua pemilik yang ada yang diterapkan ke file atau directory/folder struktur asli tetap tidak berubah setelah operasi penyalinan.

  Tentukan `owner` dengan string.
- `group` – Opsional. Nama grup untuk`object`. Jika tidak ditentukan, semua grup yang ada diterapkan ke file atau directory/folder struktur asli tetap tidak berubah setelah operasi penyalinan.

  Tentukan `group` dengan string.

- `mode` – Opsional. Nilai numerik yang menentukan izin yang akan diterapkan. `object` Pengaturan mode mengikuti sintaks perintah chmod Linux.

  > **Important**
  >
  > Jika nilainya termasuk nol di depan, Anda harus mengelilinginya dengan tanda kutip ganda, atau menghapus nol di depan sehingga hanya tersisa tiga digit.

  > **Note**
  >
  > Notasi simbolik seperti tidak **`u+x`** didukung untuk pengaturan. `mode`

  Contoh:

  - `mode: "0644"`memberikan izin baca dan tulis kepada pemilik objek (6), izin hanya-baca ke grup (4), dan izin hanya-baca untuk semua pengguna lain (4).
  - `mode: 644`memberikan izin yang sama seperti. `mode: "0644"`
  - `mode: 4755`menetapkan atribut setuid (4), memberikan izin kontrol penuh kepada pemilik (7), memberikan izin baca dan eksekusi ke grup (5), dan memberikan izin baca dan eksekusi ke semua pengguna lain (5).

    Untuk contoh lainnya, lihat dokumentasi perintah Linux chmod.

    Jika mode tidak ditentukan, semua mode yang ada diterapkan ke file asli atau struktur folder tetap tidak berubah setelah operasi penyalinan.

- `acls` – Opsional. Daftar string karakter yang mewakili satu atau lebih entri daftar kontrol akses (ACL) yang diterapkan. `object` Misalnya, **`u:bob:rw`** mewakili izin baca dan tulis untuk pengguna**bob**. (Untuk contoh lainnya, lihat contoh format entri ACL dalam dokumentasi `setfacl` perintah Linux.) Anda dapat menentukan beberapa entri ACL. Jika tidak `acls` ditentukan, ACL yang ada diterapkan ke file atau directory/folder struktur asli tetap tidak berubah setelah operasi penyalinan. Ini menggantikan ACL yang ada.

  Tentukan `acls` dengan tanda hubung (-), diikuti oleh spasi, dan kemudian string (misalnya,`- u:jane:rw`). Jika Anda memiliki lebih dari satu ACL, masing-masing ditentukan pada baris terpisah.

> **Note**
>
> Menyetel pengguna yang tidak disebutkan namanya, grup yang tidak disebutkan namanya, atau entri ACL serupa lainnya menyebabkan file gagal. AppSpec Gunakan `mode` untuk menentukan jenis izin ini sebagai gantinya.

- `context` – Opsional. Untuk instance yang diaktifkan Security-Enhanced Linux (SELinux), daftar label konteks yang relevan dengan keamanan untuk diterapkan ke objek yang disalin. Label ditentukan sebagai kunci yang berisi`user`,`type`, dan`range`. (Untuk informasi lebih lanjut, lihat dokumentasi SELinux.) Setiap kunci dimasukkan dengan string. Jika tidak ditentukan, label apa pun yang ada yang diterapkan pada file atau directory/folder struktur asli tetap tidak berubah setelah operasi penyalinan.
  - `user` – Opsional. Pengguna SELinux.
  - `type` – Opsional. Nama tipe SELinux.
  - `range` – Opsional. Penentu rentang SELinux. Ini tidak berpengaruh kecuali Multi-Level Keamanan (MLS) dan Multi-Category Keamanan (MCS) diaktifkan pada mesin. Jika tidak diaktifkan, `range` default ke. **`s0`**

  Tentukan `context` dengan string (misalnya,`user: unconfined_u`). Masing-masing `context` ditentukan pada garis terpisah.
- `type` – Opsional. Jenis objek yang menerapkan izin yang ditentukan. `type`adalah string yang dapat diatur ke **`file`** atau**`directory`**. Jika **`file`** ditentukan, izin diterapkan hanya untuk file yang segera terkandung dalam `object` setelah operasi salinan (dan bukan untuk `object` dirinya sendiri). Jika **`directory`** ditentukan, izin diterapkan secara rekursif ke semua directories/folders yang ada di mana saja `object` setelah operasi penyalinan (tetapi tidak untuk `object` dirinya sendiri).

  Tentukan `type` dengan tanda hubung (-), diikuti oleh spasi, dan kemudian string (misalnya,`- file`).

Contoh bagian 'Izin'

Contoh berikut menunjukkan cara menentukan `'permissions'` bagian dengan`object`,,`pattern`,`except`, `owner`,`mode`, dan `type` instruksi. Contoh ini hanya berlaku untuk Amazon Linux, Ubuntu Server, dan instans RHEL. Dalam contoh ini, asumsikan file dan folder berikut disalin ke instance dalam hierarki ini:

```
/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
```

AppSpec File berikut menunjukkan cara mengatur izin pada file dan folder ini setelah disalin:

```
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
```

Izin yang dihasilkan adalah sebagai berikut:

```
-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
```

Contoh berikut menunjukkan cara menentukan `'permissions'` bagian dengan penambahan `acls` dan `context` instruksi. Contoh ini hanya berlaku untuk Amazon Linux, Ubuntu Server, dan instans 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 Bagian 'kait'

Konten di `'hooks'` bagian AppSpec file bervariasi, tergantung pada platform komputasi untuk penerapan Anda. `'hooks'`Bagian untuk EC2/On-Premises penerapan berisi pemetaan yang menghubungkan kait peristiwa siklus hidup penerapan ke satu atau beberapa skrip. `'hooks'`Bagian untuk Lambda atau penerapan Amazon ECS menentukan fungsi validasi Lambda untuk dijalankan selama peristiwa siklus hidup penerapan. Jika hook peristiwa tidak ada, tidak ada operasi yang dijalankan untuk acara itu. Bagian ini diperlukan hanya jika Anda menjalankan skrip atau fungsi validasi Lambda sebagai bagian dari penerapan.

Topik

- AppSpec Bagian 'kait' untuk penerapan Amazon ECS
- AppSpec Bagian 'kait' untuk sebuah AWS Penyebaran Lambda
- AppSpec Bagian 'kait' untuk penerapan EC2/On-Premises

AppSpec Bagian 'kait' untuk penerapan Amazon ECS

Topik

- Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS
- Jalankan urutan kait dalam penerapan Amazon ECS.
- Struktur bagian 'kait'
- Contoh fungsi 'kait' Lambda

Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS

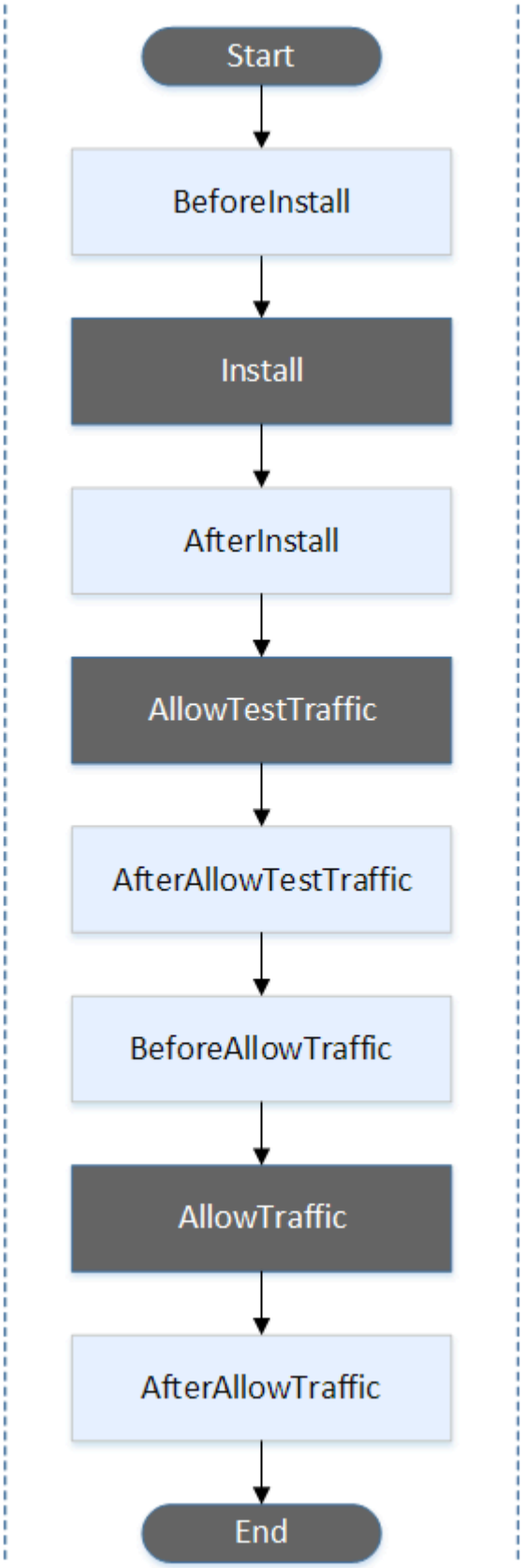
Hook AWS Lambda adalah salah satu fungsi Lambda yang ditentukan dengan string pada baris baru setelah nama peristiwa siklus hidup. Setiap hook dieksekusi satu kali per penerapan. Berikut ini adalah deskripsi peristiwa siklus hidup tempat Anda dapat menjalankan hook selama penerapan Amazon ECS.

- `BeforeInstall`— Gunakan untuk menjalankan tugas sebelum set tugas pengganti dibuat. Satu kelompok target dikaitkan dengan set tugas asli. Jika pendengar uji opsional ditentukan, ini terkait dengan set tugas asli. Rollback tidak dimungkinkan pada saat ini.
- `AfterInstall`— Gunakan untuk menjalankan tugas setelah set tugas pengganti dibuat dan salah satu kelompok target dikaitkan dengannya. Jika pendengar uji opsional ditentukan, ini terkait dengan set tugas asli. Hasil fungsi hook pada peristiwa siklus hidup ini dapat memicu rollback.
- `AfterAllowTestTraffic`— Gunakan untuk menjalankan tugas setelah pendengar pengujian menyajikan lalu lintas ke set tugas pengganti. Hasil fungsi hook pada titik ini dapat memicu rollback.
- `BeforeAllowTraffic`— Gunakan untuk menjalankan tugas setelah kelompok target kedua dikaitkan dengan set tugas pengganti, tetapi sebelum lalu lintas digeser ke set tugas pengganti. Hasil fungsi hook pada peristiwa siklus hidup ini dapat memicu rollback.
- `AfterAllowTraffic`— Gunakan untuk menjalankan tugas setelah kelompok target kedua melayani lalu lintas ke set tugas pengganti. Hasil fungsi hook pada peristiwa siklus hidup ini dapat memicu rollback.

Untuk informasi selengkapnya, lihat Apa yang terjadi selama penyebaran Amazon ECS dan Tutorial: Menyebarkan layanan Amazon ECS dengan uji validasi.

Jalankan urutan kait dalam penerapan Amazon ECS.

Dalam penerapan Amazon ECS, kait peristiwa berjalan dalam urutan berikut:


Start
BeforeInstall
Install
AfterInstall
AllowTestTraffic
AfterAllowTestTraffic
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
End

> **Note**
>
> Peristiwa Mulai, Instal TestTrafficAllowTraffic,,, dan Akhir dalam penerapan tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini.

Struktur bagian 'kait'

Berikut ini adalah contoh struktur `'hooks'` bagian tersebut.

Menggunakan YAMAL:

```
Hooks:
  - BeforeInstall: "BeforeInstallHookFunctionName"
  - AfterInstall: "AfterInstallHookFunctionName"
  - AfterAllowTestTraffic: "AfterAllowTestTrafficHookFunctionName"
  - BeforeAllowTraffic: "BeforeAllowTrafficHookFunctionName"
  - AfterAllowTraffic: "AfterAllowTrafficHookFunctionName"
```

Menggunakan JSON:

```
"Hooks": [
  {
   "BeforeInstall": "BeforeInstallHookFunctionName"
  },
  {
   "AfterInstall": "AfterInstallHookFunctionName"
  },
  {
   "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
  },
  {
   "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
  },
  {
   "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
  }
 ]
}
```

Contoh fungsi 'kait' Lambda

Gunakan `'hooks'` bagian untuk menentukan fungsi Lambda yang CodeDeploy dapat memanggil untuk memvalidasi penerapan Amazon ECS. Anda dapat menggunakan fungsi yang sama atau yang berbeda untuk peristiwa siklus hidup `BeforeInstall` `AfterInstallAfterAllowTestTraffic`,`BeforeAllowTraffic`,, dan `AfterAllowTraffic` penerapan. Setelah menyelesaikan tes validasi, fungsi `AfterAllowTraffic` Lambda memanggil CodeDeploy kembali dan memberikan hasil dari atau. `Succeeded` `Failed`

> **Important**
>
> Penerapan dianggap gagal jika tidak diberitahukan oleh CodeDeploy fungsi validasi Lambda dalam waktu satu jam.

Sebelum menjalankan fungsi hook Lambda, server harus diberi tahu tentang ID penerapan dan ID eksekusi kait peristiwa siklus hidup menggunakan perintah. `putLifecycleEventHookExecutionStatus`

Berikut ini adalah contoh fungsi kait Lambda yang ditulis dalam. 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 Bagian 'kait' untuk sebuah AWS Penyebaran Lambda

Topik

- Daftar kait peristiwa siklus hidup untuk AWS Penyebaran Lambda
- Jalankan urutan kait dalam penerapan versi fungsi Lambda
- Struktur bagian 'kait'
- Contoh fungsi 'kait' Lambda

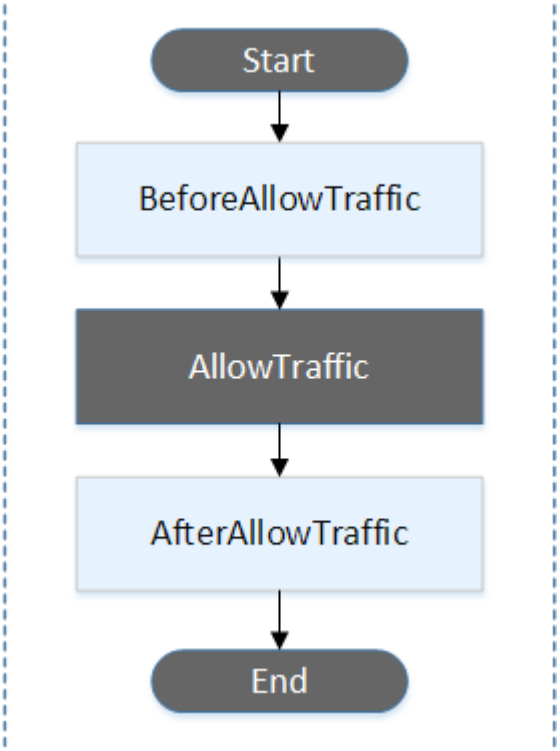
Daftar kait peristiwa siklus hidup untuk AWS Penyebaran Lambda

Hook AWS Lambda adalah salah satu fungsi Lambda yang ditentukan dengan string pada baris baru setelah nama peristiwa siklus hidup. Setiap hook dieksekusi satu kali per penerapan. Berikut adalah deskripsi kait yang tersedia untuk digunakan dalam file Anda AppSpec .

- BeforeAllowTraffic— Gunakan untuk menjalankan tugas sebelum lalu lintas dialihkan ke versi fungsi Lambda yang digunakan.
- AfterAllowTraffic— Gunakan untuk menjalankan tugas setelah semua lalu lintas dialihkan ke versi fungsi Lambda yang digunakan.

Jalankan urutan kait dalam penerapan versi fungsi Lambda

Dalam penerapan versi fungsi Lambda tanpa server, kait peristiwa berjalan dalam urutan berikut:



> **Note**
>
> Peristiwa Mulai AllowTraffic,, dan Akhir dalam penerapan tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini.

Struktur bagian 'kait'

Berikut ini adalah contoh struktur bagian 'kait'.

Menggunakan YAMAL:

```
hooks:
   - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
   - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

Menggunakan JSON:

```
"hooks": [{
    "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
    },
    {
```

```
    "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

Contoh fungsi 'kait' Lambda

Gunakan bagian 'kait' untuk menentukan fungsi Lambda CodeDeploy yang dapat memanggil untuk memvalidasi penerapan Lambda. Anda dapat menggunakan fungsi yang sama atau yang berbeda untuk peristiwa siklus hidup `BeforeAllowTraffic` dan `AfterAllowTraffic` penerapan. Setelah menyelesaikan tes validasi, fungsi validasi Lambda memanggil CodeDeploy kembali dan memberikan hasil dari atau. `Succeeded` `Failed`

> **Important**
>
> Penerapan dianggap gagal jika tidak diberitahukan oleh CodeDeploy fungsi validasi Lambda dalam waktu satu jam.

Sebelum menjalankan fungsi hook Lambda, server harus diberi tahu tentang ID penerapan dan ID eksekusi kait peristiwa siklus hidup menggunakan perintah. `putLifecycleEventHookExecutionStatus`

Berikut ini adalah contoh fungsi kait Lambda yang ditulis dalam. 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 Bagian 'kait' untuk penerapan EC2/On-Premises

Topik

- Daftar kait acara siklus hidup
- Ketersediaan kait acara siklus hidup
- Jalankan urutan kait dalam penerapan
- Struktur bagian 'kait'
- Mereferensikan file dalam skrip hook Anda
- Ketersediaan variabel lingkungan untuk kait
- Contoh kait

Daftar kait acara siklus hidup

Hook EC2/On-Premises penerapan dijalankan sekali per penerapan ke sebuah instance. Anda dapat menentukan satu atau beberapa skrip untuk dijalankan di hook. Setiap hook untuk acara siklus hidup ditentukan dengan string pada baris terpisah. Berikut adalah deskripsi kait yang tersedia untuk digunakan dalam file Anda AppSpec .

Untuk informasi tentang kait peristiwa siklus hidup mana yang valid untuk jenis penerapan dan rollback, lihat. Ketersediaan kait acara siklus hidup

- `ApplicationStop`— Peristiwa siklus hidup penerapan ini terjadi bahkan sebelum revisi aplikasi diunduh. Anda dapat menentukan skrip untuk acara ini untuk menghentikan aplikasi dengan anggun atau menghapus paket yang saat ini diinstal sebagai persiapan untuk penerapan. AppSpec File dan skrip yang digunakan untuk peristiwa siklus hidup penerapan ini berasal dari revisi aplikasi yang berhasil diterapkan sebelumnya.

  > **Note**
  >
  > AppSpec File tidak ada pada instance sebelum Anda menerapkannya. Untuk alasan ini, `ApplicationStop` hook tidak berjalan saat pertama kali Anda menerapkan ke instance. Anda dapat menggunakan `ApplicationStop` hook untuk kedua kalinya Anda menyebarkan ke sebuah instance.

  Untuk menentukan lokasi revisi aplikasi terakhir yang berhasil diterapkan, CodeDeploy agen mencari lokasi yang tercantum dalam file. *`deployment-group-id`*`_last_successful_install` File ini terletak di:

  `/opt/codedeploy-agent/deployment-root/deployment-instructions`folder di Amazon Linux, Server Ubuntu, dan instans RHEL Amazon EC2.

  `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`folder pada instans Windows Server Amazon EC2.

  Untuk memecahkan masalah penerapan yang gagal selama peristiwa siklus hidup `ApplicationStop` penerapan, lihat. Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan

- `DownloadBundle`— Selama peristiwa siklus hidup penerapan ini, CodeDeploy agen menyalin file revisi aplikasi ke lokasi sementara:

  `/opt/codedeploy-agent/deployment-root/`*`deployment-group-id`*`/`*`deployment-id`*`/deployment-archive`folder di Amazon Linux, Server Ubuntu, dan instans RHEL Amazon EC2.

  `C:\ProgramData\Amazon\CodeDeploy\`*`deployment-group-id`*`\`*`deployment-id`*`\deployment-archive`folder pada instans Windows Server Amazon EC2.

  Acara ini dicadangkan untuk CodeDeploy agen dan tidak dapat digunakan untuk menjalankan skrip.

Untuk memecahkan masalah penerapan yang gagal selama peristiwa siklus hidup `DownloadBundle` penerapan, lihat. Memecahkan masalah peristiwa siklus hidup DownloadBundle penerapan yang gagal dengan UnknownError: tidak dibuka untuk dibaca

- `BeforeInstall`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk tugas prainstal, seperti mendekripsi file dan membuat cadangan dari versi saat ini.
- `Install`— Selama peristiwa siklus hidup penerapan ini, CodeDeploy agen menyalin file revisi dari lokasi sementara ke folder tujuan akhir. Acara ini dicadangkan untuk CodeDeploy agen dan tidak dapat digunakan untuk menjalankan skrip.
- `AfterInstall`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk tugas-tugas seperti mengonfigurasi aplikasi Anda atau mengubah izin file.
- `ApplicationStart`— Anda biasanya menggunakan peristiwa siklus hidup penerapan ini untuk memulai ulang layanan yang dihentikan selama. `ApplicationStop`
- `ValidateService`— Ini adalah peristiwa siklus hidup penerapan terakhir. Ini digunakan untuk memverifikasi penyebaran telah selesai dengan sukses.
- `BeforeBlockTraffic`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk menjalankan tugas pada instance sebelum dideregistrasi dari penyeimbang beban.

  Untuk memecahkan masalah penerapan yang gagal selama peristiwa siklus hidup `BeforeBlockTraffic` penerapan, lihat. Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan
- `BlockTraffic`— Selama peristiwa siklus hidup penerapan ini, lalu lintas internet diblokir dari mengakses instance yang saat ini melayani lalu lintas. Acara ini dicadangkan untuk CodeDeploy agen dan tidak dapat digunakan untuk menjalankan skrip.
- `AfterBlockTraffic`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk menjalankan tugas pada instance setelah dideregistrasi dari penyeimbang beban masing-masing.

  Untuk memecahkan masalah penerapan yang gagal selama peristiwa siklus hidup `AfterBlockTraffic` penerapan, lihat. Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan
- `BeforeAllowTraffic`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk menjalankan tugas pada instance sebelum terdaftar dengan penyeimbang beban.
- `AllowTraffic`— Selama peristiwa siklus hidup penerapan ini, lalu lintas internet diizinkan untuk mengakses instance setelah penerapan. Acara ini dicadangkan untuk CodeDeploy agen dan tidak dapat digunakan untuk menjalankan skrip.

- `AfterAllowTraffic`— Anda dapat menggunakan peristiwa siklus hidup penerapan ini untuk menjalankan tugas pada instance setelah terdaftar dengan penyeimbang beban.

Ketersediaan kait acara siklus hidup

Tabel berikut mencantumkan kait peristiwa siklus hidup yang tersedia untuk setiap skenario penerapan dan rollback.

Nama acara siklus hidup	Penerapan peluncuran Auto Scaling[1]	Penerapan penghentian Penskalaan Otomatis[1]	In-place penyebaran[2]	Blue/green penyebaran: Contoh asli	Blue/green penyebaran: Contoh pengganti an	Blue/green rollback penerapan: Contoh asli	Blue/green rollback penerapan: Contoh pengganti an
ApplicationStop	✓	✓	✓		✓		
DownloadBundle[3]	✓		✓		✓		
BeforeInstall	✓		✓		✓		
Instal [3]	✓		✓		✓		
AfterInstall	✓		✓		✓		
ApplicationStart	✓		✓		✓		
ValidateService	✓		✓		✓		
BeforeBlockTraffic		✓	✓	✓			✓

Nama acara siklus hidup	Penerapan peluncuran Auto Scaling[1]	Penerapan penghentian Penskalaan Otomatis[1]	In-place penyebaran[2]	Blue/green penyebaran: Contoh asli	Blue/green penyebaran: Contoh pengganti an	Blue/green rollback penerapan: Contoh asli	Blue/green rollback penerapan: Contoh pengganti an
BlockTraffic[3]		✓	✓	✓			✓
AfterBlockTraffic		✓	✓	✓			✓
BeforeAllowTraffic	✓		✓		✓	✓	
AllowTraffic[3]	✓		✓		✓	✓	
AfterAllowTraffic	✓		✓		✓	✓	

[1] Untuk informasi tentang penerapan Amazon EC2 Auto Scaling, lihat. Cara Kerja Amazon EC2 Auto Scaling CodeDeploy

[2] Juga berlaku untuk rollback penerapan di tempat.

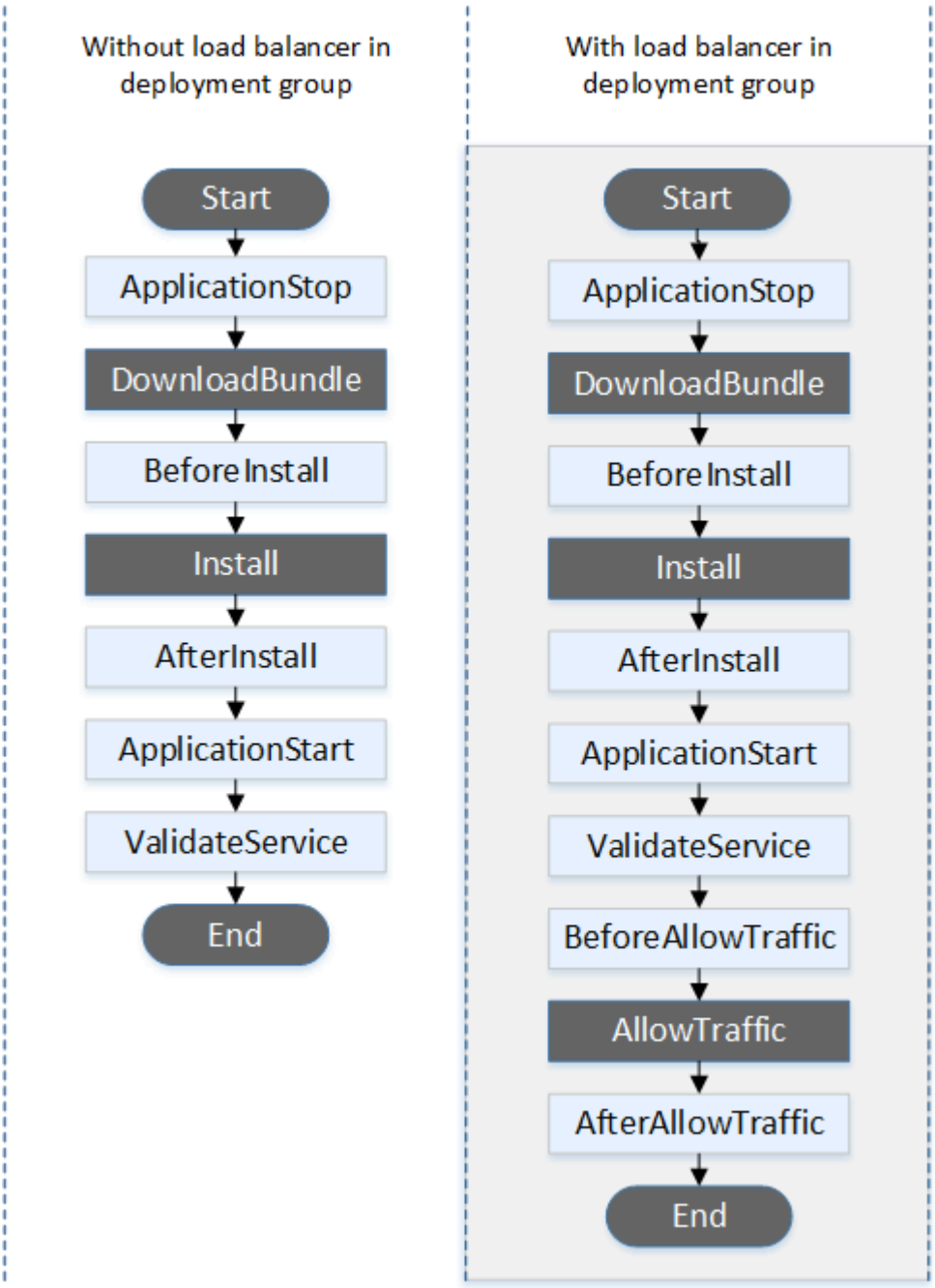
[3] Dicadangkan untuk CodeDeploy operasi. Tidak dapat digunakan untuk menjalankan skrip.

Jalankan urutan kait dalam penerapan

Penerapan peluncuran Auto Scaling

Selama penerapan peluncuran Auto Scaling, CodeDeploy jalankan event hook dalam urutan berikut.

Untuk informasi selengkapnya tentang penerapan peluncuran Auto Scaling, lihat. Cara Kerja Amazon EC2 Auto Scaling CodeDeploy



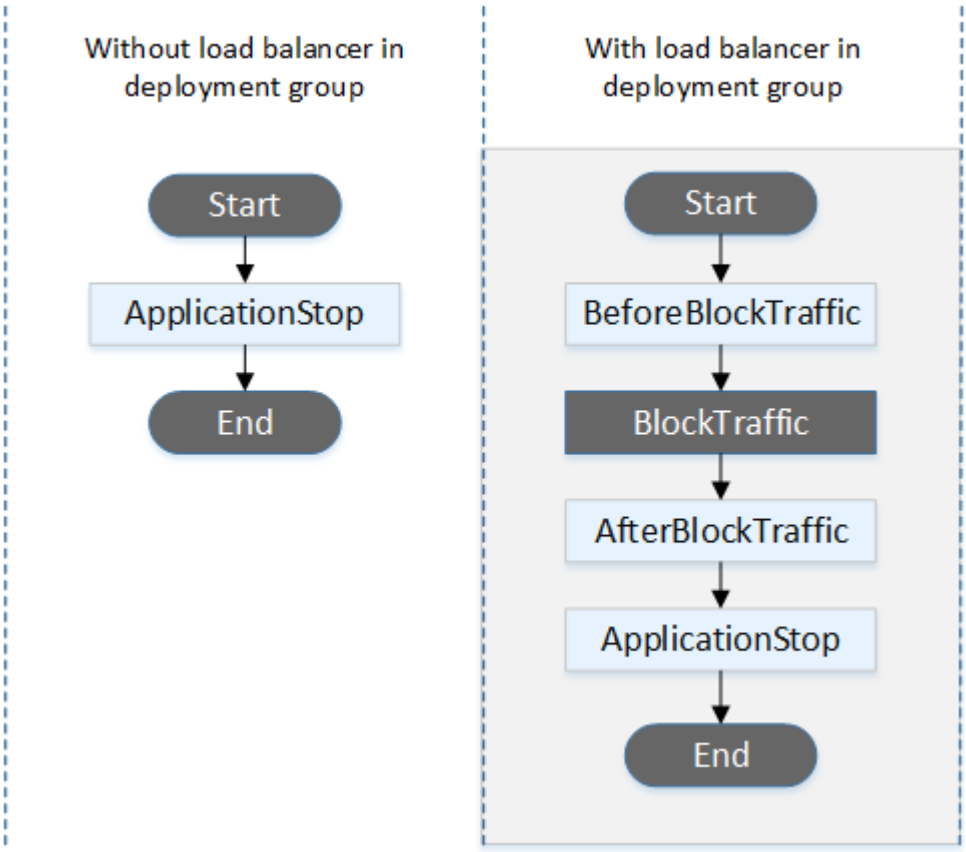
Note

Peristiwa Start DownloadBundle, AllowTraffic, Install, dan End dalam penyebaran tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini. Namun, Anda dapat mengedit `'files'` bagian AppSpec file untuk menentukan apa yang diinstal selama acara Install.

Penerapan penghentian Auto Scaling

Selama penerapan penghentian Auto Scaling, CodeDeploy jalankan event hook dalam urutan berikut.

Untuk informasi selengkapnya tentang penerapan penghentian Auto Scaling, lihat. Mengaktifkan penerapan penghentian selama peristiwa penskalaan Auto Scaling

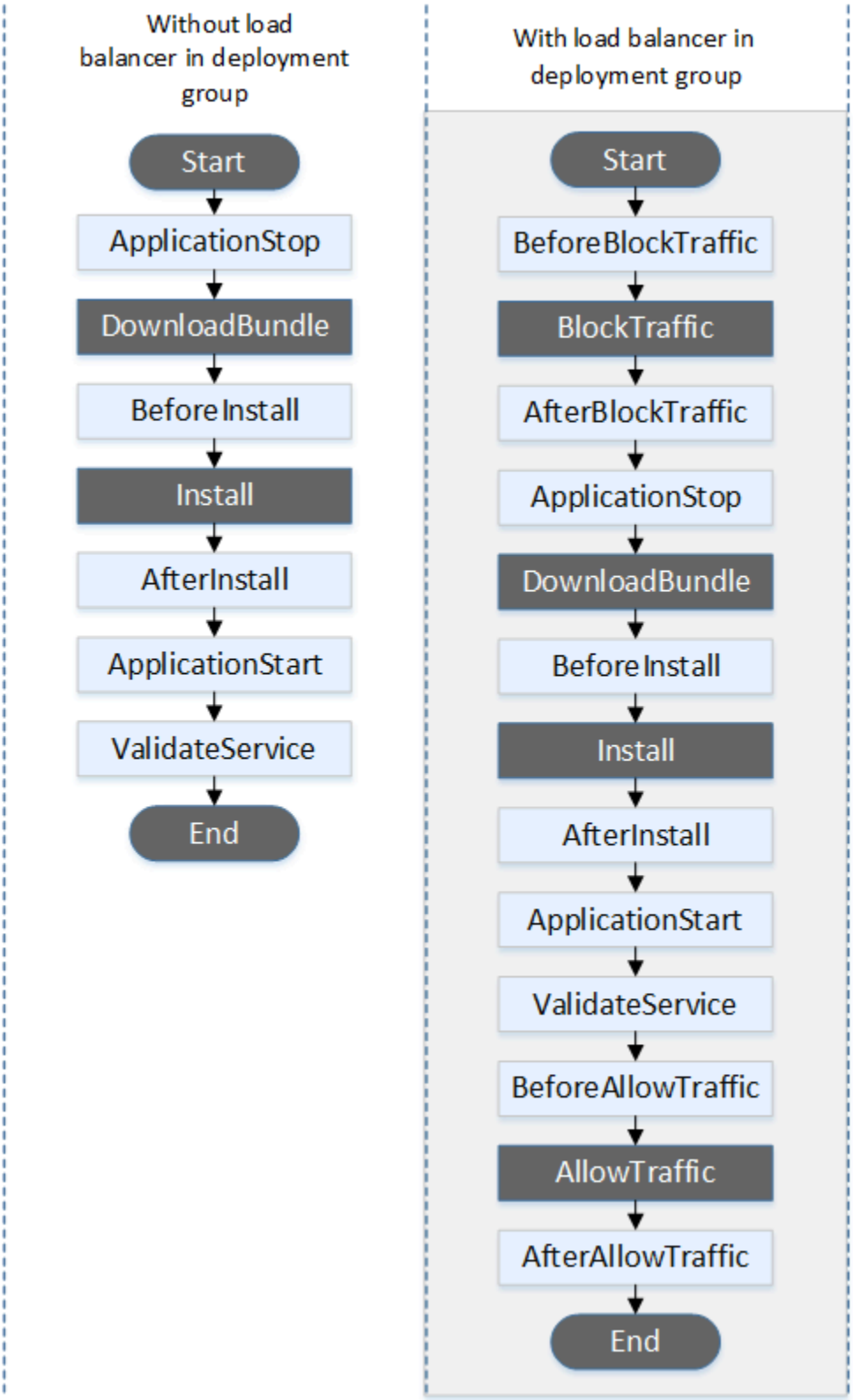


> **Note**
>
> Peristiwa Mulai BlockTraffic,, dan Akhir dalam penerapan tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini.

In-place penyebaran

Dalam penerapan di tempat, termasuk rollback penerapan di tempat, kait peristiwa dijalankan dalam urutan berikut:

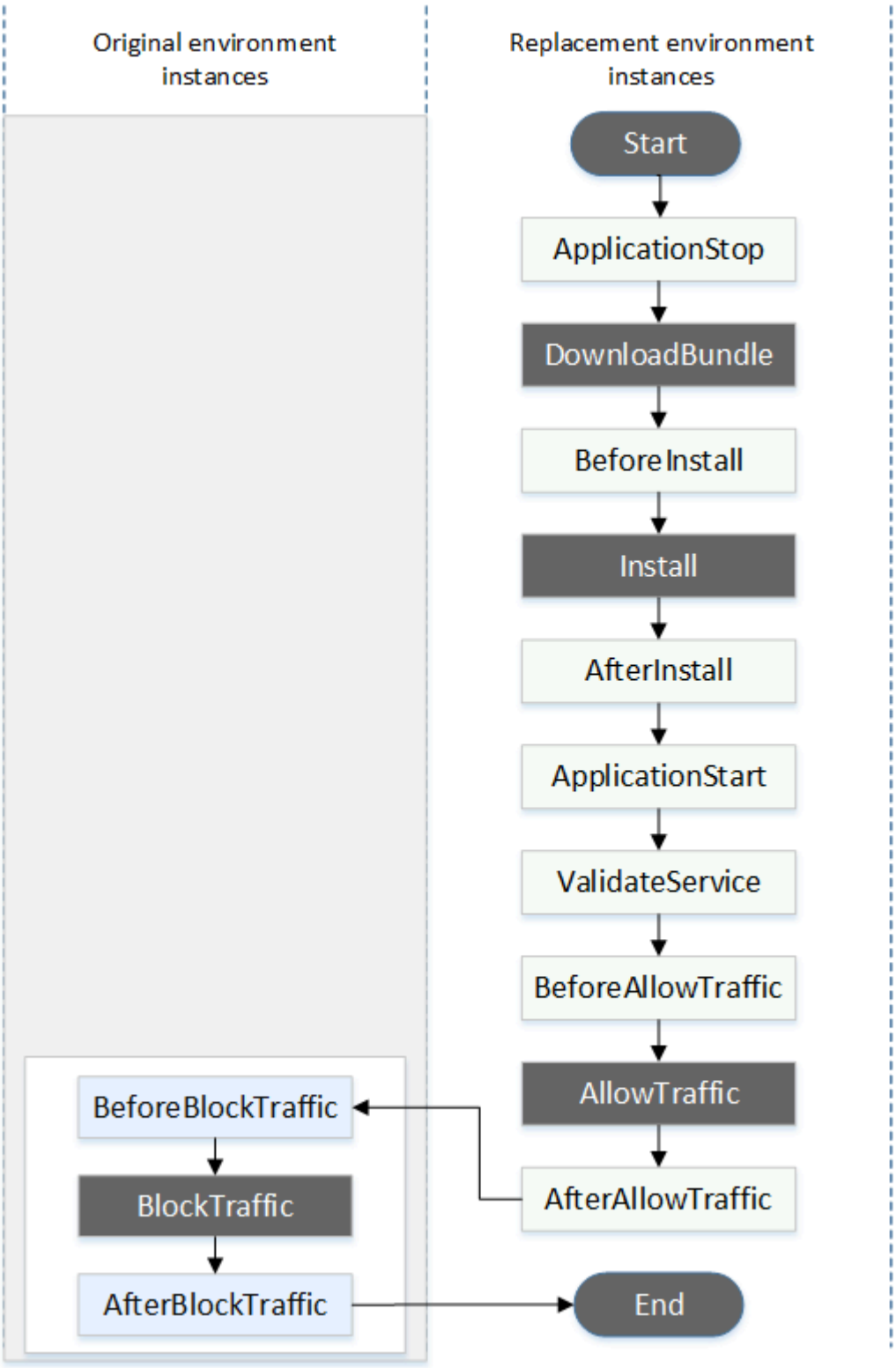
> **Note**
>
> Untuk penerapan di tempat, enam kait yang terkait dengan pemblokiran dan mengizinkan lalu lintas hanya berlaku jika Anda menentukan Classic Load Balancer, Application Load Balancer, atau Network Load Balancer dari Elastic Load Balancing dalam grup penyebaran.


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**
>
> Peristiwa Mulai DownloadBundle,, Instal, dan Akhiri dalam penerapan tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini. Namun, Anda dapat mengedit `'files'` bagian AppSpec file untuk menentukan apa yang diinstal selama acara Install.

Blue/green penyebaran

Dalam blue/green penerapan, event hook dijalankan dalam urutan sebagai berikut:


Original environment instances
Replacement environment instances
Start
ApplicationStop
DownloadBundle
BeforeInstall
Install
AfterInstall
ApplicationStart
ValidateService
BeforeAllowTraffic
AllowTraffic
AfterAllowTraffic
BeforeBlockTraffic
BlockTraffic
AfterBlockTraffic
End

> **Note**
>
> Peristiwa Mulai DownloadBundle,, Instal BlockTrafficAllowTraffic,,, dan Akhir dalam penerapan tidak dapat ditulis, itulah sebabnya mereka muncul dalam warna abu-abu dalam diagram ini. Namun, Anda dapat mengedit bagian 'file' dari AppSpec file untuk menentukan apa yang diinstal selama acara Install.

Struktur bagian 'kait'

'`hooks`' Bagian ini memiliki struktur sebagai berikut:

```
hooks:
   deployment-lifecycle-event-name:
     - location: script-location
       timeout: timeout-in-seconds
       runas: user-name
```

Anda dapat menyertakan elemen-elemen berikut dalam entri hook setelah nama peristiwa siklus hidup penerapan:

lokasi

Wajib. Lokasi dalam bundel file skrip untuk revisi. Lokasi skrip yang Anda tentukan di `hooks` bagian ini relatif terhadap root bundel revisi aplikasi. Untuk informasi selengkapnya, lihat Rencanakan revisi untuk CodeDeploy.

batas waktu

Tidak wajib. Jumlah detik untuk memungkinkan skrip dijalankan sebelum dianggap gagal. Defaultnya adalah 3600 detik (1 jam).

> **Note**
>
> 3600 detik (1 jam) adalah jumlah waktu maksimum yang diizinkan untuk eksekusi skrip untuk setiap peristiwa siklus hidup penerapan. Jika skrip melebihi batas ini, penerapan berhenti dan penerapan ke instance gagal. Pastikan jumlah detik yang ditentukan dalam batas waktu untuk semua skrip di setiap peristiwa siklus hidup penerapan tidak melebihi batas ini.

runa

Tidak wajib. Pengguna untuk meniru saat menjalankan skrip. Secara default, ini adalah CodeDeploy agen yang berjalan pada instance. CodeDeploy tidak menyimpan kata sandi, sehingga pengguna tidak dapat ditiru jika pengguna runas membutuhkan kata sandi. Elemen ini hanya berlaku untuk instance Amazon Linux dan Ubuntu Server.

Mereferensikan file dalam skrip hook Anda

Jika Anda menghubungkan skrip ke peristiwa CodeDeploy siklus hidup seperti yang dijelaskan dalamAppSpec Bagian 'kait', dan Anda ingin mereferensikan file (misalnya,`helper.sh`) dalam skrip Anda, maka Anda harus menentukan menggunakan: `helper.sh`

- (Disarankan) Jalur absolut. Lihat Menggunakan jalur absolut.
- Jalan relatif. Lihat Menggunakan jalur relatif.

Menggunakan jalur absolut

Untuk mereferensikan file menggunakan jalur absolutnya, Anda dapat:

- Tentukan jalur absolut di `files` bagian AppSpec file, di `destination` properti. Kemudian, tentukan jalur absolut yang sama di skrip hook Anda. Untuk informasi selengkapnya, lihat AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan).
- Tentukan jalur absolut dinamis dalam skrip hook Anda. Untuk informasi selengkapnya, lihat Lokasi arsip penerapan.

Lokasi arsip penyebaran

Selama peristiwa DownloadBundlesiklus hidup, CodeDeploy agen mengekstrak revisi untuk penerapan ke direktori yang memiliki format berikut:

root-directory/*deployment-group-id*/*deployment-id*/deployment-archive

*root-directory*Bagian jalur selalu diatur ke default yang ditunjukkan pada tabel berikut, atau dikendalikan oleh pengaturan `:root_dir` konfigurasi. Untuk informasi selengkapnya tentang pengaturan konfigurasi, lihatCodeDeploy referensi konfigurasi agen.

Platform agen	Direktori root default
Linux — semua distribusi rpm	`/opt/codedeploy-agent/deployment-root`
Server Ubuntu — semua distribusi deb	`/opt/codedeploy-agent/deployment-root`
Windows Server	`%ProgramData%\Amazon\CodeDeploy`

Dari skrip hook Anda, Anda dapat mengakses arsip penyebaran saat ini menggunakan jalur direktori root dan variabel `DEPLOYMENT_ID` dan `DEPLOYMENT_GROUP_ID` lingkungan. Untuk informasi lebih lanjut tentang variabel yang dapat Anda gunakan, lihatKetersediaan variabel lingkungan untuk kait.

Misalnya, berikut adalah bagaimana Anda dapat mengakses `data.json` file yang berada di root revisi Anda di 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)
```

Sebagai contoh lain, berikut adalah bagaimana Anda dapat mengakses `data.json` file yang berada di root revisi Anda menggunakan Powershell di 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)
```

Menggunakan jalur relatif

Untuk mereferensikan file menggunakan jalur relatifnya, Anda harus mengetahui direktori kerja CodeDeploy agen. Jalur file relatif terhadap direktori ini.

Tabel berikut menunjukkan direktori kerja untuk setiap platform CodeDeploy agen yang didukung.

Platform agen	Metode manajemen proses	Direktori kerja untuk skrip peristiwa siklus hidup
Linux — semua distribusi rpm	systemd (default)	`/`
	init.d - Pelajari lebih lanjut	`/opt/codedeploy-agent`
Server Ubuntu — semua distribusi debian	all	`/opt/codedeploy-agent`
Windows Server	tidak berlaku	`C:\Windows\System32`

Ketersediaan variabel lingkungan untuk kait

Selama setiap peristiwa siklus hidup penerapan, skrip hook dapat mengakses variabel lingkungan berikut:

APPLICATION_NAME

Nama aplikasi di dalamnya CodeDeploy adalah bagian dari penyebaran saat ini (misalnya,`WordPress_App`).

DEPLOYMENT_ID

ID CodeDeploy telah ditetapkan untuk penyebaran saat ini (misalnya,`d-AB1CDEF23`).

DEPLOYMENT_GROUP_NAME

Nama grup penyebaran di dalamnya CodeDeploy adalah bagian dari penerapan saat ini (misalnya,`WordPress_DepGroup`).

DEPLOYMENT_GROUP_ID

ID grup penyebaran di dalamnya CodeDeploy adalah bagian dari penerapan saat ini (misalnya,`b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE`).

LIFECYCLE_EVENT

Nama peristiwa siklus hidup penerapan saat ini (misalnya,). `AfterInstall`

Variabel lingkungan ini bersifat lokal untuk setiap peristiwa siklus hidup penerapan.

Ada variabel lingkungan tambahan yang tersedia untuk mengaitkan skrip tergantung pada sumber bundel penerapan:

Bundel dari Amazon S3

- BUNDLE_BUCKET

  Nama bucket Amazon S3 dari mana bundel penerapan diunduh (misalnya,). `my-s3-bucket`
- BUNDLE_KEY

  Kunci objek untuk bundel yang diunduh dalam bucket Amazon S3 (misalnya,`WordPress_App.zip`).
- BUNDLE_VERSION

  Versi objek untuk bundel (misalnya,`3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo`). Variabel ini hanya disetel jika bucket Amazon S3 mengaktifkan versi objek.
- BUNDLE_ETAG

  Objek etag untuk bundel (misalnya,`b10a8db164e0754105b7a99be72e3fe5-4`).

Bundel dari GitHub

- BUNDLE_COMMIT

  Hash komit SHA256 dari bundel yang dihasilkan oleh Git (misalnya,`d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26`).

Skrip berikut mengubah port mendengarkan pada server HTTP Apache menjadi 9090, bukan 80 jika nilai DEPLOYMENT_GROUP_NAME sama dengan. `Staging` Skrip ini harus dipanggil selama peristiwa siklus hidup `BeforeInstall` penerapan:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
```

```
then
    sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

Contoh skrip berikut mengubah tingkat verbositas pesan yang direkam dalam log kesalahannya dari peringatan menjadi debug jika nilai variabel lingkungan DEPLOYMENT_GROUP_NAME sama dengan. `Staging` Skrip ini harus dipanggil selama peristiwa siklus hidup `BeforeInstall` penerapan:

```
if [ "$DEPLOYMENT_GROUP_NAME" == "Staging" ]
then
    sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

Contoh skrip berikut menggantikan teks di halaman web yang ditentukan dengan teks yang menampilkan nilai variabel lingkungan ini. Skrip ini harus dipanggil selama peristiwa siklus hidup `AfterInstall` penerapan:

```
#!/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()
```

Contoh kait

Berikut adalah contoh entri kait yang menentukan dua kait untuk acara siklus hidup: `AfterInstall`

```
hooks:
```

```
   AfterInstall:
     - location: Scripts/RunResourceTests.sh
       timeout: 180
     - location: Scripts/PostDeploy.sh
       timeout: 180
```

`Scripts/RunResourceTests.sh`Skrip berjalan selama `AfterInstall` tahap proses penyebaran. Penerapan tidak berhasil jika skrip membutuhkan waktu lebih dari 180 detik (3 menit) untuk dijalankan.

Lokasi skrip yang Anda tentukan di bagian 'kait' relatif terhadap root bundel revisi aplikasi. Dalam contoh sebelumnya, sebuah file bernama `RunResourceTests.sh` dalam direktori bernama. `Scripts` `Scripts`Direktori berada di tingkat root bundel. Untuk informasi selengkapnya, lihat Rencanakan revisi untuk CodeDeploy.

AppSpec Contoh berkas

Topik ini menyediakan contoh AppSpec file untuk AWS Lambda dan penerapan. EC2/On-Premises

Topik

- AppSpec Contoh file untuk penyebaran Amazon ECS
- AppSpec Contoh file untuk AWS Penyebaran Lambda
- AppSpec Contoh file untuk EC2/On-Premises penerapan

AppSpec Contoh file untuk penyebaran Amazon ECS

Berikut adalah contoh AppSpec file yang ditulis dalam YAMG untuk menerapkan layanan 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"
```

Berikut adalah versi dari contoh sebelumnya yang ditulis dalam 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"
    }
  ]
}
```

Berikut adalah urutan peristiwa selama penerapan:

1. Sebelum aplikasi Amazon ECS yang diperbarui diinstal pada set tugas pengganti, fungsi `LambdaFunctionToValidateBeforeInstall` Lambda disebut berjalan.
2. Setelah aplikasi Amazon ECS yang diperbarui diinstal pada set tugas pengganti, tetapi sebelum menerima lalu lintas apa pun, fungsi `LambdaFunctionToValidateAfterInstall` Lambda disebut berjalan.
3. Setelah aplikasi Amazon ECS pada set tugas pengganti mulai menerima lalu lintas dari pendengar pengujian, fungsi Lambda yang disebut berjalan. `LambdaFunctionToValidateAfterTestTrafficStarts` Fungsi ini kemungkinan menjalankan tes validasi untuk menentukan apakah penerapan berlanjut. Jika Anda tidak menentukan listener pengujian dalam grup penerapan Anda, hook ini akan diabaikan.
4. Setelah tes validasi apa pun di `AfterAllowTestTraffic` hook selesai, dan sebelum lalu lintas produksi disajikan ke aplikasi Amazon ECS yang diperbarui, fungsi Lambda disebut berjalan. `LambdaFunctionToValidateBeforeAllowingProductionTraffic`
5. Setelah lalu lintas produksi disajikan ke aplikasi Amazon ECS yang diperbarui pada set tugas pengganti, fungsi `LambdaFunctionToValidateAfterAllowingProductionTraffic` Lambda disebut berjalan.

Fungsi Lambda yang berjalan selama hook apa pun dapat melakukan tes validasi atau mengumpulkan metrik lalu lintas.

AppSpec Contoh file untuk AWS Penyebaran Lambda

Berikut adalah contoh AppSpec file yang ditulis dalam YAMG untuk menerapkan versi fungsi Lambda.

```
version: 0.0
Resources:
  - myLambdaFunction:
      Type: AWS::Lambda::Function
      Properties:
        Name: "myLambdaFunction"
        Alias: "myLambdaFunctionAlias"
        CurrentVersion: "1"
        TargetVersion: "2"
Hooks:
  - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
  - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Berikut adalah versi dari contoh sebelumnya yang ditulis dalam JSON.

```
{
  "version": 0.0,
  "Resources": [{
   "myLambdaFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
     "Name": "myLambdaFunction",
     "Alias": "myLambdaFunctionAlias",
     "CurrentVersion": "1",
     "TargetVersion": "2"
    }
   }
  }],
  "Hooks": [{
    "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
      },
      {
    "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
   }
  ]
 }
```

Berikut adalah urutan peristiwa selama penerapan:

1. Sebelum mengalihkan lalu lintas dari versi 1 dari fungsi Lambda yang `myLambdaFunction` dipanggil ke versi 2, jalankan fungsi Lambda yang `LambdaFunctionToValidateBeforeTrafficShift` disebut yang memvalidasi penerapan siap untuk memulai pergeseran lalu lintas.
2. Jika `LambdaFunctionToValidateBeforeTrafficShift` mengembalikan kode keluar 0 (sukses), mulailah mengalihkan lalu lintas ke versi 2. `myLambdaFunction` Konfigurasi penyebaran untuk penyebaran ini menentukan tingkat di mana lalu lintas digeser.
3. Setelah pergeseran lalu lintas dari versi 1 dari fungsi Lambda yang `myLambdaFunction` dipanggil ke versi 2 selesai, jalankan fungsi Lambda yang `LambdaFunctionToValidateAfterTrafficShift` disebut yang memvalidasi penerapan berhasil diselesaikan.

AppSpec Contoh file untuk EC2/On-Premises penerapan

Berikut adalah contoh AppSpec file untuk penyebaran di tempat ke Amazon Linux, Ubuntu Server, atau instance RHEL.

> **Note**
>
> Penerapan ke instance Windows Server tidak mendukung elemen. `runas` Jika Anda menyebarkan ke instance Windows Server, jangan sertakan dalam file Anda AppSpec .

```
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
```

Untuk instance Windows Server, ubah `os: linux` keos: `windows`. Juga, Anda harus sepenuhnya memenuhi syarat `destination` jalur (misalnya, `c:\temp\webapps\Config` danc:\temp `\webapps\myApp`). Jangan sertakan `runas` elemen.

Berikut adalah urutan peristiwa selama penerapan:

1. Jalankan skrip yang terletak di`Scripts/UnzipResourceBundle.sh`.

2. Jika skrip sebelumnya mengembalikan kode keluar 0 (sukses), jalankan skrip yang terletak di`Scripts/UnzipDataBundle.sh`.
3. Salin file dari jalur `Config/config.txt` ke jalur`/webapps/Config/config.txt`.
4. Salin semua file dalam direktori secara rekursif ke `source` `/webapps/myApp` direktori.
5. Jalankan skrip yang terletak di `Scripts/RunResourceTests.sh` dengan batas waktu 180 detik (3 menit).
6. Jalankan skrip yang terletak di `Scripts/RunFunctionalTests.sh` dengan batas waktu 3600 detik (1 jam).
7. Jalankan skrip yang terletak di `Scripts/MonitorService.sh` sebagai pengguna `codedeploy` dengan batas waktu 3600 detik (1 jam).

AppSpec Spasi file

Berikut ini adalah format yang benar untuk spasi AppSpec file. Angka-angka dalam tanda kurung siku menunjukkan jumlah spasi yang harus terjadi di antara item. Misalnya, `[4]` berarti menyisipkan empat spasi di antara item. CodeDeploy menimbulkan kesalahan yang mungkin sulit untuk di-debug jika lokasi dan jumlah spasi dalam AppSpec file tidak benar.

```
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
```

Berikut adalah contoh AppSpec file dengan spasi yang benar:

```
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
```

Untuk informasi selengkapnya tentang spasi, lihat spesifikasi YAMB.

Validasi AppSpec File dan lokasi file Anda

Sintaks file

Anda dapat menggunakan skrip AppSpec Asisten yang AWS disediakan untuk memvalidasi konten file AppSpec . Anda dapat menemukan skrip bersama dengan template AppSpec file di GitHub.

Anda juga dapat menggunakan alat berbasis browser seperti YAML lint atau online YAMB parser untuk membantu Anda memeriksa sintaks YAMB Anda.

Lokasi berkas

Untuk memverifikasi bahwa Anda telah menempatkan AppSpec file Anda di direktori root struktur direktori konten sumber aplikasi, jalankan salah satu perintah berikut:

Pada instance Linux, macOS, atau Unix lokal:

```
ls path/to/root/directory/appspec.yml
```

Jika AppSpec file tidak terletak di sana, kesalahan “Tidak ada file atau direktori seperti itu” ditampilkan.

Pada instance Windows lokal:

```
dir path\to\root\directory\appspec.yml
```

Jika AppSpec file tidak terletak di sana, kesalahan “File Tidak Ditemukan” ditampilkan.

CodeDeploy referensi konfigurasi agen

Ketika CodeDeploy agen diinstal, file konfigurasi ditempatkan pada instance. File konfigurasi ini menentukan jalur direktori dan pengaturan lain CodeDeploy untuk digunakan saat berinteraksi dengan instance. Anda dapat mengubah beberapa opsi konfigurasi dalam file.

Untuk Amazon Linux, Ubuntu Server, dan Red Hat Enterprise Linux (RHEL) instance, file konfigurasi diberi nama. `codedeployagent.yml` Itu ditempatkan di `/etc/codedeploy-agent/conf` direktori.

Untuk instance Windows Server, file konfigurasi diberi nama`conf.yml`. Itu ditempatkan di `C:\ProgramData\Amazon\CodeDeploy` direktori.

Pengaturan konfigurasi meliputi:

:log_aws_wire:	Setel ke CodeDeploy agen **`true`** untuk menangkap log kawat dari Amazon S3 dan menuliskannya ke file log di lokasi yang ditunjukkan oleh setelan:log_dir:.

Untuk versi 2.0.x dan yang lebih baru, nama file adalah `codedeploy-agent.aws_wire.log` (dibuat dengan izin 0640).

Untuk versi 1.8.x dan sebelumnya, nama filenya adalah. `codedeploy-agent.wire.log`

> **Warning**
>
> Anda harus menyetel:log_aws_wire: hanya untuk jumlah waktu yang diperlukan `true` untuk menangkap log kawat. File log kawat dapat tumbuh ke ukuran yang sangat besar dengan cepat. Output log kawat mungkin berisi informasi sensitif, termasuk konten teks biasa dari file yang ditransfer ke, atau keluar dari, Amazon S3 saat pengaturan ini disetel ke. `true` Log kawat berisi informasi tentang semua aktivitas Amazon S3 yang terkait dengan AWS akun saat pengaturan ini disetel`true`, bukan hanya aktivitas yang terkait CodeDeploy dengan penerapan.

Pengaturan default-nya adalah `false`.

Pengaturan ini berlaku untuk semua jenis instance. Anda harus menambahkan pengaturan konfigurasi ini ke instance Windows Server untuk dapat menggunakannya.

:log_dir:	Folder pada instance tempat file log yang terkait dengan operasi CodeDeploy agen disimpan. Pengaturan default adalah `'/var/log/aws/codedeploy-agent'` untuk Amazon Linux, Ubuntu Server, dan instans RHEL dan `C:\ProgramData\Amazon\CodeDeploy\log` untuk instance Windows Server.
:pid_dir:	Folder tempat `codedeploy-agent.pid` disimpan. File ini berisi ID proses (PID) CodeDeploy agen. Pengaturan default-nya adalah `'/opt/codedeploy-agent/state/.pid'` . Pengaturan ini hanya berlaku untuk Amazon Linux, Ubuntu Server, dan instans RHEL.
:program_name:	Nama program CodeDeploy agen. Pengaturan default-nya adalah `codedeploy-agent`. Pengaturan ini hanya berlaku untuk Amazon Linux, Ubuntu Server, dan instans RHEL.
:root_dir:	Folder tempat revisi terkait, riwayat penerapan, dan skrip penerapan pada instance disimpan. Pengaturan default adalah `/opt/codedeploy-agent/deployment-root` untuk Amazon Linux, Ubuntu Server, dan instans RHEL dan `C:\ProgramData\Amazon\CodeDeploy` untuk instance Windows Server.

:bertele-tele:	Setel ke `true` CodeDeploy agen untuk mencetak file log pesan debug pada instance. Pengaturan default-nya adalah `false`.
:wait_between_runs:	Interval, dalam hitungan detik, antara polling CodeDeploy agen CodeDeploy untuk penerapan yang tertunda. Untuk versi 2.0.x dan yang lebih baru, pengaturan defaultnya adalah. `30` Untuk versi 1.8.x dan sebelumnya, pengaturan defaultnya adalah. 1
:on_premises_config_file:	Untuk instance lokal, jalur ke lokasi alternati f untuk file konfigurasi bernama `codedeplo y.onpremises.yml` (untuk Ubuntu Server dan RHEL) atau `conf.onpremises.yml` (untuk Windows Server). Secara default, file-file ini disimpan di`/etc/ codedeploy-agent/conf /codedeplo y.onpremises.yml` untuk Ubuntu Server dan RHEL dan`C:\ProgramData\Amazon \CodeDeploy \conf.onpremises.yml` untuk Windows Server. Tersedia dalam versi 1.0.1.686 dan versi agen yang lebih baru. CodeDeploy
:proxy_uri:	(Opsional) Proxy HTTP di mana Anda ingin CodeDeploy agen terhubung AWS untuk CodeDeploy operasi Anda. Gunakan format yang mirip dengan`https://user:passw ord@my.proxy:443/path?query` . Tersedia dalam versi 1.0.1.824 dan versi agen yang lebih baru. CodeDeploy

:max_revisi:	(Opsional) Jumlah revisi aplikasi untuk grup penyebaran yang Anda ingin CodeDeploy agen arsipkan. Setiap revisi yang melebihi jumlah yang ditentukan akan dihapus. Masukkan bilangan bulat positif. Jika tidak ada nilai yang ditentukan, CodeDeploy akan mempertahankan lima revisi terbaru selain revisi yang saat ini diterapkan. Didukung dalam versi 1.0.1.966 dan versi agen yang lebih baru. CodeDeploy
:enable_auth_policy:	(Opsional) Setel ke `true` jika Anda ingin menggunakan otorisasi IAM untuk mengonfig urasi kontrol akses dan membatasi izin peran IAM atau pengguna yang digunakan CodeDeploy Agen. UntukGunakan CodeDeploy dengan Amazon Virtual Private Cloud, nilai ini harus`true`. Pengaturan default-nya adalah `false`.
:disable_imds_v1:	Pengaturan ini tersedia dengan CodeDeploy agen 1.7.0 dan yang lebih baru. Setel `true` untuk menonaktifkan fallback ke IMDSv1 saat kesalahan IMDSv2 terjadi. Default ke `false` (aktifkan fallback).
:disable_core_dumps:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menekan dump inti. Ditetapka n `false` untuk investigasi kecelakaan. Pengaturan default-nya adalah `true`.

:enable_command_port:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk mengaktifkan antarmuka manajemen lokal khusus loopback (127.0.0.1) untuk menanyakan status agen dan menyuntikkan perintah secara lokal. Pengaturan default-nya adalah `false`.
:enable_archive_gunakan kembali:	Untuk versi 2.1.0 dan yang lebih baru. Atur `false` untuk selalu mengunduh revisi aplikasi, bahkan dalam `RESTART` penerapan , di mana revisi sudah ada di instance. Ketika pengaturan ini`true`, agen menginstal dari salinan yang sudah ada pada instance, dan mengunduh revisi jika salinan itu hilang atau tidak lengkap. Pengaturan default-nya adalah `true`. Untuk informasi lebih lanjut tentang perilaku ini, lihatRiwayat versi CodeDeploy agen.
:reject_symlinks_in_bundle:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menolak bundel yang berisi symlink atau hardlink. Pengaturan default-nya adalah `false`.
:reject_path_traversal_in_bundle:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menolak entri arsip yang berisi `../` jalur traversal atau jalur absolut, dan bidang AppSpec sumber yang keluar dari root bundel. Pengaturan default-nya adalah `false`.

:reject_unsafe_permissions_in_bundle:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menolak bundel yang berisi SUID/SGID file atau mode AppSpec izin yang disetel. SUID/SGID Pengaturan default-nya adalah `false`.
:reject_unconfined_selinux_in_bundle:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menolak bundel yang berisi file dengan tipe MAC-disabling SELinux. Pengaturan default-nya adalah `false`.
:reject_symlink_permission_targets:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menolak `permissions` entri AppSpec file yang targetnya adalah tautan simbolis. Secara default, agen mengikuti tautan saat menerapkan perubahan kepemilikan, mode, ACL, dan SELinux. Pengaturan default-nya adalah `false`.
:ignore_ownership_in_bundle:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk mengatur ulang kepemilik an file yang diekstrak dari `tar` dan `tgz` bundel `root` alih-alih menerapkan ID pengguna dan ID grup yang direkam dalam arsip. File dalam `zip` bundel dimiliki oleh `root` terlepas dari pengaturan ini. Pengaturan default-nya adalah `false`.

:restrict_agent_dir_permissions:	Untuk versi 2.0.x dan yang lebih baru. Instance Linux saja. Setel `true` untuk membatasi direktori dan file di bawah root instalasi agen ke akses hanya root alih-alih mode yang dapat dibaca dunia yang digunakan secara default. Pengaturan default-nya adalah `false`.
:restrict_log_dir_permissions:	Untuk versi 2.0.x dan yang lebih baru. Instance Linux saja. Setel `true` untuk membatasi direktori log agen, yang berisi log agen dan log pembaru, ke akses khusus root alih-alih mode yang dapat dibaca dunia yang digunakan secara default. Pengaturan default-nya adalah `false`.
:restrict_hook_env_to_allowlist:	Untuk versi 2.0.x dan yang lebih baru. Setel `true` untuk menjalankan kait peristiwa siklus hidup dengan lingkungan minimal yang hanya berisi variabel shell dasar dan variabel penerapan CodeDeploy. Secara default, kait mewarisi lingkungan penuh dari proses agen. Pengaturan default-nya adalah `false`.

:strip_loader_env_in_hooks:	Untuk versi 2.0.x dan yang lebih baru. Instans Linux, dan instance macOS dengan versi 2.1.2 dan yang lebih baru. Setel `true` untuk menghapus`LD_PRELOAD` ,`LD_LIBRARY_PATH` , dan `LD_AUDIT` variabel dari lingkungan kait peristiwa siklus hidup. Di macOS, pengaturan menghapus ,`DYLD_INSERT_LIBRARIES` ,,,`DYLD_LIBRARY_PATH` , `DYLD_FRAMEWORK_PATH` `DYLD_FALLBACK_LIBRARY_PATH` `DYLD_FALLBACK_FRAMEWORK_PATH` `DYLD_VERSIONED_LIBRARY_PATH` , dan `DYLD_VERSIONED_FRAMEWORK_PATH` variabel sebagai gantinya. Pengaturan default-nya adalah `false`.
:disable_powershell_profile_in_hooks:	Untuk versi 2.0.x dan yang lebih baru. Instans Windows Server saja. Setel `true` untuk menjalankan kait peristiwa `.ps1` siklus hidup dengan opsi dan`-NoProfile` `.-NonInteractive` Secara default, kait ini memuat PowerShell profil pengguna yang dijalankan agen. Pengaturan default-nya adalah `false`.
:archive_max_extraction_size:	Untuk versi 2.0.x dan yang lebih baru. Menetapkan tutup pada ukuran bundel yang tidak terkompresi yang dinyatakan. Jika konten yang diekstraksi melebihi nilai ini, penerapan gagal. Defaultnya adalah no cap (unlimited).

:wait_after_error:	Jumlah detik agen menunggu setelah mengalami kesalahan sebelum melanjutkan operasi normal. Pengaturan default-nya adalah `30`.
:http_read_timeout:	Batas waktu baca HTTP dalam hitungan detik untuk panggilan API layanan. Pengaturan default-nya adalah `80`.
:kill_agent_max_wait_time_seconds:	Jumlah maksimum detik agen menunggu shutdown yang anggun sebelum penghentian paksa. Untuk versi 2.0.x dan yang lebih baru, nilai ini dibatasi pada 24 jam (86400 detik). Pengaturan default adalah `7200` (2 jam).
:enable_deployments_log:	Setel `true` untuk menulis log penerapan agregat (). `codedeploy-agent-deployments.log` Pengaturan default-nya adalah `true`.
:use_fips_mode:	Setel `true` untuk menggunakan FIPS-compliant titik akhir. Hanya berlaku untuk Wilayah AS. Pengaturan default-nya adalah `false`.
:deploy_control_endpoint:	(Opsional) URL titik akhir bidang kontrol khusus untuk layanan. CodeDeploy Defaultnya adalah no override (agen menyelesaikan titik akhir dari Region).

:s3_endpoint_override:	(Opsional) URL endpoint Amazon S3 khusus untuk unduhan bundel. Defaultnya adalah tidak ada penggantian (agen menyelesaikan titik akhir Amazon S3 dari Wilayah).
:s3_download_max_concurrency:	Untuk versi 2.1.2 dan yang lebih baru. Jumlah permintaan rentang HTTP paralel yang digunakan agen untuk mengunduh revisi aplikasi dari Amazon S3. Setel ke nilai yang lebih besar daripada 1 mengunduh revisi besar dalam rentang paralel. Dengan`1`, agen mengunduh revisi dengan satu `GetObject` permintaan. `64`Nilai-nilai di atas diperlakukan sebagai`64`, dan `0` diperlakukan sebagai`1`. Pengaturan default-nya adalah `1`.
:s3_download_chunk_size:	Untuk versi 2.1.2 dan yang lebih baru. Ukuran setiap permintaan rentang saat `:s3_download_max_concurrency:` lebih besar dari`1`. Masukkan sejumlah byte atau string ukuran seperti `"16MiB"` atau`"16MB"`. Revisi ukuran ini atau lebih kecil diunduh dengan satu permintaan. Nilai di bawah 64 KiB diperlakukan sebagai 64 KiB. Pengaturan default-nya adalah `"8MiB"`.

:ongoing_deployment_tracking:	Nama subdirektori yang digunakan untuk status pelacakan penerapan. Pengaturan default-nya adalah `ongoing-deployment` .

Topik terkait

[Bekerja dengan CodeDeploy agen]

[Mengelola operasi CodeDeploy agen]

CloudFormation template untuk CodeDeploy referensi

Bagian ini memperkenalkan CloudFormation sumber daya, transformasi, dan kait yang dirancang untuk bekerja dengan CodeDeploy penerapan. Untuk panduan membuat pembaruan tumpukan yang dikelola oleh CloudFormation hook for CodeDeploy, lihat [Buat blue/green penyebaran Amazon ECS melalui CloudFormation]

> **Note**
>
> CloudFormation kait adalah bagian dari CloudFormation komponen untuk AWS dan berbeda dari kait peristiwa CodeDeploy siklus hidup.

Selain metode lain yang tersedia untuk Anda CodeDeploy, Anda dapat menggunakan CloudFormation templat untuk melakukan tugas-tugas berikut:

- Buat aplikasi.
- Buat grup penyebaran dan tentukan revisi target.
- Buat konfigurasi penerapan.
- Buat instans Amazon EC2.

CloudFormation adalah layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda menggunakan templat. CloudFormation Template adalah file teks yang formatnya sesuai dengan standar JSON. Anda membuat template yang menjelaskan semua sumber AWS daya yang

Anda inginkan, dan CloudFormation mengurus penyediaan dan konfigurasi sumber daya tersebut untuk Anda.

Untuk informasi lebih lanjut, lihat Apa itu AWS CloudFormation? dan Bekerja dengan AWS CloudFormation template di Panduan AWS CloudFormation Pengguna.

Jika Anda berencana untuk menggunakan CloudFormation template yang kompatibel dengan CodeDeploy di organisasi Anda, sebagai administrator, Anda harus memberikan akses ke CloudFormation dan ke AWS layanan dan tindakan yang CloudFormation bergantung pada. Untuk memberikan izin untuk membuat aplikasi, grup penerapan, dan konfigurasi penerapan, tambahkan kebijakan berikut ke kumpulan izin pengguna yang akan bekerja dengan: CloudFormation

JSON

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

Untuk informasi selengkapnya tentang kebijakan, lihat topik berikut:

- Untuk melihat kebijakan yang harus ditambahkan ke kumpulan izin pengguna yang akan membuat instans Amazon EC2, lihat. Buat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat)
- Untuk informasi tentang menambahkan kebijakan ke set izin, lihat Membuat set izin di Panduan Pengguna IAM.
- Untuk mempelajari cara membatasi pengguna pada serangkaian CodeDeploy tindakan dan sumber daya terbatas, lihatAWS kebijakan terkelola (standar) untuk CodeDeploy.

Tabel berikut menunjukkan tindakan yang dapat dilakukan CloudFormation template atas nama Anda dan menyertakan tautan ke informasi selengkapnya tentang jenis AWS sumber daya dan jenis propertinya yang dapat Anda tambahkan ke CloudFormation templat.

Tindakan	CloudFormation referensi	Jenis referensi
Buat CodeDeploy aplikasi.	AWS::CodeDeploy: :aplikasi	CloudFormation sumber daya
Buat dan tentukan detail untuk grup penyebaran yang akan digunakan untuk menyebarkan revisi aplikasi Anda. [1]	AWS::CodeDeploy::DeploymentGroup	CloudFormation sumber daya
Buat seperangkat aturan penerapan, kondisi keberhasilan penerapan, dan kondisi kegagalan penerapan yang CodeDeploy akan digunakan selama penerapan.	AWS::CodeDeploy::DeploymentConfig	CloudFormation sumber daya
Buat instans Amazon EC2. [2]	AWS: :EC2: :contoh	CloudFormation sumber daya
Gunakan CloudFormation `AWS::CodeDeployBlueGreen` transform dan `AWS::CodeDeploy::BlueGreen` hook untuk mengelola pembaruan tumpukan, membuat sumber daya, dan mengalihkan lalu lintas untuk CodeDeploy blue/green penerapan. [3]	AWS::CodeDeployBlueGreen	`AWS::CodeDeployBlueGreen` Transformasi adalah makro yang dihosting oleh CloudFormation
	AWS::CodeDeploy::BlueGreen	`AWS::CodeDeploy::BlueGreen` Hook disusun sebagai Hook sumber daya di CloudFormation. Hook mencakup parameter yang menggantikan CodeDeploy AppSpec file Anda dengan menunjuk ke kait peristiwa CodeDeploy siklus hidup yang ditentukan.

Tindakan	CloudFormation referensi	Jenis referensi
[1] Jika Anda menentukan versi revisi aplikasi yang ingin Anda gunakan sebagai bagian dari grup penyebaran, revisi target Anda akan diterapkan segera setelah proses penyediaan selesai. Untuk informasi selengkapnya tentang konfigurasi templat, lihat revisi CodeDeploy DeploymentGroup penerapan S3Lokasi dan GitHubLocationrevisi CodeDeploy DeploymentGroup penerapan di Panduan Pengguna.AWS CloudFormation [2] Kami menyediakan templat yang dapat Anda gunakan untuk membuat instans Amazon EC2 di Wilayah yang CodeDeploy didukung. Untuk informasi selengkapnya tentang menggunakan templat ini, lihatBuat instans Amazon EC2 untuk (CodeDeploy CloudFormation templat). [3] Hanya blue/green penerapan Amazon ECS yang didukung oleh konfigurasi penerapan ini. Untuk informasi selengkapnya tentang konfigurasi penerapan untuk blue/green penerapan Amazon ECS, lihat. CloudFormationKonfigurasi penerapan untuk CloudFormation blue/green penerapan (Amazon ECS) Untuk informasi selengkapnya tentang blue/green penerapan Amazon ECS CloudFormation dan cara melihat penerapan Anda, lihat. CodeDeploy Buat blue/green penyebaran Amazon ECS melalui CloudFormation		

Gunakan CodeDeploy dengan Amazon Virtual Private Cloud

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan VPC. CodeDeploy Anda dapat menggunakan koneksi ini untuk memungkinkan CodeDeploy untuk berkomunikasi dengan sumber daya Anda di VPC Anda tanpa melalui internet publik.

Amazon VPC adalah AWS layanan yang dapat Anda gunakan untuk meluncurkan AWS sumber daya di jaringan virtual yang Anda tentukan. Dengan VPC, Anda memiliki kendali terhadap pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan pintu masuk jaringan. Dengan titik

akhir VPC, perutean antara VPC dan AWS layanan ditangani oleh AWS jaringan, dan Anda dapat menggunakan kebijakan IAM untuk mengontrol akses ke sumber daya layanan.

Untuk menghubungkan VPC Anda CodeDeploy, Anda menentukan titik akhir VPC antarmuka untuk. CodeDeploy Endpoint antarmuka adalah elastic network interface dengan alamat IP pribadi yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan ke layanan yang didukung AWS. Endpoint menyediakan konektivitas yang andal dan dapat diskalakan CodeDeploy tanpa memerlukan gateway internet, instance terjemahan alamat jaringan (NAT), atau koneksi VPN. Untuk informasi selengkapnya, silakan lihat Apa itu Amazon VPC dalam Panduan Pengguna Amazon VPC.

Endpoint VPC antarmuka didukung oleh AWS PrivateLink, sebuah AWS teknologi yang memungkinkan komunikasi pribadi antara AWS layanan menggunakan antarmuka jaringan elastis dengan alamat IP pribadi. Untuk informasi selengkapnya, lihat AWS PrivateLink.

Langkah-langkah berikut ditujukan untuk para pengguna Amazon VPC. Untuk informasi selengkapnya, silakan lihat Getting Started di Panduan Pengguna Amazon VPC.

Ketersediaan

CodeDeploy memiliki dua titik akhir VPC: satu untuk operasi CodeDeploy agen, dan satu untuk CodeDeploy operasi API. Tabel di bawah ini menunjukkan AWS Wilayah yang didukung untuk setiap titik akhir.

Untuk informasi selengkapnya dan untuk titik akhir FIPS, lihat AWS CodeDeploy titik akhir dan kuota.

Nama wilayah	Kode Wilayah	Titik akhir agen	Titik akhir API	Wilayah yang sesuai dengan FIPS?
US East (Northern Virginia)	us-east-1	Ya	Ya	Ya
US East (Ohio)	us-east-2	Ya	Ya	Ya
US West (N. California)	us-west-1	Ya	Ya	Ya
US West (Oregon)	us-west-2	Ya	Ya	Ya

Nama wilayah	Kode Wilayah	Titik akhir agen	Titik akhir API	Wilayah yang sesuai dengan FIPS?
Africa (Cape Town)	af-south-1	Ya	Ya	
Asia Pasifik (Hong Kong)	ap-east-1	Ya	Ya	
Asia Pasifik (Hyderabad)	ap-south-2	Ya	Ya	
Asia Pasifik (Jakarta)	ap-southeast-3	Ya	Ya	
Asia Pacific (Melbourne)	ap-southeast-4	Ya	Ya	
Asia Pasifik (Malaysia)	ap-southeast-5	Ya	Ya	
Asia Pasifik (Selandia Baru)	ap-tenggara 6	Ya	Ya	
Asia Pasifik (Thailand)	ap-tenggara 7	Ya	Ya	
Asia Pasifik (Taipei)	ap-timur-2	Ya	Ya	
Asia Pacific (Mumbai)	ap-south-1	Ya	Ya	
Asia Pacific (Osaka)	ap-northeast-3	Ya	Ya	
Asia Pacific (Seoul)	ap-northeast-2	Ya	Ya	

Nama wilayah	Kode Wilayah	Titik akhir agen	Titik akhir API	Wilayah yang sesuai dengan FIPS?
Asia Pacific (Singapore)	ap-southeast-1	Ya	Ya	
Asia Pacific (Sydney)	ap-southeast-2	Ya	Ya	
Asia Pacific (Tokyo)	ap-northeast-1	Ya	Ya	
Canada (Central)	ca-sentral-1	Ya	Ya	
China (Beijing)	cn-north-1	Ya	Tidak	
China (Ningxia)	cn-northwest-1	Tidak	Tidak	
Eropa (Frankfurt)	eu-central-1	Ya	Ya	
Eropa (Irlandia)	eu-west-1	Ya	Ya	
Eropa (London)	eu-west-2	Ya	Ya	
Europe (Milan)	eu-south-1	Ya	Ya	
Europe (Paris)	eu-west-3	Ya	Ya	
Eropa (Spanyol)	eu-south-2	Ya	Ya	
Eropa (Stockholm)	eu-north-1	Ya	Ya	
Europe (Zurich)	eu-central-2	Ya	Ya	
Israel (Tel Aviv)	il-central-1	Ya	Ya	
Timur Tengah (Bahrain)	me-south-1	Ya	Ya	

Nama wilayah	Kode Wilayah	Titik akhir agen	Titik akhir API	Wilayah yang sesuai dengan FIPS?
Timur Tengah (UAE)	me-central-1	Ya	Ya	
Meksiko (Tengah)	mx-pusat-1	Ya	Ya	
South America (São Paulo)	sa-east-1	Ya	Ya	
AWS GovCloud (US-East)	us-gov-east-1	Tidak	Tidak	Ya
AWS GovCloud (US-West)	us-gov-west-1	Tidak	Tidak	Ya

Buat titik akhir VPC untuk CodeDeploy

Untuk mulai menggunakan CodeDeploy dengan VPC Anda, buat antarmuka VPC endpoint untuk. CodeDeploy CodeDeploymemerlukan titik akhir terpisah untuk operasi Git agen dan untuk operasi CodeDeploy API. Bergantung pada kebutuhan bisnis Anda, Anda mungkin perlu membuat lebih dari satu VPC endpoint. Saat Anda membuat titik akhir VPC untuk CodeDeploy, pilih AWS Layanan, dan di Nama Layanan, pilih dari opsi berikut:

- com.amazonaws. ***region***.codedeploy: Pilih opsi ini jika Anda ingin membuat titik akhir VPC untuk operasi API. CodeDeploy Misalnya, pilih opsi ini jika pengguna Anda menggunakan AWS CLI, CodeDeploy API, atau AWS SDK untuk berinteraksi dengan CodeDeploy operasi seperti`CreateApplication`,`GetDeployment`, dan`ListDeploymentGroups`.
- com.amazonaws. ***region***.codedeploy-fips: Pilih opsi ini jika Anda ingin membuat titik akhir VPC untuk operasi API untuk titik akhir FIPS. CodeDeploy
- com.amazonaws. ***region***.codedeploy-commands-secure: Pilih opsi ini jika Anda ingin membuat titik akhir VPC untuk operasi agen. CodeDeploy Anda juga perlu mengatur `:enable_auth_policy:` ke `true` dalam file konfigurasi agen Anda dan melampirkan izin yang diperlukan. Untuk informasi selengkapnya, lihat Konfigurasikan CodeDeploy izin agen dan IAM.

Jika Anda menggunakan penerapan Lambda atau ECS, Anda hanya perlu membuat titik akhir VPC untuk com.amazonaws. ***region***.codedeploy. Pelanggan yang menggunakan penerapan Amazon EC2 akan membutuhkan titik akhir VPC untuk kedua com.amazonaws. *region*.codedeploy dan com.amazonaws. ***region***.codedeploy-commands-secure.

Konfigurasikan CodeDeploy izin agen dan IAM

Untuk menggunakan titik akhir Amazon VPC CodeDeploy, Anda harus menetapkan nilai `:enable_auth_policy:` ke `true` dalam file konfigurasi agen yang terletak di EC2 atau instans lokal. Untuk informasi selengkapnya tentang file konfigurasi agen, lihatCodeDeploy referensi konfigurasi agen.

Anda juga harus menambahkan izin IAM berikut ke profil instans Amazon EC2 (jika Anda menggunakan instans Amazon EC2) atau pengguna atau peran IAM (jika Anda menggunakan instans lokal).

```
{
  "Statement": [
    {
      "Action": [
        "codedeploy-commands-secure:GetDeploymentSpecification",
        "codedeploy-commands-secure:PollHostCommand",
        "codedeploy-commands-secure:PutHostCommandAcknowledgement",
        "codedeploy-commands-secure:PutHostCommandComplete"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Pembuatan Titik Akhir Antarmuka dalam Panduan Pengguna Amazon VPC.

CodeDeploy referensi kit sumber daya

Banyak file yang CodeDeploy diandalkan disimpan dalam ember Amazon AWS Region-specific S3 yang tersedia untuk umum. File-file ini termasuk file instalasi untuk CodeDeploy agen, template, dan file aplikasi sampel. Kami menyebut koleksi file ini CodeDeploy Resource Kit.

Topik

- Nama bucket kit sumber daya menurut Wilayah
- Isi kit sumber daya
- Menampilkan daftar file kit sumber daya
- Unduh file kit sumber daya

Nama bucket kit sumber daya menurut Wilayah

Tabel ini mencantumkan nama-nama *bucket-name* penggantian yang diperlukan untuk beberapa prosedur dalam panduan ini. Ini adalah nama-nama bucket Amazon S3 yang berisi file CodeDeploy Resource Kit.

> **Note**
>
> Untuk mengakses bucket Amazon S3 di Wilayah Asia Pasifik (Hong Kong), Anda harus mengaktifkan wilayah tersebut di akun Anda AWS . Untuk informasi selengkapnya, lihat Mengelola AWS Wilayah.

Nama wilayah	***Bucket-name*** penggantian	Pengidentifikasi wilayah
AS Timur (Virginia Utara)	aws-codedeploy-us-east-1	us-east-1
AS Timur (Ohio)	aws-codedeploy-us-east-2	us-east-2
AS Barat (California Utara)	aws-codedeploy-us-west-1	us-west-1
AS Barat (Oregon)	aws-codedeploy-us-west-2	as-barat-2
Africa (Cape Town)	aws-codedeploy-af-selatan-1	af-south-1
Asia Pasifik (Hong Kong)	aws-codedeploy-ap-east-1	ap-east-1
Asia Pasifik (Hyderabad)	aws-codedeploy-ap-selatan-2	ap-south-2
Asia Pasifik (Jakarta)	aws-codedeploy-ap-southeast-3	ap-southeast-3

Nama wilayah	*Bucket-name* penggantian	Pengidentifikasi wilayah
Asia Pacific (Melbourne)	aws-codedeploy-ap-southeast-4	ap-southeast-4
Asia Pasifik (Malaysia)	aws-codedeploy-ap-southeast-5	ap-southeast-5
Asia Pasifik (Selandia Baru)	aws-codedeploy-ap-southeast-6	ap-tenggara 6
Asia Pasifik (Thailand)	aws-codedeploy-ap-southeast-7	ap-tenggara 7
Asia Pasifik (Taipei)	aws-codedeploy-ap-east-2	ap-timur-2
Asia Pasifik (Mumbai)	aws-codedeploy-ap-south-1	ap-south-1
Asia Pasifik (Osaka)	aws-codedeploy-ap-northeast-3	ap-northeast-3
Asia Pasifik (Seoul)	aws-codedeploy-ap-northeast-2	ap-northeast-2
Asia Pasifik (Singapura)	aws-codedeploy-ap-southeast-1	ap-southeast-1
Asia Pasifik (Sydney)	aws-codedeploy-ap-southeast-2	ap-southeast-2
Asia Pasifik (Tokyo)	aws-codedeploy-ap-northeast-1	ap-northeast-1
Kanada (Pusat)	aws-codedeploy-ca-central-1	ca-central-1
Eropa (Frankfurt)	aws-codedeploy-eu-central-1	eu-central-1
Eropa (Irlandia)	aws-codedeploy-eu-west-1	eu-west-1
Eropa (London)	aws-codedeploy-eu-west-2	eu-west-2

Nama wilayah	*Bucket-name* penggantian	Pengidentifikasi wilayah
Europe (Milan)	aws-codedeploy-eu-selatan-1	eu-south-1
Eropa (Paris)	aws-codedeploy-eu-west-3	eu-west-3
Eropa (Spanyol)	aws-codedeploy-eu-selatan-2	eu-south-2
Eropa (Stockholm)	aws-codedeploy-eu-utara-1	eu-north-1
Europe (Zurich)	aws-codedeploy-eu-central-2	eu-central-2
Israel (Tel Aviv)	aws-codedeploy-il-central-1	il-central-1
Timur Tengah (Bahrain)	aws-codedeploy-me-selatan-1	me-south-1
Timur Tengah (UAE)	aws-codedeploy-me-central-1	me-central-1
Meksiko (Tengah)	aws-codedeploy-mx-central-1	mx-pusat-1
Amerika Selatan (Sao Paulo)	aws-codedeploy-sa-east-1	sa-east-1
AWS GovCloud (US-East)	aws-codedeploy-us-gov-east-1	us-gov-east-1
AWS GovCloud (US-West)	aws-codedeploy-us-gov-west-1	us-gov-west-1

Isi kit sumber daya

Tabel berikut mencantumkan file dalam CodeDeploy Resource Kit.

File	Deskripsi
`LATEST_VERSION`	File yang digunakan oleh mekanisme pembaruan seperti Amazon EC2 Systems Manager untuk menentukan versi terbaru agen CodeDeploy .
`VERSION`	Mekanisme pembaruan otomatis telah dihapus di CodeDeploy agen versi 1.1.0 dan file ini

File	Deskripsi
	tidak lagi digunakan. File yang digunakan oleh CodeDeploy agen untuk memperbarui diri mereka sendiri saat mereka berjalan pada instance.
`codedeploy-agent.noarch.rpm`	CodeDeploy Agen untuk Amazon Linux dan Red Hat Enterprise Linux (RHEL). Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`-1.0-0`).
`codedeploy-agent_all.deb`	CodeDeploy Agen untuk Server Ubuntu. Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`_1.0-0`).
`codedeploy-agent.msi`	CodeDeploy Agen untuk Windows Server. Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`-1.0-0`).
`install`	File yang dapat Anda gunakan untuk menginstal CodeDeploy agen dengan lebih mudah. `latestv2/` Awalan menyajikan skrip instalasi versi 2.0.x.
`CodeDeploy_SampleCF_Template.json`	AWS CloudFormation Template yang dapat Anda gunakan untuk meluncurkan dari satu hingga tiga instans Amazon EC2 yang menjalankan Amazon Linux atau Windows Server. Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`-1.0.0`).

File	Deskripsi
`CodeDeploy_SampleCF_ELB_Integration.json`	AWS CloudFormation Template yang dapat Anda gunakan untuk membuat situs web sampel load-balanced yang berjalan di Apache Web Server. Aplikasi dikonfigurasi untuk menjangkau semua Availability Zone di Wilayah tempat Anda membuatnya. Template ini membuat tiga instans Amazon EC2 dan profil instans IAM untuk memberikan instans akses ke sumber daya di Amazon S3, Amazon EC2 Auto Scaling, dan Elastic Load Balancing . CloudFormation Ini juga menciptakan penyeimbang beban dan peran CodeDeploy layanan.
`SampleApp_ELB_Integration.zip`	Contoh revisi aplikasi yang dapat Anda terapkan ke instans Amazon EC2 yang terdaftar ke penyeimbang beban Elastic Load Balancing.
`SampleApp_Linux.zip`	Contoh revisi aplikasi yang dapat Anda terapkan ke instans Amazon EC2 yang menjalankan Amazon Linux atau ke Server Ubuntu atau instans RHEL. Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`-1.0`).
`SampleApp_Windows.zip`	Contoh revisi aplikasi yang dapat Anda terapkan ke instance Windows Server. Mungkin ada beberapa file dengan nama file dasar yang sama, tetapi versi yang berbeda (seperti`-1.0`).

Menampilkan daftar file kit sumber daya

Untuk melihat daftar file, gunakan aws s3 ls perintah untuk Wilayah Anda.

> **Note**
>
> File di setiap bucket dirancang untuk bekerja dengan sumber daya di Wilayah terkait.

- `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`

Unduh file kit sumber daya

Untuk mengunduh file, gunakan aws s3 cp perintah untuk Wilayah Anda.

> **Note**
>
> Pastikan untuk menggunakan periode (`.`) di dekat akhir. Ini mengunduh file ke direktori Anda saat ini.

Misalnya, perintah berikut mengunduh satu file bernama `SampleApp_Linux.zip` dari salah satu folder ember`/samples/latest/`:

- `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
  ```

Untuk mengunduh semua file, gunakan salah satu perintah berikut untuk Wilayah Anda:

- ```
  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 kuota

Tabel berikut menjelaskan kuota di CodeDeploy.

> **Note**
>
> Penerapan EC2/On-Premises di tempat yang dijalankan dalam batas jam bervariasi. Untuk konfigurasi penerapan khusus yang dibuat sebelum Juni 2023, batasnya adalah 8 jam. Untuk konfigurasi penerapan khusus yang dibuat pada Juni 2023 atau lebih baru, batasnya adalah 12 jam. Untuk konfigurasi penerapan yang telah ditentukan sebelumnya, batasnya adalah 12 jam.

Nama	Default	Dapat disesuaikan	Deskripsi
AWS Penerapan Lambda berjalan dalam hitungan jam	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum jam penyebaran AWS Lambda dapat berjalan (48 jam untuk waktu maksimum antara shift lalu lintas pertama dan terakhir ditambah satu jam untuk masing-masing dua kemungkinan kait siklus hidup)
Aplikasi yang terkait per akun per wilayah	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum aplikasi yang terkait dengan AWS akun di satu wilayah
Alarm terkait per grup penyebaran	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum alarm yang terkait dengan grup penyebaran
Grup Auto Scaling dalam grup deployment	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum grup Amazon EC2 Auto Scaling dalam grup penerapan

Nama	Default	Dapat disesu an	Deskripsi
Deployment bersamaan per akun	Setiap Wilayah yang didukung: 1.300	Ya	Jumlah maksimum penerapan bersamaan yang terkait dengan akun. AWS Setiap penerapan ke instans Amazon EC2 yang ditingkatkan dalam grup Penskalaa n Otomatis Amazon EC2 dihitung sebagai penerapan bersamaan tunggal untuk setiap aplikasi yang dikaitkan dengan instans EC2.
Deployment bersamaan per grup deployment	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum penerapan bersamaan ke grup penerapan. Batas ini mencegah penerapan aplikasi yang sama secara bersamaan ke grup penerapan yang sama.
Konfigurasi deployment kustom per akun	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum konfigurasi penerapan kustom yang terkait dengan akun AWS
Grup deployment yang terkait dengan satu aplikasi	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum grup penyebaran yang terkait dengan satu aplikasi

Nama	Default	Dapat disesuaikan	Deskripsi
EC2/On-Premises blue/green penerapan berjalan dalam beberapa jam	Setiap Wilayah yang didukung: 109	Tidak	Jumlah maksimum jam EC2/On-Premises blue/green penerapan dapat dijalankan (48 jam untuk masing-masing dari dua batas di atas ditambah satu jam untuk masing-masing dari 13 kemungkinan peristiwa siklus hidup)
EC2/On-Premises penerapan di tempat berjalan dalam hitungan jam	Setiap Wilayah yang didukung: 12	Tidak	Jumlah jam maksimum penerapan EC2/On-Premises di tempat dapat dijalankan
Pemicu notifikasi peristiwa dalam grup deployment	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum pemicu pemberitahuan peristiwa dalam grup penyebaran
GitHub token koneksi per akun	Setiap Wilayah yang didukung: 25	Tidak	Jumlah maksimum token GitHub koneksi untuk satu AWS akun
Jam antara penyelesaian penerapan dan penghentian instance asli selama penerapan EC2/On-Premises blue/green	Setiap Wilayah yang didukung: 48	Tidak	Jumlah maksimum jam antara penyelesaian penerapan dan penghentian instans asli selama penerapan EC2/On-Premises blue/green

Nama	Default	Dapat disesuaikan	Deskripsi
Jam antara penyebaran revisi dan ketika lalu lintas bergeser ke instance pengganti selama penyebaran EC2/On-Premises blue/green	Setiap Wilayah yang didukung: 48	Tidak	Jumlah maksimum jam antara penyebaran revisi dan ketika lalu lintas bergeser ke instance pengganti selama penyebaran EC2/On-Premises blue/green
Jumlah instans per deployment	us-east-1:2.000 us-east-2:2.000 us-west-2:2.000 eu-central-1:2.000 Masing-masing Wilayah yang didukung lainnya: 1.000	Ya	Jumlah maksimum instance dalam satu penerapan
Menit blue/green penerapan dapat menunggu setelah penerapan berhasil sebelum menghentikan instance dari penerapan asli	Setiap Wilayah yang didukung: 2.800	Tidak	Jumlah menit maksimum blue/green penerapan dapat menunggu setelah penerapan berhasil sebelum menghentikan instance dari penerapan asli

Nama	Default	Dapat disesuaikan	Deskripsi
Menit antara pergeseran lalu lintas pertama dan terakhir selama kenari AWS Lambda atau penyebaran linier	Setiap Wilayah yang didukung: 2,880	Tidak	Jumlah maksimum menit antara pergeseran lalu lintas pertama dan terakhir selama kenari AWS Lambda atau penyebaran linier
Menit hingga deployment gagal jika peristiwa siklus hidup tidak diluncurkan	Setiap Wilayah yang didukung: 5	Tidak	Jumlah menit maksimum hingga penerapan gagal jika peristiwa siklus hidup tidak dimulai setelah (1) penerapan dipicu dengan menggunakan konsol atau perintah AWS CLI create-deployment, atau (2) peristiwa siklus hidup sebelumnya selesai.
Jumlah grup deployment yang dapat dikaitkan dengan layanan Amazon ECS	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum grup penyebaran yang dapat dikaitkan dengan layanan Amazon ECS
Jumlah instance yang dapat diteruskan ke aksi BatchGetOnPremisesInstances API	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum instance yang dapat diteruskan ke aksi BatchGetOnPremises Instances API

Nama	Default	Dapat disesuaikan	Deskripsi
Jumlah instans yang digunakan oleh deployment bersamaan yang sedang berlangsung per akun	us-east-1:3.000 Masing-masing Wilayah yang didukung lainnya: 1.000	Ya	Jumlah maksimum instance yang dapat digunakan oleh penerapan bersamaan yang sedang berlangsung dan terkait dengan satu akun
Jumlah listener untuk rute lalu lintas selama deployment Amazon ECS	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum pendengar untuk rute lalu lintas selama penyebaran Amazon ECS
Detik hingga peristiwa siklus hidup deployment gagal jika tidak selesai	Setiap Wilayah yang didukung: 3,600 Detik	Tidak	Jumlah maksimum detik hingga peristiwa siklus hidup penerapan gagal jika tidak selesai
Ukuran nama grup deployment	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum karakter dalam nama grup penyebaran
Ukuran kunci tag	Setiap Wilayah yang didukung: 128	Tidak	Jumlah maksimum karakter dalam kunci tag
Ukuran nilai tag	Setiap Wilayah yang didukung: 256	Tidak	Jumlah maksimum karakter dalam nilai tag
Tag dalam grup deployment	Setiap Wilayah yang didukung: 10	Tidak	Jumlah maksimum tag dalam grup penyebaran

Nama	Default	Dapat disesuaikan	Deskripsi
Lalu lintas yang dapat digeser dalam satu kenaikan selama penyebaran Lambda AWS	Setiap Wilayah yang didukung: 99	Tidak	Persentase maksimum lalu lintas yang dapat digeser dalam satu kenaikan selama penyebaran Lambda AWS

Pemecahan masalah CodeDeploy

Gunakan topik di bagian ini untuk membantu memecahkan masalah dan kesalahan yang mungkin Anda temui saat menggunakan CodeDeploy.

> **Note**
>
> Anda dapat mengidentifikasi penyebab banyak kegagalan penerapan dengan meninjau file log yang dibuat selama proses penyebaran. Untuk mempermudah, sebaiknya gunakan Amazon CloudWatch Logs untuk memonitor file log secara terpusat alih-alih melihatnya secara instance per instance. Untuk informasi, lihat Monitoring Deployments with Amazon CloudWatch Tools.

Topik

- Pemecahan masalah umum
- Memecahkan masalah EC2/On-Premises penerapan
- Memecahkan masalah penerapan Amazon ECS
- Pemecahan Masalah AWS Masalah penyebaran Lambda
- Memecahkan masalah grup penyebaran
- Memecahkan masalah contoh
- Memecahkan masalah GitHub token
- Memecahkan masalah Amazon EC2 Auto Scaling
- Kode kesalahan untuk AWS CodeDeploy

Pemecahan masalah umum

Topik

- Daftar periksa pemecahan masalah umum
- CodeDeploy sumber daya penerapan hanya didukung di beberapa AWS Wilayah
- Prosedur dalam panduan ini tidak cocok dengan CodeDeploy konsol
- Peran IAM yang diperlukan tidak tersedia

- Menggunakan beberapa editor teks untuk membuat AppSpec file dan skrip shell dapat menyebabkan penerapan gagal
- Menggunakan Finder di macOS untuk menggabungkan revisi aplikasi dapat menyebabkan penerapan gagal

Daftar periksa pemecahan masalah umum

Anda dapat menggunakan daftar periksa berikut untuk memecahkan masalah penerapan yang gagal.

1. Lihat Lihat CodeDeploy detail penerapan dan View Instance Details untuk menentukan mengapa penerapan gagal. Jika Anda tidak dapat menentukan penyebabnya, tinjau item dalam daftar periksa ini.
2. Periksa apakah Anda telah mengonfigurasi instance dengan benar:
   - Apakah instance diluncurkan dengan key pair EC2 ditentukan? Untuk informasi selengkapnya, lihat Pasangan Kunci EC2 di Panduan Pengguna Amazon EC2.
   - Apakah profil instans IAM yang benar dilampirkan ke instance? Untuk informasi selengkapnya, lihat Konfigurasikan instans Amazon EC2 agar berfungsi CodeDeploy dan Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda.
   - Apakah instance itu ditandai? Untuk informasi selengkapnya, lihat Bekerja dengan tag di konsol di Panduan Pengguna Amazon EC2.
   - Apakah CodeDeploy agen diinstal, diperbarui, dan berjalan pada instance? Untuk informasi selengkapnya, lihat Mengelola operasi CodeDeploy agen. Untuk memeriksa versi agen mana yang diinstal, lihatTentukan versi CodeDeploy agen.
3. Periksa pengaturan grup aplikasi dan penyebaran:
   - Untuk memeriksa pengaturan aplikasi Anda, lihatLihat detail aplikasi dengan CodeDeploy.
   - Untuk memeriksa setelan grup penerapan, lihatLihat detail grup penerapan dengan CodeDeploy.
4. Konfirmasikan revisi aplikasi dikonfigurasi dengan benar:
   - Periksa format AppSpec file Anda. Untuk informasi selengkapnya, lihat Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploy dan CodeDeploy AppSpec referensi file.
   - Periksa bucket atau GitHub repositori Amazon S3 Anda untuk memverifikasi revisi aplikasi Anda berada di lokasi yang diharapkan.
   - Tinjau detail revisi CodeDeploy aplikasi Anda untuk memastikan bahwa itu terdaftar dengan benar. Untuk informasi, lihat Lihat detail revisi aplikasi dengan CodeDeploy.

- Jika Anda menerapkan dari Amazon S3, periksa bucket Amazon S3 Anda untuk CodeDeploy memverifikasi izin telah diberikan untuk mengunduh revisi aplikasi. Untuk informasi tentang kebijakan bucket, lihatPrasyarat penyebaran.
- Jika Anda menerapkan dari GitHub, periksa GitHub repositori Anda untuk memverifikasi CodeDeploy telah diberikan izin untuk mengunduh revisi aplikasi. Untuk informasi selengkapnya, lihat Buat penerapan dengan CodeDeploy dan GitHub otentikasi dengan aplikasi di CodeDeploy.

5. Periksa apakah peran layanan dikonfigurasi dengan benar. Untuk informasi, lihat Langkah 2: Buat peran layanan untuk CodeDeploy.
6. Konfirmasikan bahwa Anda mengikuti langkah-langkah Memulai dengan CodeDeploy untuk:
   - Menyediakan pengguna dengan izin yang sesuai.
   - Instal atau tingkatkan dan konfigurasikan file AWS CLI.
   - Buat profil instans IAM dan peran layanan.

   Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk AWS CodeDeploy.
7. Konfirmasikan bahwa Anda menggunakan AWS CLI versi 1.6.1 atau yang lebih baru. Untuk memeriksa versi yang telah Anda instal, hubungiaws --version.

Jika Anda masih tidak dapat memecahkan masalah penerapan yang gagal, tinjau masalah lain dalam topik ini.

CodeDeploy sumber daya penerapan hanya didukung di beberapa AWS Wilayah

Jika Anda tidak melihat atau tidak dapat mengakses aplikasi, grup penyebaran, instance, atau sumber daya penerapan lainnya dari AWS CLI atau CodeDeploy konsol, pastikan Anda mereferensikan salah satu Wilayah yang terdaftar di Wilayah dan AWS titik akhir di. Referensi Umum AWS

Instans EC2 dan grup Amazon EC2 Auto Scaling yang digunakan CodeDeploy dalam penerapan harus diluncurkan dan dibuat di salah satu Wilayah ini. AWS

Jika Anda menggunakan AWS CLI, jalankan aws configure perintah dari file AWS CLI. Kemudian Anda dapat melihat dan mengatur AWS Wilayah default Anda.

Jika Anda menggunakan CodeDeploy konsol, pada bilah navigasi, dari pemilih Wilayah, pilih salah satu AWS Wilayah yang didukung.

> **Important**
>
> Untuk menggunakan layanan di Wilayah Tiongkok (Beijing) atau Wilayah Tiongkok (Ningxia), Anda harus memiliki akun dan kredensional untuk wilayah tersebut. Akun dan kredensil untuk AWS wilayah lain tidak berfungsi untuk Wilayah Beijing dan Ningxia, dan sebaliknya. Informasi tentang beberapa sumber daya untuk Wilayah Tiongkok, CodeDeploy seperti nama bucket Resource Kit CodeDeploy dan prosedur pemasangan agen, tidak termasuk dalam edisi Panduan Pengguna ini. CodeDeploy
> Untuk informasi selengkapnya:
>
> - CodeDeployMemulai dengan AWS di Wilayah Tiongkok (Beijing)
> - CodeDeploy Panduan Pengguna untuk Wilayah Tiongkok (versi bahasa Inggris | Versi bahasa Mandarin)

Prosedur dalam panduan ini tidak cocok dengan CodeDeploy konsol

Prosedur dalam panduan ini ditulis untuk mencerminkan desain konsol baru. Jika Anda menggunakan versi konsol yang lebih lama, banyak konsep dan prosedur dasar dalam panduan ini masih berlaku. Untuk mengakses bantuan di konsol baru, pilih ikon informasi.

Peran IAM yang diperlukan tidak tersedia

Jika Anda mengandalkan profil instans IAM atau peran layanan yang dibuat sebagai bagian dari AWS CloudFormation tumpukan, jika Anda menghapus tumpukan, semua peran IAM juga akan dihapus. Ini mungkin mengapa peran IAM tidak lagi ditampilkan di konsol IAM dan CodeDeploy tidak lagi berfungsi seperti yang diharapkan. Untuk memperbaiki masalah ini, Anda harus secara manual membuat ulang peran IAM yang dihapus.

Menggunakan beberapa editor teks untuk membuat AppSpec file dan skrip shell dapat menyebabkan penerapan gagal

Beberapa editor teks memperkenalkan karakter yang tidak sesuai dan tidak dicetak ke dalam file. Jika Anda menggunakan editor teks untuk membuat atau memodifikasi AppSpec file atau file skrip shell untuk dijalankan di Amazon Linux, Ubuntu Server, atau instans RHEL, maka penerapan apa pun yang bergantung pada file ini mungkin gagal. Saat CodeDeploy menggunakan file-file ini selama penerapan, keberadaan karakter ini dapat menyebabkan kegagalan validasi AppSpec file yang sulit dipecahkan dan kegagalan eksekusi skrip.

Di CodeDeploy konsol, pada halaman detail acara untuk penerapan, pilih Lihat log. (Atau Anda menggunakan AWS CLI untuk memanggil perintah get-deployment-instance.) Cari kesalahan seperti`invalid character`,`command not found`, atau`file not found`.

Untuk mengatasi masalah ini, kami merekomendasikan yang berikut:

- Jangan gunakan editor teks yang memperkenalkan karakter non-cetak seperti carriage return (`^M`karakter) ke dalam AppSpec file dan file skrip shell Anda.
- Gunakan editor teks yang menampilkan karakter non-pencetakan seperti carriage return dalam AppSpec file dan file skrip shell Anda, sehingga Anda dapat menemukan dan menghapus semua yang mungkin diperkenalkan. Untuk contoh jenis editor teks ini, cari di internet untuk editor teks yang menunjukkan pengembalian carriage.
- Gunakan editor teks yang berjalan di Amazon Linux, Ubuntu Server, atau instans RHEL untuk membuat file skrip shell yang berjalan di Amazon Linux, Ubuntu Server, atau instans RHEL. Untuk contoh jenis editor teks ini, cari di internet untuk editor skrip shell Linux.
- Jika Anda harus menggunakan editor teks di Windows atau macOS untuk membuat file skrip shell agar berjalan di Amazon Linux, Ubuntu Server, atau instance RHEL, gunakan program atau utilitas yang mengonversi teks dalam format Windows atau macOS ke format Unix. Untuk contoh program dan utilitas ini, cari di internet untuk DOS ke UNIX atau Mac ke UNIX. Pastikan untuk menguji file skrip shell yang dikonversi pada sistem operasi target.

Menggunakan Finder di macOS untuk menggabungkan revisi aplikasi dapat menyebabkan penerapan gagal

Penerapan mungkin gagal jika Anda menggunakan aplikasi antarmuka pengguna grafis Finder (GUI) di Mac untuk menggabungkan (zip) AppSpec file dan file serta skrip terkait ke dalam file arsip revisi aplikasi (.zip). Ini karena Finder membuat `__MACOSX` folder perantara dalam file.zip dan menempatkan file komponen ke dalamnya. CodeDeploy tidak dapat menemukan file komponen, sehingga penerapan gagal.

Untuk mengatasi masalah ini, kami sarankan Anda menggunakan perintah AWS CLI to call the push, yang mem-zip file komponen ke dalam struktur yang diharapkan. Atau, Anda dapat menggunakan Terminal alih-alih GUI untuk zip file komponen. Terminal tidak membuat `__MACOSX` folder perantara.

Memecahkan masalah EC2/On-Premises penerapan

Topik

- CodeDeploy CommandPoller kesalahan kredensialnya hilang plugin
- Penerapan gagal dengan pesan “Validasi pesan yang ditandatangani PKCS7 gagal”
- Penerapan atau pemindahan file yang sama ke lokasi instance yang sama gagal dengan kesalahan “Penerapan gagal karena file tertentu sudah ada di lokasi ini”
- Jalur file yang panjang menyebabkan kesalahan “Tidak ada file atau direktori seperti itu”
- Long-running proses dapat menyebabkan penerapan gagal
- Memecahkan masalah peristiwa AllowTraffic siklus hidup yang gagal tanpa kesalahan yang dilaporkan dalam log penerapan
- Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan
- Memecahkan masalah peristiwa siklus hidup DownloadBundle penerapan yang gagal dengan UnknownError: tidak dibuka untuk dibaca
- Memecahkan masalah semua peristiwa siklus hidup yang dilewati kesalahan
- PowerShell Skrip Windows gagal menggunakan versi Windows 64-bit PowerShell secara default

> **Note**
>
> Penyebab banyak kegagalan penerapan dapat diidentifikasi dengan meninjau file log yang dibuat selama proses penyebaran. Untuk mempermudah, sebaiknya gunakan Amazon CloudWatch Logs untuk memonitor file log secara terpusat alih-alih melihatnya secara instance per instance. Untuk selengkapnya, lihat Melihat CodeDeploy Log di Konsol CloudWatch Log.

> **Tip**
>
> Untuk runbook yang mengotomatiskan banyak tugas pemecahan masalah yang terkait dengan EC2/On-Premises penerapan, lihat di referensi runbook AWSSupport-TroubleshootCodeDeploySystems Manager AWS Automation.

CodeDeploy CommandPoller kesalahan kredensialnya hilang plugin

Jika Anda menerima kesalahan yang mirip dengan`InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing`

`credentials - please check if this instance was started with an IAM instance profile`, itu mungkin disebabkan oleh salah satu dari berikut ini:

- Instance yang Anda terapkan tidak memiliki profil instans IAM yang terkait dengannya.
- Profil instans IAM Anda tidak memiliki izin yang benar yang dikonfigurasi.

Profil instans IAM memberikan izin CodeDeploy agen untuk berkomunikasi dengan CodeDeploy dan mengunduh revisi Anda dari Amazon S3. Untuk instans EC2, lihat. Identitas dan manajemen akses untuk AWS CodeDeploy Untuk instance lokal, lihat. Working with On-Premises Instances

Penerapan gagal dengan pesan “Validasi pesan yang ditandatangani PKCS7 gagal”

Pesan kesalahan ini menunjukkan instance menjalankan versi CodeDeploy agen yang hanya mendukung algoritma SHA-1 hash. Support untuk algoritma SHA-2 hash diperkenalkan dalam versi 1.0.1.854 dari CodeDeploy agen, dirilis pada November 2015. Efektif 17 Oktober 2016, penerapan gagal jika versi CodeDeploy agen lebih awal dari 1.0.1.854 diinstal. Untuk informasi selengkapnya, lihat AWS untuk beralih ke algoritma hash SHA256 untuk Sertifikat SSL, PEMBERITAHUAN: Menghentikan agen CodeDeploy host yang lebih lama dari versi 1.0.1.85, dan. Perbarui CodeDeploy agen

Penerapan atau pemindahan file yang sama ke lokasi instance yang sama gagal dengan kesalahan “Penerapan gagal karena file tertentu sudah ada di lokasi ini”

Saat CodeDeploy mencoba menyebarkan file ke instance tetapi file dengan nama yang sama sudah ada di lokasi target yang ditentukan, penerapan ke instance itu mungkin gagal. Anda mungkin menerima pesan kesalahan “Penerapan gagal karena file tertentu sudah ada di lokasi ini:*`location-name`*.” Ini karena, selama setiap penerapan, CodeDeploy pertama-tama menghapus semua file dari penerapan sebelumnya, yang tercantum dalam file log pembersihan. Jika ada file di folder instalasi target yang tidak tercantum dalam file pembersihan ini, CodeDeploy agen secara default menafsirkan ini sebagai kesalahan dan gagal penerapan.

> **Note**
>
> Pada instance Amazon Linux, RHEL, dan Ubuntu Server, file pembersihan terletak di. `/opt/codedeploy-agent/deployment-root/deployment-instructions/` Pada

instance Windows Server, lokasinya adalah`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`.

Cara termudah untuk menghindari kesalahan ini adalah dengan menentukan opsi selain perilaku default untuk gagal penerapan. Untuk setiap penerapan, Anda dapat memilih apakah akan gagal penerapan, menimpa file yang tidak tercantum dalam file pembersihan, atau untuk menyimpan file yang sudah ada di instance.

Opsi overwrite berguna ketika, misalnya, Anda secara manual menempatkan file pada instance setelah penerapan terakhir, tetapi kemudian menambahkan file dengan nama yang sama ke revisi aplikasi berikutnya.

Anda dapat memilih opsi pertahankan untuk file yang Anda tempatkan pada instance yang Anda inginkan untuk menjadi bagian dari penerapan berikutnya tanpa harus menambahkannya ke paket revisi aplikasi. Opsi pertahankan juga berguna jika file aplikasi Anda sudah ada di lingkungan produksi Anda dan Anda ingin menerapkan menggunakan CodeDeploy untuk pertama kalinya. Untuk informasi selengkapnya, lihat Membuat penerapan Platform EC2/On-Premises Komputasi (konsol) dan Perilaku rollback dengan konten yang ada.

Pemecahan masalah **Penerapan gagal karena file tertentu sudah ada pada kesalahan penerapan lokasi ini**

Jika Anda memilih untuk tidak menentukan opsi untuk menimpa atau mempertahankan konten yang CodeDeploy mendeteksi di lokasi penyebaran target Anda (atau jika Anda tidak menentukan opsi penerapan apa pun untuk menangani konten yang ada dalam perintah terprogram), Anda dapat memilih untuk memecahkan masalah kesalahan.

Informasi berikut hanya berlaku jika Anda memilih untuk tidak menyimpan atau menimpa konten.

Jika Anda mencoba menerapkan ulang file dengan nama dan lokasi yang sama, pemindahan akan lebih mungkin berhasil jika Anda menentukan nama aplikasi dan grup penyebaran dengan ID grup penyebaran dasar yang sama yang Anda gunakan sebelumnya. CodeDeploy menggunakan ID grup penyebaran yang mendasari untuk mengidentifikasi file yang akan dihapus sebelum pemindahan.

Menyebarkan file baru atau memindahkan file yang sama ke lokasi yang sama pada instance dapat gagal karena alasan berikut:

- Anda menetapkan nama aplikasi yang berbeda untuk pemindahan revisi yang sama ke instance yang sama. Penerapan ulang gagal karena meskipun nama grup penyebaran sama, penggunaan nama aplikasi yang berbeda berarti ID grup penyebaran dasar yang berbeda sedang digunakan.
- Anda menghapus dan membuat ulang grup penyebaran untuk aplikasi dan kemudian mencoba menerapkan revisi yang sama ke grup penyebaran. Penerapan ulang gagal karena meskipun nama grup penerapan sama, CodeDeploy mereferensikan ID grup penerapan dasar yang berbeda.
- Anda menghapus aplikasi dan grup penyebaran di CodeDeploy, dan kemudian membuat grup aplikasi dan penyebaran baru dengan nama yang sama dengan yang Anda hapus. Setelah itu, Anda mencoba menerapkan revisi yang telah diterapkan ke grup penerapan sebelumnya ke yang baru dengan nama yang sama. Penerapan ulang gagal karena meskipun nama grup aplikasi dan penyebaran sama, CodeDeploy masih mereferensikan ID grup penyebaran yang Anda hapus.
- Anda menerapkan revisi ke grup penerapan dan kemudian menerapkan revisi yang sama ke grup penerapan lain ke instance yang sama. Penerapan kedua gagal karena CodeDeploy mereferensikan ID grup penyebaran dasar yang berbeda.
- Anda menerapkan revisi ke satu grup penerapan dan kemudian menerapkan revisi lain ke grup penerapan lain ke instance yang sama. Setidaknya ada satu file dengan nama yang sama dan di lokasi yang sama yang coba diterapkan oleh grup penyebaran kedua. Penerapan kedua gagal karena CodeDeploy tidak menghapus file yang ada sebelum penerapan kedua dimulai. Kedua penerapan mereferensikan ID grup penerapan yang berbeda.
- Anda menerapkan revisi di CodeDeploy, tetapi setidaknya ada satu file dengan nama yang sama dan di lokasi yang sama. Penerapan gagal karena, secara default, CodeDeploy tidak menghapus file yang ada sebelum penerapan dimulai.

Untuk mengatasi situasi ini, lakukan salah satu hal berikut:

- Hapus file dari lokasi dan instance yang sebelumnya digunakan, lalu coba penerapannya lagi.
- Dalam AppSpec file revisi Anda, baik dalam peristiwa siklus hidup ApplicationStop atau BeforeInstall penerapan, tentukan skrip khusus untuk menghapus file di lokasi mana pun yang cocok dengan file yang akan dipasang revisi Anda.
- Menerapkan atau menyebarkan ulang file ke lokasi atau instance yang bukan bagian dari penerapan sebelumnya.
- Sebelum Anda menghapus aplikasi atau grup penyebaran, gunakan revisi yang berisi file yang menetapkan tidak ada AppSpec file yang akan disalin ke instance. Untuk penerapan, tentukan nama aplikasi dan nama grup penyebaran yang menggunakan ID grup aplikasi dan penyebaran dasar yang sama dengan yang akan Anda hapus. (Anda dapat menggunakan perintah get-

deployment-group untuk mengambil ID grup penyebaran.) CodeDeploymenggunakan ID dan AppSpec file grup penyebaran yang mendasari untuk menghapus semua file yang diinstal dalam penerapan yang berhasil sebelumnya.

Jalur file yang panjang menyebabkan kesalahan “Tidak ada file atau direktori seperti itu”

Untuk penerapan ke instance Windows, jika Anda memiliki jalur file yang lebih besar dari 260 karakter di bagian file file appspec.yl Anda, Anda mungkin melihat penerapan gagal dengan kesalahan yang mirip dengan berikut ini:

`No such file or directory @ dir_s_mkdir - C:\your-long-file-path`

Kesalahan ini terjadi karena Windows secara default tidak mengizinkan jalur file lebih dari 260 karakter, seperti yang dijelaskan dalam dokumentasi Microsoft.

Untuk versi CodeDeploy agen 1.4.0 atau yang lebih baru, Anda dapat mengaktifkan jalur file panjang dengan dua cara, tergantung pada proses instalasi agen:

Jika CodeDeploy agen belum diinstal:

1. Pada mesin tempat Anda berencana untuk menginstal CodeDeploy agen, aktifkan kunci registri `LongPathsEnabled` Windows menggunakan perintah ini:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
          -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Instal CodeDeploy agen. Untuk informasi selengkapnya, lihat Instal CodeDeploy agen.

Jika CodeDeploy agen telah diinstal:

1. Pada mesin CodeDeploy agen, aktifkan kunci registri `LongPathsEnabled` Windows menggunakan perintah ini:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Mulai ulang CodeDeploy agen agar perubahan kunci registri berlaku. Untuk me-restart agen, gunakan perintah ini:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Long-running proses dapat menyebabkan penerapan gagal

Untuk penerapan ke Amazon Linux, Ubuntu Server, dan instans RHEL, jika Anda memiliki skrip penerapan yang memulai proses yang berjalan lama, CodeDeploy mungkin menghabiskan waktu lama menunggu dalam peristiwa siklus hidup penerapan dan kemudian gagal penerapan. Ini karena jika proses berjalan lebih lama dari proses latar depan dan latar belakang dalam peristiwa itu diharapkan untuk mengambil, CodeDeploy menghentikan dan gagal penerapan, bahkan jika proses masih berjalan seperti yang diharapkan.

Misalnya, revisi aplikasi berisi dua file di akarnya, `after-install.sh` dan`sleep.sh`. AppSpec File-nya berisi instruksi berikut:

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

`after-install.sh`File berjalan selama acara siklus hidup AfterInstall aplikasi. Berikut isinya:

```
#!/bin/bash
/tmp/sleep.sh
```

`sleep.sh`File berisi berikut ini, yang menangguhkan eksekusi program selama tiga menit (180 detik), mensimulasikan beberapa proses yang berjalan lama:

```
#!/bin/bash
sleep 180
```

Ketika `after-install.sh` panggilan`sleep.sh`, `sleep.sh` mulai dan berjalan selama tiga menit (180 detik), yaitu dua menit (120 detik) melewati waktu yang CodeDeploy diharapkan `sleep.sh`

(dan, menurut hubungan,`after-install.sh`) untuk berhenti berjalan. Setelah batas waktu satu menit (60 detik), CodeDeploy berhenti dan gagal penerapan pada acara siklus hidup AfterInstall aplikasi, meskipun `sleep.sh` terus berjalan seperti yang diharapkan. Kesalahan berikut ditampilkan:

`Script at specified location: after-install.sh failed to complete in 60 seconds`.

Anda tidak bisa begitu saja menambahkan ampersand (&) `after-install.sh` untuk dijalankan `sleep.sh` di latar belakang.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

Melakukannya dapat membuat penerapan dalam status tertunda hingga periode batas waktu peristiwa siklus hidup penerapan satu jam default, setelah itu CodeDeploy menghentikan dan gagal penerapan pada acara siklus hidup aplikasi seperti sebelumnya. AfterInstall

Dalam`after-install.sh`, panggil `sleep.sh` sebagai berikut, yang memungkinkan CodeDeploy untuk melanjutkan setelah proses mulai berjalan:

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

Pada panggilan sebelumnya, `sleep.sh` adalah nama proses yang ingin Anda mulai jalankan di latar belakang, mengarahkan stdout, stderr, dan stdin ke. `/dev/null`

Memecahkan masalah peristiwa AllowTraffic siklus hidup yang gagal tanpa kesalahan yang dilaporkan dalam log penerapan

Dalam beberapa kasus, blue/green penerapan gagal selama peristiwa AllowTraffic siklus hidup, tetapi log penerapan tidak menunjukkan penyebab kegagalan.

Kegagalan ini biasanya disebabkan oleh pemeriksaan kesehatan yang tidak dikonfigurasi dengan benar di Elastic Load Balancing untuk Classic Load Balancer, Application Load Balancer, atau Network Load Balancer yang digunakan untuk mengelola lalu lintas untuk grup penyebaran.

Untuk mengatasi masalah ini, tinjau dan perbaiki kesalahan apa pun dalam konfigurasi pemeriksaan kesehatan untuk penyeimbang beban.

Untuk Classic Load Balancer, lihat Mengonfigurasi Pemeriksaan Kesehatan di Panduan Pengguna untuk Penyeimbang Beban Klasik dan ConfigureHealthCheckdi Referensi API Elastic Load Balancing versi 2012-06-01.

Untuk Application Load Balancers, lihat Health Check for Your Target Groups di Panduan Pengguna untuk Application Load Balancers.

Untuk Network Load Balancer, lihat Pemeriksaan Kesehatan untuk Grup Target Anda di Panduan Pengguna Network Load Balancer.

Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan

Selama penerapan, CodeDeploy agen menjalankan skrip yang ditentukan untuk ApplicationStop, BeforeBlockTraffic, dan AfterBlockTraffic dalam AppSpec file dari penerapan yang berhasil sebelumnya. (Semua skrip lain dijalankan dari AppSpec file dalam penerapan saat ini.) Jika salah satu skrip ini berisi kesalahan dan tidak berhasil berjalan, deployment dapat gagal.

Kemungkinan alasan kegagalan ini meliputi:

- CodeDeploy Agen menemukan *`deployment-group-id`*`_last_successful_install` file di lokasi yang benar, tetapi lokasi yang tercantum dalam *`deployment-group-id`*`_last_successful_install` file tidak ada.

  Di Amazon Linux, Ubuntu Server, dan instans RHEL, file ini harus ada di. `/opt/codedeploy-agent/deployment-root/deployment-instructions`

  Pada instance Windows Server, file ini harus disimpan di `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions` folder.
- Di lokasi yang tercantum dalam *`deployment-group-id`*`_last_successful_install` file, AppSpec file tersebut tidak valid atau skrip tidak berjalan dengan sukses.
- Skrip berisi kesalahan yang tidak dapat diperbaiki, sehingga tidak pernah berjalan dengan sukses.

Gunakan CodeDeploy konsol untuk menyelidiki mengapa penerapan mungkin gagal selama salah satu peristiwa ini. Pada halaman detail untuk penerapan, pilih Lihat peristiwa. Pada halaman detail untuk instance, di AfterBlockTrafficbaris ApplicationStopBeforeBlockTraffic, atau, pilih Lihat log. Atau gunakan AWS CLI untuk memanggil perintah get-deployment-instance.

Jika penyebab kegagalan adalah skrip dari penerapan terakhir yang berhasil yang tidak pernah berjalan dengan sukses, buat penerapan dan tentukan bahwa ApplicationStop, BeforeBlockTraffic, dan AfterBlockTraffic kegagalan harus diabaikan. Ada dua cara untuk melakukan hal ini:

- Gunakan CodeDeploy konsol untuk membuat penyebaran. Pada halaman Buat penerapan, di bawah kegagalan peristiwa ApplicationStop siklus hidup, pilih Jangan gagal penerapan ke instance jika peristiwa siklus hidup ini pada instance gagal.
- Gunakan tombol AWS CLI untuk memanggil create-deployment perintah dan sertakan `--ignore-application-stop-failures` opsi.

Saat Anda menerapkan revisi aplikasi lagi, penerapan berlanjut meskipun salah satu dari tiga peristiwa siklus hidup ini gagal. Jika revisi baru menyertakan skrip tetap untuk peristiwa siklus hidup tersebut, penerapan masa depan dapat berhasil tanpa menerapkan perbaikan ini.

Memecahkan masalah peristiwa siklus hidup DownloadBundle penerapan yang gagal dengan UnknownError: tidak dibuka untuk dibaca

Jika Anda mencoba menerapkan revisi aplikasi dari Amazon S3, dan penerapan gagal selama peristiwa siklus hidup penerapan DownloadBundle dengan kesalahan: `UnknownError: not opened for reading`

- Ada kesalahan layanan Amazon S3 internal. Terapkan revisi aplikasi lagi.
- Profil instans IAM pada instans EC2 Anda tidak memiliki izin untuk mengakses revisi aplikasi di Amazon S3. Untuk informasi tentang kebijakan bucket Amazon S3, lihat Dorong revisi CodeDeploy ke Amazon S3 EC2/On-Premises (hanya penerapan) dan. Prasyarat penyebaran
- Instans yang Anda gunakan terkait dengan satu AWS Wilayah (misalnya, US West (Oregon)), tetapi bucket Amazon S3 yang berisi revisi aplikasi dikaitkan dengan AWS Wilayah lain (misalnya, US East (Virginia N.)). Pastikan revisi aplikasi ada di bucket Amazon S3 yang terkait dengan Wilayah yang AWS sama dengan instans.

Pada halaman detail acara untuk penerapan, di baris bundel Unduh, pilih Lihat log. Atau gunakan AWS CLI untuk memanggil perintah get-deployment-instance. Jika kesalahan ini terjadi, harus ada kesalahan dalam output dengan kode kesalahan `UnknownError` dan pesan kesalahan`not opened for reading`.

Untuk menentukan alasan kesalahan ini:

1. Aktifkan wire logging pada setidaknya satu instance, lalu terapkan revisi aplikasi lagi.
2. Periksa file wire logging untuk menemukan kesalahan. Pesan kesalahan umum untuk masalah ini termasuk frasa “akses ditolak.”
3. Setelah Anda memeriksa file log, kami sarankan Anda menonaktifkan wire logging untuk mengurangi ukuran file log dan jumlah informasi sensitif yang mungkin muncul dalam output dalam teks biasa pada instance di masa depan.

Untuk informasi tentang cara menemukan file wire logging dan mengaktifkan dan menonaktifkan wire logging, lihat `:log_aws_wire:` di referensi konfigurasi CodeDeploy agen.

Memecahkan masalah semua peristiwa siklus hidup yang dilewati kesalahan

Jika semua peristiwa siklus hidup EC2 atau penerapan lokal dilewati, Anda mungkin menerima kesalahan yang mirip dengan. `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)` Berikut adalah beberapa kemungkinan penyebab dan solusi:

- CodeDeploy Agen mungkin tidak diinstal atau dijalankan pada instance. Untuk menentukan apakah CodeDeploy agen sedang berjalan:
  - Untuk Amazon Linux RHEL atau server Ubuntu, jalankan yang berikut ini:

    ```
    systemctl status codedeploy-agent
    ```

  - Untuk Windows, jalankan yang berikut ini:

    ```
    powershell.exe -Command Get-Service -Name CodeDeployagent
    ```

  Jika CodeDeploy agen tidak diinstal atau berjalan, lihatVerifikasi CodeDeploy agen sedang berjalan.

  Instans Anda mungkin tidak dapat mencapai titik akhir publik Amazon S3 CodeDeploy atau menggunakan port 443. Cobalah salah satu cara berikut ini:

  - Tetapkan alamat IP publik ke instance dan gunakan tabel rute untuk memungkinkan akses internet. Pastikan grup keamanan yang terkait dengan instance memungkinkan akses keluar

melalui port 443 (HTTPS). Untuk informasi selengkapnya, lihat Protokol komunikasi dan port untuk CodeDeploy agen.

  - Jika sebuah instance disediakan dalam subnet pribadi, gunakan gateway NAT alih-alih gateway internet di tabel rute. Lihat informasi yang lebih lengkap di Gateway NAT.

- Peran layanan untuk CodeDeploy mungkin tidak memerlukan izin. Untuk mengonfigurasi peran CodeDeploy layanan, lihatLangkah 2: Buat peran layanan untuk CodeDeploy.
- Jika Anda menggunakan proxy HTTP, pastikan itu ditentukan dalam `:proxy_uri:` pengaturan dalam file konfigurasi CodeDeploy agen. Untuk informasi selengkapnya, lihat CodeDeploy referensi konfigurasi agen.
- Tanda tangan tanggal dan waktu instans penerapan Anda mungkin tidak cocok dengan tanda tangan tanggal dan waktu permintaan penerapan Anda. Cari kesalahan yang mirip dengan `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` di file log CodeDeploy agen Anda. Jika Anda melihat kesalahan ini, ikuti langkah-langkahnyaPemecahan masalah "InvalidSignatureException - Tanda tangan kedaluwarsa: [waktu] sekarang lebih awal dari [waktu]" kesalahan penerapan. Untuk informasi selengkapnya, lihat Lihat data log untuk CodeDeploy EC2/On-Premises penerapan.
- CodeDeploy Agen mungkin berhenti berjalan karena instance kehabisan memori atau ruang hard disk. Cobalah untuk menurunkan jumlah penerapan yang diarsipkan pada instans Anda dengan memperbarui `max_revisions` pengaturan dalam konfigurasi agen. CodeDeploy Jika Anda melakukan ini untuk instans EC2 dan masalah tetap ada, pertimbangkan untuk menggunakan instance yang lebih besar. Misalnya, jika tipe instans Anda`t2.small`, coba gunakan`t2.medium`. Untuk informasi selengkapnya, lihatFile yang diinstal oleh CodeDeploy agen,CodeDeploy referensi konfigurasi agen, dan Jenis instans.
- Instance yang Anda gunakan mungkin tidak memiliki profil instans IAM yang dilampirkan, atau mungkin memiliki profil instans IAM yang dilampirkan yang tidak memiliki izin yang diperlukan.
  - Jika profil instans IAM tidak dilampirkan ke instance Anda, buat profil dengan izin yang diperlukan lalu lampirkan.
  - Jika profil instans IAM sudah dilampirkan ke instans Anda, pastikan ia memiliki izin yang diperlukan.

  Setelah Anda mengonfirmasi profil instans terlampir Anda dikonfigurasi dengan izin yang diperlukan, mulai ulang instance Anda. Untuk informasi selengkapnya, lihat Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda dan Peran IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2.

PowerShell Skrip Windows gagal menggunakan versi Windows 64-bit PowerShell secara default

Jika PowerShell skrip Windows yang berjalan sebagai bagian dari penerapan bergantung pada fungsionalitas 64-bit (misalnya, karena menghabiskan lebih banyak memori daripada aplikasi 32-bit yang memungkinkan atau memanggil pustaka yang ditawarkan hanya dalam versi 64-bit), skrip mungkin macet atau tidak berjalan seperti yang diharapkan. Ini karena, secara default, CodeDeploy menggunakan versi Windows 32-bit PowerShell untuk menjalankan PowerShell skrip Windows yang merupakan bagian dari revisi aplikasi.

Tambahkan kode seperti berikut ini ke awal skrip apa pun yang harus dijalankan dengan Windows versi 64-bit 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...
# ...
```

Meskipun informasi jalur file dalam kode ini mungkin tampak berlawanan dengan intuisi, Windows 32-bit PowerShell menggunakan jalur seperti:

```
c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
```

Windows 64-bit PowerShell menggunakan jalur seperti:

```
c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Memecahkan masalah penerapan Amazon ECS

Topik

- Waktu tunggu terjadi saat menunggu set tugas pengganti
- Waktu tunggu terjadi saat menunggu pemberitahuan dilanjutkan
- Peran IAM tidak memiliki izin yang cukup
- Waktu penerapan habis saat menunggu panggilan balik status
- Penerapan gagal karena satu atau lebih fungsi validasi peristiwa siklus hidup gagal
- ELB tidak dapat diperbarui karena kesalahan berikut: Kelompok target tugas utama harus berada di belakang pendengar
- Penerapan saya terkadang gagal saat menggunakan Auto Scaling
- Hanya ALB yang mendukung perutean lalu lintas bertahap, gunakan perutean AllAtOnce lalu lintas sebagai gantinya saat Anda grup create/update Deployment
- Meskipun penerapan saya berhasil, set tugas pengganti gagal dalam pemeriksaan kesehatan Elastic Load Balancing, dan aplikasi saya mati
- Dapatkah saya melampirkan beberapa penyeimbang beban ke grup penerapan?
- Dapatkah saya melakukan CodeDeploy blue/green penerapan tanpa penyeimbang beban?
- Bagaimana cara memperbarui layanan Amazon ECS saya dengan informasi baru selama penerapan?

Waktu tunggu terjadi saat menunggu set tugas pengganti

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: CodeDeploy

```
The deployment timed out while waiting for the replacement task set to
become healthy. This time out period is 60 minutes.
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika ada kesalahan dalam file definisi tugas atau file terkait penerapan lainnya. Misalnya, jika ada kesalahan ketik di `image` bidang dalam file definisi tugas Anda, Amazon ECS akan mencoba menarik gambar kontainer yang salah dan terus gagal, menyebabkan kesalahan ini.

Kemungkinan perbaikan dan langkah selanjutnya:

- Perbaiki kesalahan ketik dan masalah konfigurasi di file definisi tugas Anda dan file lainnya.
- Periksa acara layanan Amazon ECS terkait dan cari tahu mengapa tugas penggantian tidak menjadi sehat. Untuk informasi selengkapnya tentang acara Amazon ECS, lihat acara Amazon ECS di Panduan Pengembang Layanan Kontainer Amazon Elastic.
- Periksa bagian pemecahan masalah Amazon ECS di Panduan Pengembang Layanan Kontainer Amazon Elas tic untuk mengetahui kesalahan yang terkait dengan pesan dalam acara tersebut.

Waktu tunggu terjadi saat menunggu pemberitahuan dilanjutkan

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: CodeDeploy

`The deployment timed out while waiting for a notification to continue. This time out period is` *`n`* `minutes.`

Kemungkinan penyebabnya: Kesalahan ini mungkin terjadi jika Anda menentukan waktu tunggu di bidang Tentukan kapan harus mengalihkan rute lalu lintas saat Anda membuat grup penerapan, tetapi penerapan tidak dapat selesai sebelum waktu tunggu berakhir.

Kemungkinan perbaikan dan langkah selanjutnya:

- Di grup penerapan Anda, setel T entukan kapan mengarahkan ulang lalu lintas ke jumlah waktu yang lebih besar dan penerapan ulang. Untuk informasi selengkapnya, lihat Membuat grup penerapan untuk penyebaran Amazon ECS (konsol).
- Di grup penerapan Anda, ubah T entukan kapan mengarahkan ulang lalu lintas ke Mengalihkan lalu lintas segera dan menyebarkan kembali. Untuk informasi selengkapnya, lihat Membuat grup penerapan untuk penyebaran Amazon ECS (konsol).
- Depload ulang dan kemudian jalankan AWS CLI perintah aws deploy continue-deployment dengan `--deployment-wait-type` opsi disetel ke. `READY_WAIT` Pastikan untuk menjalankan perintah ini sebelum waktu yang ditentukan dalam T entukan kapan untuk mengalihkan rute lalu lintas kedaluwarsa.

Peran IAM tidak memiliki izin yang cukup

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: CodeDeploy

```
The IAM role role-arn does not give you permission to perform operations in
the following AWS service: AWSLambda.
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika Anda menentukan fungsi Lambda di `Hooks` bagian AppSpec file, tetapi Anda tidak memberikan CodeDeploy izin ke layanan Lambda.

Kemungkinan perbaikan: Tambahkan `lambda:InvokeFunction` izin ke peran CodeDeploy layanan. Untuk menambahkan izin ini, tambahkan salah satu kebijakan AWS-managed berikut ke peran: **`AWSCodeDeployRoleForECS`** atau**`AWSCodeDeployRoleForECSLimited`**. Untuk informasi tentang kebijakan ini dan cara menambahkannya ke peran CodeDeploy layanan, lihatLangkah 2: Buat peran layanan untuk CodeDeploy.

Waktu penerapan habis saat menunggu panggilan balik status

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: 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.
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika Anda menentukan fungsi Lambda di `Hooks` bagian AppSpec file, tetapi fungsi Lambda tidak dapat memanggil `PutLifecycleEventHookExecutionStatus` API yang diperlukan untuk mengembalikan `Failed` status `Succeeded` atau. CodeDeploy

Kemungkinan perbaikan dan langkah selanjutnya:

- Tambahkan `codedeploy:putlifecycleEventHookExecutionStatus` izin ke peran eksekusi Lambda yang digunakan oleh fungsi Lambda yang Anda tentukan dalam file. AppSpec Izin ini memberikan fungsi Lambda kemampuan untuk mengembalikan status `Succeeded` atau `Failed` ke CodeDeploy. Untuk informasi selengkapnya tentang peran eksekusi Lambda, lihat Per an eksekusi Lambda di Panduan AWS Lambda Pengguna.
- Periksa kode fungsi Lambda Anda dan log eksekusi untuk memastikan fungsi Lambda Anda memanggil CodeDeploy `PutLifecycleEventHookExecutionStatus` API untuk

menginform CodeDeploy asikan apakah uji `Succeeded` validasi siklus hidup atau. `Failed` Untuk informasi tentang `putlifecycleEventHookExecutionStatus` API, lihat PutLifecycleEventHookExecutionStatus di Referensi AWS CodeDeploy API. Untuk informasi tentang log eksekusi Lambda, lihat Meng akses CloudWatch log Amazon untuk AWS Lambda.

Penerapan gagal karena satu atau lebih fungsi validasi peristiwa siklus hidup gagal

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: CodeDeploy

```
The deployment failed because one or more of the lifecycle event validation functions failed.
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika Anda menentukan fungsi Lambda di `Hooks` bagian AppSpec file, tetapi fungsi Lambda kembali `Failed` ke CodeDeploy saat dipanggil. `PutLifecycleEventHookExecutionStatus` Kegagalan ini menunjukkan CodeDeploy bahwa uji validasi siklus hidup gagal.

Kemungkinan langkah selanjutnya: Periksa log eksekusi Lambda Anda untuk melihat mengapa kode uji validasi gagal. Untuk informasi tentang log eksekusi Lambda, lihat Meng akses CloudWatch log Amazon untuk AWS Lambda.

ELB tidak dapat diperbarui karena kesalahan berikut: Kelompok target tugas utama harus berada di belakang pendengar

Masalah: Anda melihat pesan galat berikut saat menerapkan aplikasi Amazon ECS menggunakan: CodeDeploy

```
The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika Anda telah mengonfigurasi pendengar pengujian opsional, dan dikonfigurasi dengan kelompok target yang salah. Untuk informasi selengkapnya tentang pendengar tes di CodeDeploy, lihat Sebelum memulai deployment Amazon ECS danApa yang terjadi selama penyebaran Amazon ECS. Untuk informasi selengkapnya tentang set tugas, lihat Refer TaskSet ensi API Layanan Kontainer Amazon Elastic dan des kripsi-task-set di bagian Amazon ECS Referensi Perintah. AWS CLI

Kemungkinan perbaikan: Pastikan pendengar produksi dan pendengar uji Elastic Load Balancing sama-sama menunjuk ke grup target yang saat ini melayani beban kerja Anda. Ada tiga tempat untuk memeriksa:

- Di Amazon EC2, di pengaturan Listener dan aturan penyeim bang beban Anda. Untuk informasi selengkapnya, lihat Pen dengar untuk Penyeimbang Beban Aplikasi Anda di Panduan Pengguna untuk Penyeimbang Beban Aplikasi, atau Pen dengar untuk Penyeimbang Beban Jaringan Anda di Panduan Pengguna untuk Penyeimbang Beban Jaringan.
- Di Amazon ECS, di cluster Anda, di bawah konfigurasi Jaringan layanan Anda. Untuk informasi selengkapnya, lihat Pertimbangan Penyeimbang Beban Aplikasi dan Penyeimbang Beban Jaringan di Panduan Pengembang Amazon Elastic Container Service.
- Di dalam CodeDeploy, di setelan grup penerapan Anda. Untuk informasi selengkapnya, lihat Membuat grup penerapan untuk penyebaran Amazon ECS (konsol).

Penerapan saya terkadang gagal saat menggunakan Auto Scaling

Masalah: Anda menggunakan Auto Scaling dengan CodeDeploy dan Anda melihat bahwa penerapan Anda terkadang gagal. Untuk informasi selengkapnya tentang gejala masalah ini, lihat topik yang berbunyi Untuk layanan yang dikonfigurasi untuk menggunakan penskalaan otomatis layanan dan jenis blue/green penerapan, penskalaan otomatis tidak diblokir selama penerapan tetapi penerapan mungkin gagal dalam beberapa keadaan di Panduan Pengembang Layanan Kontainer Amazon Elastic.

Kemungkinan penyebab: Masalah ini mungkin terjadi jika CodeDeploy dan proses Auto Scaling bertentangan.

Kemungkinan perbaikan: Tangguhkan dan lanjutkan proses Penskalaan Otomatis selama pener `RegisterScalableTarget` apan menggunakan API (atau `register-scalable-target` AWS CLI perintah yang sesuai). CodeDeploy Untuk informasi selengkapnya, lihat Men angguhkan dan melanjutkan penskalaan untuk Penskalaan Otomatis Aplikasi di Panduan Pengguna Penskalaan Otomatis Aplikasi.

> **Note**
>
> CodeDeploy Tidak bisa menelepon `RegisterScaleableTarget` secara langsung. Untuk menggunakan API ini, Anda harus mengonfigurasi CodeDeploy untuk mengirim pemberitahuan atau acara ke Amazon Simple Notification Service (atau Amazon

CloudWatch). Anda kemudian harus mengonfigurasi Amazon SNS (or CloudWatch) untuk memanggil fungsi Lambda, dan mengonfigurasi fungsi Lambda untuk memanggil API. `RegisterScalableTarget` `RegisterScalableTarget`API harus dipanggil dengan `SuspendedState` parameter yang disetel `true` untuk menangguhkan operasi Auto Scaling, dan `false` untuk melanjutkannya.
Pemberitahuan atau peristiwa yang CodeDeploy dikirim harus terjadi saat penerapan dimulai (untuk memicu operasi penangguhan Penskalaan Otomatis), atau ketika penerapan berhasil, gagal, atau berhenti (untuk memicu operasi melanjutkan Penskalaan Otomatis).
Untuk informasi tentang cara mengonfigurasi CodeDeploy untuk menghasilkan notifikasi atau CloudWatch acara Amazon SNS, lihatMemantau penerapan dengan Amazon Events CloudWatch. danMonitoring Deployments with Amazon SNS Event Notifications.

Hanya ALB yang mendukung perutean lalu lintas bertahap, gunakan perutean AllAtOnce lalu lintas sebagai gantinya saat Anda grup create/update Deployment

Masalah: Anda melihat pesan galat berikut saat membuat atau memperbarui grup penerapan di CodeDeploy:

```
Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group.
```

Kemungkinan penyebab: Kesalahan ini mungkin terjadi jika Anda menggunakan Network Load Balancer dan mencoba menggunakan konfigurasi penerapan yang telah ditentukan sebelumnya selain`CodeDeployDefault.ECSAllAtOnce`.

Kemungkinan perbaikan:

- Ubah konfigurasi penerapan yang telah ditentukan sebelumnya menjadi`CodeDeployDefault.ECSAllAtOnce`. Ini adalah satu-satunya konfigurasi penerapan yang telah ditentukan yang didukung oleh Network Load Balancers.

  Untuk informasi selengkapnya tentang konfigurasi penerapan yang telah ditentukan sebelumnya, lihatKonfigurasi penerapan yang telah ditentukan sebelumnya untuk platform komputasi Amazon ECS.
- Ubah penyeimbang beban Anda ke Application Load Balancer. Application Load Balancer mendukung semua konfigurasi penerapan yang telah ditentukan sebelumnya. Untuk informasi

selengkapnya tentang membuat Application Load Balancer, lihatSiapkan penyeimbang beban, grup target, dan pendengar untuk penerapan Amazon CodeDeploy ECS.

Meskipun penerapan saya berhasil, set tugas pengganti gagal dalam pemeriksaan kesehatan Elastic Load Balancing, dan aplikasi saya mati

Masalah: Meskipun CodeDeploy menunjukkan bahwa penerapan saya berhasil, set tugas pengganti gagal dalam pemeriksaan kesehatan dari Elastic Load Balancing, dan aplikasi saya mati.

Kemungkinan penyebab: Masalah ini mungkin terjadi jika Anda melakukan pener CodeDeploy apan serba sekaligus, dan set tugas pengganti (hijau) berisi kode buruk yang menyebabkan pemeriksaan kesehatan Elastic Load Balancing gagal. Dengan konfigurasi penerapan serba sekaligus, pemeriksaan kesehatan penyeimbang beban mulai berjalan pada set tugas pengganti setelah lalu lintas dialihkan ke sana (yaitu, setelah peristiwa *`AllowTraffic`* siklus hidup CodeDeploy terjadi). Itu sebabnya Anda akan melihat pemeriksaan kesehatan gagal pada tugas penggantian yang ditetapkan setelah lalu lintas bergeser, tetapi tidak sebelumnya. Untuk informasi tentang peristiwa siklus hidup yang CodeDeploy dihasilkan, lihatApa yang terjadi selama penyebaran Amazon ECS.

Kemungkinan perbaikan:

- Ubah konfigurasi penerapan Anda dari serba sekaligus menjadi kenari atau linier. Dalam konfigurasi canary atau linear, pemeriksaan kesehatan penyeimbang beban mulai berjalan pada set tugas pengganti saat CodeDeploy menginstal aplikasi Anda di lingkungan pengganti, dan sebelum lalu lintas digeser (yaitu, selama peristiwa `Install` siklus hidup, dan sebelum `AllowTraffic` acara). Dengan mengizinkan pemeriksaan berjalan selama instalasi aplikasi tetapi sebelum lalu lintas dialihkan, kode aplikasi yang buruk akan terdeteksi dan menyebabkan kegagalan penerapan sebelum aplikasi tersedia untuk umum.

  Untuk informasi tentang cara mengkonfigurasi penerapan canary atau linear, lihat. Ubah pengaturan grup penerapan dengan CodeDeploy

  Untuk informasi tentang peristiwa CodeDeploy siklus hidup yang berjalan selama penerapan Amazon ECS, lihatApa yang terjadi selama penyebaran Amazon ECS.

> **Note**
>
> Konfigurasi penyebaran Canary dan linier hanya didukung dengan Application Load Balancers.

- Jika Anda ingin mempertahankan konfigurasi penerapan serba sekaligus, siapkan pendengar pengujian dan periksa status kesehatan dari set tugas pengganti dengan kait `BeforeAllowTraffic` siklus hidup. Untuk informasi selengkapnya, lihat Daftar kait peristiwa siklus hidup untuk penerapan Amazon ECS.

Dapatkah saya melampirkan beberapa penyeimbang beban ke grup penerapan?

Tidak. Jika Anda ingin menggunakan beberapa Penyeimbang Beban Aplikasi atau Penyeimbang Beban Jaringan, gunakan pembaruan bergulir Amazon ECS alih-alih penerapan. CodeDeploy blue/green Untuk informasi selengkapnya tentang pembaruan bergulir, lihat Pembaruan ber gulir di Panduan Pengembang Layanan Kontainer Amazon Elastic. Untuk informasi selengkapnya tentang menggunakan beberapa penyeimbang beban dengan Amazon ECS, lihat Mendaf tarkan beberapa grup target dengan layanan di Panduan Pengembang Layanan Kontainer Amazon Elastic.

Dapatkah saya melakukan CodeDeploy blue/green penerapan tanpa penyeimbang beban?

Tidak, Anda tidak dapat melakukan CodeDeploy blue/green penerapan tanpa penyeimbang beban. Jika Anda tidak dapat menggunakan penyeimbang beban, gunakan fitur pembaruan bergulir Amazon ECS sebagai gantinya. Untuk informasi selengkapnya tentang pembaruan bergulir Amazon ECS, lihat Pembaruan bergulir di Panduan Pengembang Layanan Kontainer Amazon Elastic.

Bagaimana cara memperbarui layanan Amazon ECS saya dengan informasi baru selama penerapan?

Untuk CodeDeploy memperbarui layanan Amazon ECS Anda dengan parameter baru saat melakukan penerapan, tentukan parameter di `resources` bagian AppSpec file. Hanya beberapa parameter Amazon ECS yang didukung oleh CodeDeploy, seperti file definisi tugas dan parameter nama wadah. Untuk daftar lengkap parameter Amazon ECS yang CodeDeploy dapat diperbarui, lihatAppSpec Bagian 'sumber daya' untuk penerapan Amazon ECS.

> **Note**
>
> Jika Anda perlu memperbarui layanan Amazon ECS Anda dengan parameter yang tidak didukung oleh CodeDeploy, selesaikan tugas-tugas berikut:
>
> 1. Hubungi `UpdateService` API Amazon ECS dengan parameter yang ingin Anda perbarui. Untuk daftar lengkap parameter yang dapat diperbarui, lihat UpdateService Referensi API Layanan Kontainer Amazon Elastic.
> 2. Untuk menerapkan perubahan pada tugas, buat penerapan Amazon ECS blue/green baru. Lihat informasi yang lebih lengkap di Membuat penerapan Platform Komputasi Amazon ECS (konsol).

Pemecahan Masalah AWS Masalah penyebaran Lambda

Topik

- AWS Lambda penerapan gagal setelah menghentikan penerapan Lambda secara manual yang tidak memiliki rollback yang dikonfigurasi

AWS Lambda penerapan gagal setelah menghentikan penerapan Lambda secara manual yang tidak memiliki rollback yang dikonfigurasi

Dalam beberapa kasus, alias fungsi Lambda yang ditentukan dalam penerapan mungkin mereferensikan dua versi fungsi yang berbeda. Hasilnya adalah upaya selanjutnya untuk menerapkan fungsi Lambda gagal. Penerapan Lambda dapat masuk dalam status ini ketika tidak memiliki rollback yang dikonfigurasi dan dihentikan secara manual. Untuk melanjutkan, gunakan AWS Lambda konsol untuk memastikan fungsi tidak dikonfigurasi untuk mengalihkan lalu lintas antara dua versi:

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Dari panel kiri, pilih Fungsi.
3. Pilih nama fungsi Lambda yang ada di penerapan Anda CodeDeploy .
4. Dari Alias, pilih alias yang digunakan dalam CodeDeploy penerapan Anda, lalu pilih Edit.

5. Dari alias Tertimbang, pilih. **none** Ini memastikan alias tidak dikonfigurasi untuk menggeser persentase, atau bobot, lalu lintas ke lebih dari satu versi. Catat versi yang dipilih di Versi.
6. Pilih Simpan.
7. Buka CodeDeploy konsol dan coba penerapan versi yang ditampilkan di menu tarik-turun di langkah 5.

Memecahkan masalah grup penyebaran

Menandai instance sebagai bagian dari grup penerapan tidak secara otomatis menyebarkan aplikasi Anda ke instance baru

CodeDeploy tidak secara otomatis menyebarkan aplikasi Anda ke instance yang baru diberi tag. Anda harus membuat penyebaran baru di grup penyebaran.

Anda dapat menggunakan CodeDeploy untuk mengaktifkan penerapan otomatis ke instans EC2 baru di grup Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling.

Memecahkan masalah contoh

Topik

- Tag harus diatur dengan benar
- AWS CodeDeploy agen harus diinstal dan berjalan pada instance
- Penerapan tidak gagal hingga satu jam saat instance dihentikan selama penerapan
- Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance
- Buat file CodeDeploy log baru jika tidak sengaja dihapus
- Pemecahan masalah "InvalidSignatureException - Tanda tangan kedaluwarsa: [waktu] sekarang lebih awal dari [waktu]” kesalahan penerapan

Tag harus diatur dengan benar

Gunakan perintah list-deployment-instance untuk mengonfirmasi bahwa instance yang digunakan untuk penerapan diberi tag dengan benar. Jika instans EC2 hilang dalam output, gunakan konsol EC2 untuk mengonfirmasi bahwa tag telah disetel pada instance. Untuk informasi selengkapnya, lihat Bekerja dengan tag di konsol di Panduan Pengguna Amazon EC2.

> **Note**
>
> Jika Anda menandai instance dan segera menggunakannya CodeDeploy untuk menyebarkan aplikasi ke instans tersebut, instance tersebut mungkin tidak disertakan dalam penerapan. Hal ini karena dapat mengambil beberapa menit sebelum CodeDeploy dapat membaca tag. Kami menyarankan Anda menunggu setidaknya lima menit antara waktu Anda menandai instance dan mencoba menerapkannya.

AWS CodeDeploy agen harus diinstal dan berjalan pada instance

Untuk memverifikasi CodeDeploy agen diinstal dan berjalan pada sebuah instance, lihatVerifikasi CodeDeploy agen sedang berjalan.

Untuk menginstal, menghapus, atau menginstal ulang CodeDeploy agen, lihat. Instal CodeDeploy agen

Penerapan tidak gagal hingga satu jam saat instance dihentikan selama penerapan

CodeDeploy menyediakan jendela satu jam untuk setiap peristiwa siklus hidup penerapan agar berjalan hingga selesai. Ini memberikan waktu yang cukup untuk skrip yang berjalan lama.

Jika skrip tidak berjalan hingga selesai saat peristiwa siklus hidup sedang berlangsung (misalnya, jika instance dihentikan atau CodeDeploy agen dimatikan), mungkin diperlukan waktu hingga satu jam agar status penerapan ditampilkan sebagai Gagal. Ini benar bahkan jika periode batas waktu yang ditentukan dalam skrip lebih pendek dari satu jam. Ini karena ketika instance dihentikan, CodeDeploy agen dimatikan dan tidak dapat memproses lebih banyak skrip.

Jika instance dihentikan antara peristiwa siklus hidup atau sebelum langkah peristiwa siklus hidup pertama dimulai, batas waktu akan terjadi setelah hanya lima menit.

Menganalisis file log untuk menyelidiki kegagalan penerapan pada instance

Jika status instance dalam penerapan adalah apa pun selain`Succeeded`, Anda dapat meninjau data file log penerapan untuk membantu mengidentifikasi masalah. Untuk informasi tentang mengakses data log penerapan, lihat. Lihat data log untuk CodeDeploy EC2/On-Premises penerapan

Buat file CodeDeploy log baru jika tidak sengaja dihapus

Jika Anda secara tidak sengaja menghapus file log penyebaran pada sebuah instance, CodeDeploy tidak membuat file log pengganti. Untuk membuat file log baru, masuk ke instance, lalu jalankan perintah ini:

Untuk contoh Amazon Linux, Ubuntu Server, atau RHEL, jalankan perintah ini dalam urutan ini, satu per satu:

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Untuk contoh Windows Server:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Pemecahan masalah "InvalidSignatureException - Tanda tangan kedaluwarsa: [waktu] sekarang lebih awal dari [waktu]" kesalahan penerapan

CodeDeploy membutuhkan referensi waktu yang akurat untuk melakukan operasinya. Jika tanggal dan waktu pada instans Anda tidak disetel dengan benar, mereka mungkin tidak cocok dengan tanggal tanda tangan permintaan penerapan Anda, yang CodeDeploy menolak.

Untuk menghindari kegagalan penerapan yang terkait dengan pengaturan waktu yang salah, lihat topik berikut:

- Mengatur Waktu untuk Instans Linux Anda
- Mengatur Waktu untuk Instans Windows

Memecahkan masalah GitHub token

Token OAuth tidak valid GitHub

CodeDeploy aplikasi yang dibuat setelah Juni 2017 menggunakan token GitHub OAuth untuk setiap AWS Wilayah. Penggunaan token yang terkait dengan AWS Wilayah tertentu memberi Anda kontrol lebih besar atas CodeDeploy aplikasi mana yang memiliki akses ke GitHub repositori.

Jika Anda menerima kesalahan GitHub token, Anda mungkin memiliki token lama yang sekarang tidak valid.

Untuk memperbaiki token OAuth yang tidak valid GitHub

1. Hapus token lama menggunakan salah satu metode berikut:
   - Untuk menghapus token lama menggunakan API, gunakan DeleteGitHubAccountToken.
   - Untuk menghapus token lama menggunakan AWS Command Line Interface:
     a. Pergi ke komputer tempat token berada.
     b. Pastikan AWS CLI sudah terpasang di komputer ini. Untuk petunjuk penginstalan, lihat Menginstal, memperbarui, dan menghapus instalasi AWS CLI di AWS Command Line Interface Panduan Pengguna
     c. Masukkan perintah berikut di komputer tempat token berada:

        ```
        aws delete-git-hub-account-token
        ```

        Untuk detail tentang sintaks perintah, lihat delete-git-hub-account-token.
2. Tambahkan token OAuth baru. Untuk informasi selengkapnya, lihat Integrasi dengan CodeDeploy GitHub.

Jumlah maksimum token GitHub OAuth terlampaui

Saat Anda membuat CodeDeploy penerapan, jumlah maksimum GitHub token yang diizinkan adalah 10. Jika Anda menerima kesalahan tentang token GitHub OAuth, pastikan Anda memiliki 10 atau lebih sedikit token. Jika Anda memiliki lebih dari 10 token, token pertama yang dibuat tidak valid. Misalnya, jika Anda memiliki 11 token, token pertama yang Anda buat tidak valid. Jika Anda memiliki 12 token, dua token pertama yang Anda buat tidak valid. Untuk informasi tentang penggunaan CodeDeploy API untuk menghapus token lama, lihat DeleteGitHubAccountToken.

Memecahkan masalah Amazon EC2 Auto Scaling

Topik

- Pemecahan masalah Penskalaan Otomatis Amazon EC2 Umum
- “CodeDeployRole tidak memberi Anda izin untuk melakukan operasi sebagai berikut AWS layanan: AmazonAutoScaling "kesalahan
- Instans dalam grup Amazon EC2 Auto Scaling terus disediakan dan dihentikan sebelum revisi dapat diterapkan
- Mengakhiri atau me-reboot instans Amazon EC2 Auto Scaling dapat menyebabkan penerapan gagal
- Hindari mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2
- Instans EC2 dalam grup Amazon EC2 Auto Scaling gagal diluncurkan dan menerima kesalahan “Heartbeat Timeout”
- Kait siklus hidup Amazon EC2 Auto Scaling yang tidak cocok dapat menyebabkan penerapan otomatis ke grup Amazon EC2 Auto Scaling berhenti atau gagal
- Kesalahan “Penerapan gagal karena tidak ada instance yang ditemukan untuk grup penerapan Anda”

Pemecahan masalah Penskalaan Otomatis Amazon EC2 Umum

Penerapan ke instans EC2 dalam grup Amazon EC2 Auto Scaling dapat gagal karena alasan berikut:

- Amazon EC2 Auto Scaling terus meluncurkan dan menghentikan instans EC2. Jika CodeDeploy tidak dapat menerapkan revisi aplikasi secara otomatis, Amazon EC2 Auto Scaling akan terus meluncurkan dan menghentikan instans EC2.

  Putuskan grup Penskalaan Otomatis Amazon EC2 dari CodeDeploy grup penerapan atau ubah konfigurasi grup Penskalaan Otomatis Amazon EC2 Anda sehingga jumlah instans yang diinginkan cocok dengan jumlah instans saat ini (sehingga mencegah Amazon EC2 Auto Scaling meluncurkan instans EC2 lagi). Untuk informasi selengkapnya, lihat Ubah pengaturan grup penerapan dengan CodeDeploy atau Penskalaan Manual untuk Amazon EC2 Auto Scaling.
- CodeDeploy Agen tidak responsif. CodeDeploy Agen mungkin tidak diinstal jika skrip inisialisasi (misalnya, skrip cloud-init) yang berjalan segera setelah instans EC2 diluncurkan atau dimulai membutuhkan waktu lebih dari satu jam untuk dijalankan. CodeDeploy memiliki batas waktu

satu jam bagi CodeDeploy agen untuk menanggapi penerapan yang tertunda. Untuk mengatasi masalah ini, pindahkan skrip inisialisasi Anda ke dalam revisi CodeDeploy aplikasi Anda.

- Instans EC2 dalam grup Amazon EC2 Auto Scaling reboot selama penerapan. Penerapan Anda bisa gagal jika instans EC2 di-boot ulang selama penerapan atau CodeDeploy agen dimatikan saat memproses perintah penerapan. Untuk informasi selengkapnya, lihat Mengakhiri atau me-reboot instans Amazon EC2 Auto Scaling dapat menyebabkan penerapan gagal.
- Beberapa revisi aplikasi diterapkan secara bersamaan ke instans EC2 yang sama dalam grup Amazon EC2 Auto Scaling. Menerapkan beberapa revisi aplikasi ke instans EC2 yang sama dalam grup Amazon EC2 Auto Scaling secara bersamaan dapat gagal jika salah satu penerapan memiliki skrip yang berjalan selama lebih dari beberapa menit. Jangan menerapkan beberapa revisi aplikasi ke instans EC2 yang sama di grup Penskalaan Otomatis Amazon EC2.
- Penerapan gagal untuk instans EC2 baru yang diluncurkan sebagai bagian dari grup Amazon EC2 Auto Scaling. Dalam skenario ini, menjalankan skrip dalam penerapan dapat mencegah peluncuran instans EC2 di grup Amazon EC2 Auto Scaling. (Instans EC2 lainnya di grup Amazon EC2 Auto Scaling mungkin tampak berjalan normal.) Untuk mengatasi masalah ini, pastikan semua skrip lainnya selesai terlebih dahulu:
  - CodeDeploy agen tidak termasuk dalam AMI Anda: Jika Anda menggunakan cfn-init perintah untuk menginstal CodeDeploy agen saat meluncurkan instance baru, letakkan skrip instalasi agen di akhir `cfn-init` bagian CloudFormation template Anda.
  - CodeDeploy agen disertakan dalam AMI Anda: Konfigurasikan AMI sehingga agen dalam `Stopped` keadaan saat instance dibuat, lalu sertakan skrip untuk memulai agen sebagai langkah terakhir di pustaka `cfn-init` skrip Anda.

“CodeDeployRole tidak memberi Anda izin untuk melakukan operasi sebagai berikut AWS layanan: AmazonAutoScaling "kesalahan

Penerapan yang menggunakan grup Auto Scaling yang dibuat dengan template peluncuran memerlukan izin berikut. Ini adalah tambahan dari izin yang diberikan oleh kebijakan `AWSCodeDeployRole` AWS terkelola.

- `EC2:RunInstances`
- `EC2:CreateTags`
- `iam:PassRole`

Anda mungkin menerima kesalahan ini jika Anda kehilangan izin ini. Untuk informasi selengkapnya, lihatTutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling, Membuat template peluncuran untuk grup Auto Scaling, dan Izin di Panduan Pengguna Amazon EC2 Auto Scaling.

Instans dalam grup Amazon EC2 Auto Scaling terus disediakan dan dihentikan sebelum revisi dapat diterapkan

Dalam beberapa kasus, kesalahan dapat mencegah penerapan yang berhasil ke instans yang baru disediakan di grup Penskalaan Otomatis Amazon EC2. Hasilnya bukan contoh yang sehat dan tidak ada penerapan yang berhasil. Karena penerapan tidak dapat berjalan atau diselesaikan dengan sukses, instance dihentikan segera setelah dibuat. Konfigurasi grup Amazon EC2 Auto Scaling kemudian menyebabkan kumpulan instans lain disediakan untuk mencoba memenuhi persyaratan host sehat minimum. Batch ini juga dihentikan, dan siklus berlanjut.

Kemungkinan penyebabnya meliputi:

- Pemeriksaan kesehatan grup Amazon EC2 Auto Scaling yang gagal.
- Kesalahan dalam revisi aplikasi.

Untuk mengatasi masalah ini, ikuti langkah-langkah berikut:

1. Buat instans EC2 secara manual yang bukan bagian dari grup Amazon EC2 Auto Scaling. Tandai instance dengan tag instans EC2 yang unik.
2. Tambahkan instance baru ke grup penyebaran yang terpengaruh.
3. Terapkan revisi aplikasi baru yang bebas kesalahan ke grup penyebaran.

Ini meminta grup Amazon EC2 Auto Scaling untuk menerapkan revisi aplikasi ke instans masa depan di grup Amazon EC2 Auto Scaling.

> **Note**
>
> Setelah mengonfirmasi bahwa penerapan berhasil, hapus instans yang Anda buat untuk menghindari tagihan yang sedang berlangsung ke akun Anda AWS .

Mengakhiri atau me-reboot instans Amazon EC2 Auto Scaling dapat menyebabkan penerapan gagal

Jika instans EC2 diluncurkan melalui Amazon EC2 Auto Scaling, dan instans kemudian dihentikan atau di-boot ulang, penerapan ke instance tersebut mungkin gagal karena alasan berikut:

- Selama penerapan yang sedang berlangsung, peristiwa scale-in atau peristiwa penghentian lainnya menyebabkan instans terlepas dari grup Penskalaan Otomatis Amazon EC2 dan kemudian dihentikan. Karena penerapan tidak dapat diselesaikan, itu gagal.
- Instans di-boot ulang, tetapi dibutuhkan lebih dari lima menit untuk memulai instance. CodeDeploy memperlakukan ini sebagai batas waktu. Layanan gagal semua penerapan saat ini dan masa depan ke instance.

Untuk mengatasi masalah ini:

- Secara umum, pastikan bahwa semua penerapan selesai sebelum instance dihentikan atau di-boot ulang. Pastikan bahwa semua penerapan dimulai setelah instance dimulai atau di-boot ulang.
- Penerapan dapat gagal jika Anda menentukan Amazon Machine Image (AMI) berbasis Windows Server untuk konfigurasi Amazon EC2 Auto Scaling dan menggunakan layanan EC2config untuk menyetel nama komputer instance. Untuk memperbaiki masalah ini, di basis Windows Server AMI, pada tab Umum Properti Layanan EC2, hapus Set Nama Komputer. Setelah Anda menghapus kotak centang ini, perilaku ini dinonaktifkan untuk semua instans Penskalaan Otomatis Amazon EC2 Windows Server baru yang diluncurkan dengan AMI basis Windows Server. Untuk instans Penskalaan Otomatis Amazon EC2 Windows Server yang mengaktifkan perilaku ini, Anda tidak perlu menghapus kotak centang ini. Cukup terapkan ulang penerapan yang gagal ke instance tersebut setelah di-boot ulang.

Hindari mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2

Sebagai praktik terbaik, Anda harus mengaitkan hanya satu grup penerapan dengan setiap grup Penskalaan Otomatis Amazon EC2.

Ini karena jika Amazon EC2 Auto Scaling meningkatkan skala instance yang memiliki kait yang terkait dengan beberapa grup penerapan, maka akan mengirimkan notifikasi untuk semua hook sekaligus. Hal ini menyebabkan beberapa penerapan ke setiap instance dimulai pada waktu yang

bersamaan. Saat beberapa penerapan mengirim perintah ke CodeDeploy agen secara bersamaan, batas waktu lima menit antara peristiwa siklus hidup dan awal penerapan atau akhir peristiwa siklus hidup sebelumnya mungkin tercapai. Jika ini terjadi, penerapan gagal, bahkan jika proses penerapan berjalan seperti yang diharapkan.

> **Note**
>
> Batas waktu default untuk skrip dalam acara siklus hidup adalah 30 menit. Anda dapat mengubah batas waktu ke nilai yang berbeda dalam AppSpec file. Untuk informasi selengkapnya, lihat Menambahkan AppSpec file untuk EC2/On-Premises penerapan.

Tidak mungkin untuk mengontrol urutan penerapan terjadi jika lebih dari satu penerapan mencoba dijalankan pada saat yang bersamaan.

Terakhir, jika penerapan ke instans apa pun gagal, Amazon EC2 Auto Scaling segera menghentikan instance. Ketika instance pertama dimatikan, penerapan lain yang berjalan mulai gagal. Karena CodeDeploy memiliki batas waktu satu jam bagi CodeDeploy agen untuk merespons penerapan yang tertunda, dibutuhkan waktu hingga 60 menit untuk setiap instance hingga time out.

Untuk informasi selengkapnya tentang Amazon EC2 Auto Scaling, lihat Under the CodeDeploy hood: dan integrasi Auto Scaling.

Instans EC2 dalam grup Amazon EC2 Auto Scaling gagal diluncurkan dan menerima kesalahan “Heartbeat Timeout”

Grup Amazon EC2 Auto Scaling mungkin gagal meluncurkan instans EC2 baru, menghasilkan pesan yang mirip dengan berikut ini:

```
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.
```

Pesan ini biasanya menunjukkan salah satu dari yang berikut:

- Jumlah maksimum penerapan bersamaan yang terkait dengan AWS akun tercapai. Untuk informasi selengkapnya tentang batas penerapan, lihatCodeDeploy kuota.

- Grup Auto Scaling mencoba meluncurkan terlalu banyak instans EC2 terlalu cepat. Panggilan API ke RecordLifecycleActionHeartbeatatau CompleteLifecycleActionuntuk setiap instance baru dibatasi.
- Aplikasi di CodeDeploy dihapus sebelum grup penyebaran terkait diperbarui atau dihapus.

  Saat Anda menghapus aplikasi atau grup penerapan, CodeDeploy mencoba membersihkan kait Penskalaan Otomatis Amazon EC2 yang terkait dengannya, tetapi beberapa kait mungkin tetap ada. Jika Anda menjalankan perintah untuk menghapus grup penyebaran, kait sisa dikembalikan dalam output. Namun, jika Anda menjalankan perintah untuk menghapus aplikasi, kait sisa tidak muncul di output.

  Oleh karena itu, sebagai praktik terbaik, Anda harus menghapus semua grup penyebaran yang terkait dengan aplikasi sebelum Anda menghapus aplikasi. Anda dapat menggunakan output perintah untuk mengidentifikasi kait siklus hidup yang harus dihapus secara manual.

Jika Anda menerima pesan galat “Heartbeat Timeout”, Anda dapat menentukan apakah kait siklus hidup yang tersisa adalah penyebabnya dan menyelesaikan masalah dengan melakukan hal berikut:

1. Lakukan salah satu tindakan berikut:

   - Panggil perintah delete-deployment-group untuk menghapus grup deployment yang terkait dengan grup Auto Scaling yang menyebabkan batas waktu detak jantung.
   - Panggil perintah update-deployment-group dengan daftar nama grup Auto Scaling yang tidak kosong untuk melepaskan semua kait siklus hidup Auto Scaling. CodeDeploy-managed

     Misalnya, masukkan AWS CLI perintah berikut:

     ```
     aws deploy update-deployment-group --application-name my-example-app
     --current-deployment-group-name my-deployment-group --auto-scaling-
     groups
     ```

     Sebagai contoh lain, jika Anda menggunakan CodeDeploy API dengan Java, panggil `UpdateDeploymentGroup` dan atur `autoScalingGroups` kenew `ArrayList<String>()`. Ini ditetapkan `autoScalingGroups` ke daftar kosong dan menghapus daftar yang ada. Jangan gunakan`null`, yang merupakan default, karena ini meninggalkan apa `autoScalingGroups` adanya, yang bukan yang Anda inginkan.

Periksa output panggilan. Jika output berisi `hooksNotCleanedUp` struktur dengan daftar kait siklus hidup Amazon EC2 Auto Scaling, ada sisa kait siklus hidup.

2. Panggil perintah describe-lifecycle-hooks, dengan menentukan nama grup Amazon EC2 Auto Scaling yang terkait dengan instans EC2 yang gagal diluncurkan. Dalam output, cari salah satu dari berikut ini:

   - Nama kait siklus hidup Amazon EC2 Auto Scaling yang sesuai dengan struktur yang Anda identifikasi pada `hooksNotCleanedUp` langkah 1.
   - Nama kait siklus hidup Amazon EC2 Auto Scaling yang berisi nama grup penerapan yang terkait dengan grup Auto Scaling yang gagal.
   - Nama kait siklus hidup Amazon EC2 Auto Scaling yang mungkin menyebabkan batas waktu detak jantung untuk penerapan. CodeDeploy

3. Jika hook jatuh ke dalam salah satu kategori yang tercantum dalam langkah 2, panggil perintah delete-lifecycle-hook untuk menghapusnya. Tentukan grup Penskalaan Otomatis Amazon EC2 dan kait siklus hidup dalam panggilan.

   > **Important**
   >
   > Hanya hapus kait yang menyebabkan masalah, seperti yang diuraikan pada langkah 2. Jika Anda menghapus hook yang layak, penerapan Anda mungkin gagal, atau CodeDeploy mungkin tidak dapat menerapkan revisi aplikasi Anda ke instans EC2 yang diskalakan.

4. Panggil perintah update-deployment-group atau create-deployment-group dengan nama grup Auto Scaling yang diinginkan. CodeDeploymenginstal ulang kait Auto Scaling dengan UUID baru.

> **Note**
>
> Jika Anda melepaskan grup Auto Scaling dari grup penerapan, setiap penerapan CodeDeploy yang sedang berlangsung ke grup Auto Scaling mungkin gagal, dan instans EC2 baru yang diskalakan oleh grup Auto Scaling tidak akan menerima revisi aplikasi Anda. CodeDeploy Agar Auto Scaling berfungsi kembali CodeDeploy, Anda harus

melampirkan kembali grup Auto Scaling ke grup penerapan dan memanggil yang baru `CreateDeployment` untuk memulai penerapan di seluruh armada.

Kait siklus hidup Amazon EC2 Auto Scaling yang tidak cocok dapat menyebabkan penerapan otomatis ke grup Amazon EC2 Auto Scaling berhenti atau gagal

Amazon EC2 Auto CodeDeploy Scaling dan gunakan kait siklus hidup untuk menentukan revisi aplikasi mana yang harus diterapkan ke instans EC2 mana setelah diluncurkan di grup Amazon EC2 Auto Scaling. Penerapan otomatis dapat berhenti atau gagal jika kait siklus hidup dan informasi tentang kait ini tidak sama persis di Amazon EC2 Auto Scaling dan. CodeDeploy

Jika penerapan ke grup Penskalaan Otomatis Amazon EC2 gagal, lihat apakah nama kait siklus hidup di Amazon EC2 Auto Scaling dan cocok. CodeDeploy Jika tidak, gunakan panggilan AWS CLI perintah ini.

Pertama, dapatkan daftar nama kait siklus hidup untuk grup Amazon EC2 Auto Scaling dan grup penerapan:

1. Panggil perintah describe-lifecycle-hooks, dengan menentukan nama grup Amazon EC2 Auto Scaling yang terkait dengan grup penerapan di. CodeDeploy Dalam output, dalam `LifecycleHooks` daftar, buat catatan setiap `LifecycleHookName` nilai.
2. Panggil perintah get-deployment-group, dengan menentukan nama grup penyebaran yang terkait dengan grup Amazon EC2 Auto Scaling. Dalam output, dalam `autoScalingGroups` daftar, temukan setiap item yang nilai namanya cocok dengan nama grup Amazon EC2 Auto Scaling, lalu catat nilai yang sesuai. `hook`

Sekarang bandingkan dua set nama kait siklus hidup. Jika mereka cocok persis, karakter untuk karakter, maka ini bukan masalahnya. Anda mungkin ingin mencoba langkah-langkah pemecahan masalah Amazon EC2 Auto Scaling lainnya yang dijelaskan di tempat lain di bagian ini.

Namun, jika dua set nama kait siklus hidup tidak sama persis, karakter untuk karakter, lakukan hal berikut:

1. Jika ada nama kait siklus hidup dalam output describe-lifecycle-hooks perintah yang tidak juga dalam output get-deployment-group perintah, maka lakukan hal berikut:

   a. Untuk setiap nama hook siklus hidup dalam output describe-lifecycle-hooks perintah, panggil perintah delete-lifecycle-hook.
   b. Panggil perintah update-deployment-group, dengan menentukan nama grup Amazon EC2 Auto Scaling asli. CodeDeploy membuat kait siklus hidup pengganti baru di grup Penskalaan Otomatis Amazon EC2 dan mengaitkan kait siklus hidup dengan grup penerapan. Penerapan otomatis sekarang harus dilanjutkan saat instans baru ditambahkan ke grup Penskalaan Otomatis Amazon EC2.

2. Jika ada nama kait siklus hidup dalam output get-deployment-group perintah yang tidak juga dalam output describe-lifecycle-hooks perintah, lakukan hal berikut:

   a. Panggil perintah update-deployment-group, tetapi jangan tentukan nama grup Amazon EC2 Auto Scaling asli.
   b. Panggil update-deployment-group perintah lagi, tetapi kali ini tentukan nama grup Amazon EC2 Auto Scaling asli. CodeDeploy membuat ulang kait siklus hidup yang hilang di grup Amazon EC2 Auto Scaling. Penerapan otomatis sekarang harus dilanjutkan saat instans baru ditambahkan ke grup Penskalaan Otomatis Amazon EC2.

Setelah Anda mendapatkan dua set nama kait siklus hidup agar sama persis, karakter untuk karakter, revisi aplikasi harus diterapkan lagi, tetapi hanya untuk instance baru saat ditambahkan ke grup Amazon EC2 Auto Scaling. Penerapan tidak terjadi secara otomatis ke instans yang sudah ada di grup Amazon EC2 Auto Scaling.

Kesalahan “Penerapan gagal karena tidak ada instance yang ditemukan untuk grup penerapan Anda”

Baca bagian ini jika Anda melihat CodeDeploy kesalahan berikut:

```
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.
```

Kemungkinan penyebab kesalahan ini adalah:

1. Setelan grup penerapan Anda menyertakan tag untuk instans EC2, instans lokal, atau grup Auto Scaling yang tidak benar. Untuk memperbaiki masalah ini, periksa apakah tag Anda sudah benar, lalu gunakan kembali aplikasi Anda.

2. Armada Anda diperkecil setelah penyebaran dimulai. Dalam skenario ini, Anda melihat contoh sehat di `InService` negara bagian dalam armada Anda, tetapi Anda juga melihat kesalahan di atas. Untuk memperbaiki masalah ini, gunakan kembali aplikasi Anda.
3. Grup Auto Scaling Anda tidak menyertakan instance apa pun yang berada dalam status. `InService` Dalam skenario ini, ketika Anda mencoba melakukan penerapan di seluruh armada, penerapan gagal dengan pesan kesalahan di atas karena CodeDeploy membutuhkan setidaknya satu instance untuk berada dalam status. `InService` Ada banyak alasan mengapa Anda mungkin tidak memiliki contoh di `InService` negara bagian. Beberapa di antaranya termasuk:
   - Anda menjadwalkan (atau mengonfigurasi secara manual) ukuran grup Auto Scaling. `0`
   - Auto Scaling mendeteksi instans EC2 yang buruk (misalnya, instans EC2 mengalami kegagalan perangkat keras), jadi membatalkan semuanya, sehingga tidak meninggalkan status apa pun. `InService`
   - Selama acara skala keluar dari `0` ke`1`, CodeDeploy menerapkan revisi yang sebelumnya berhasil (disebut revisi sukses terakhir) yang telah menjadi tidak sehat sejak penerapan terakhir. Hal ini menyebabkan penerapan pada instance scaled-out gagal, yang pada gilirannya menyebabkan Auto Scaling membatalkan instance, sehingga tidak ada instance dalam status. `InService`

     Jika Anda menemukan bahwa Anda tidak memiliki instance di `InService` negara bagian, pecahkan masalah seperti yang dijelaskan dalam prosedur berikut,. To troubleshoot the error if there are no instances in the InService state

Untuk memecahkan masalah kesalahan jika tidak ada contoh di negara bagian InService

1. Di konsol Amazon EC2, verifikasi pengaturan Kapasitas yang Diinginkan. Jika nol, atur ke angka positif. Tunggu instance menjadi`InService`, yang berarti penerapan berhasil. Anda telah memperbaiki masalah dan dapat melewati langkah-langkah yang tersisa dari prosedur pemecahan masalah ini. Untuk informasi tentang menyetel setelan Kapasitas yang Diinginkan, lihat Menyetel batas kapasitas pada grup Auto Scaling di Panduan Pengguna Amazon EC2 Auto Scaling.
2. Jika Auto Scaling terus mencoba meluncurkan instans EC2 baru untuk memenuhi kapasitas yang diinginkan tetapi tidak pernah dapat memenuhi skala, biasanya karena hook siklus hidup Auto Scaling yang gagal. Memecahkan masalah ini sebagai berikut:

a. Untuk memeriksa peristiwa hook siklus hidup Auto Scaling mana yang gagal, lihat Memverifikasi aktivitas penskalaan untuk grup Auto Scaling di Panduan Pengguna Amazon EC2 Auto Scaling.

b. Jika nama hook yang gagal adalah`CodeDeploy-managed-automatic-launch-deployment-hook-`*`DEPLOYMENT_GROUP_NAME`*, buka, cari grup penerapan, dan temukan penerapan gagal yang dimulai oleh Auto Scaling. CodeDeploy Kemudian selidiki mengapa penerapan gagal.

c. Jika Anda memahami mengapa penerapan gagal (misalnya, CloudWatch alarm terjadi) dan Anda dapat memperbaiki masalah tanpa mengubah revisi, lakukan sekarang.

d. Jika, setelah penyelidikan, Anda menentukan bahwa CodeDeploy revisi terakhir yang berhasil tidak lagi sehat, dan tidak ada contoh sehat di grup Auto Scaling Anda, Anda berada dalam skenario kebuntuan penerapan. Untuk mengatasi masalah ini, Anda harus memperbaiki CodeDeploy revisi buruk dengan menghapus CodeDeploy sementara kait siklus hidup dari grup Auto Scaling, lalu menginstal ulang hook dan menerapkan kembali revisi baru (baik). Untuk instruksi, lihat:

   - To fix the deployment deadlock issue (CLI)
   - To fix the deployment deadlock issue (console)

Untuk memperbaiki masalah kebuntuan penerapan (CLI)

1. (Opsional) Blokir CI/CD pipeline Anda yang menyebabkan CodeDeploy kesalahan sehingga penerapan yang tidak terduga tidak terjadi saat Anda memperbaiki masalah ini.
2. Perhatikan pengaturan Auto Scaling DesiredCapacityAnda saat ini:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name ASG_NAME
   ```

   Anda mungkin perlu mengurangi angka ini di akhir prosedur ini.
3. Setel pengaturan Auto Scaling DesiredCapacityAnda ke. 1 Ini opsional jika kapasitas yang Anda inginkan lebih besar daripada 1 awalnya. Dengan menguranginya menjadi1, instance akan membutuhkan lebih sedikit waktu untuk menyediakan dan menerapkan nanti, yang mempercepat pemecahan masalah. Jika kapasitas yang diinginkan Auto Scaling Anda awalnya disetel ke0, Anda harus meningkatkannya menjadi. 1 Ini wajib.

   aws autoscaling set-desired-capacity --auto-scaling-group-name --desired-capacity 1 *ASG_NAME*

> **Note**
>
> Langkah-langkah yang tersisa dari prosedur ini mengasumsikan Anda telah DesiredCapacitymengaturnya1.

Pada titik ini, Auto Scaling mencoba menskalakan ke satu instance. Kemudian, karena hook yang CodeDeploy ditambahkan masih ada, CodeDeploy mencoba menerapkan; penerapan gagal; Auto Scaling membatalkan instance; dan Auto Scaling mencoba meluncurkan kembali instance untuk mencapai kapasitas yang diinginkan, yang lagi-lagi gagal. Anda berada dalam loop cancel-relaunch.

4. De-register grup Auto Scaling dari grup penerapan:

> **Warning**
>
> Perintah berikut akan meluncurkan instans EC2 baru tanpa perangkat lunak di dalamnya. Sebelum menjalankan perintah, pastikan bahwa `InService` instance Auto Scaling yang tidak menjalankan perangkat lunak dapat diterima. Misalnya, pastikan penyeimbang beban yang terkait dengan instance tidak mengirimkan lalu lintas ke host ini tanpa perangkat lunak.

> **Important**
>
> Gunakan CodeDeploy perintah yang ditunjukkan di bawah ini untuk menghapus kait. Jangan lepaskan kait melalui layanan Auto Scaling, karena penghapusan tidak akan dikenali oleh. CodeDeploy

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups
```

Setelah menjalankan perintah ini, berikut ini terjadi:

a. CodeDeploy menghapus registrasi grup Auto Scaling dari grup penerapan.

b. CodeDeploy menghapus hook siklus hidup Auto Scaling dari grup Auto Scaling.

c. Karena hook yang menyebabkan penerapan gagal tidak lagi ada, Auto Scaling membatalkan instans EC2 yang ada dan segera meluncurkan yang baru untuk menskalakan ke kapasitas yang diinginkan. Contoh baru harus segera pindah ke `InService` negara bagian. Contoh baru tidak termasuk perangkat lunak.

5. Tunggu instans EC2 memasuki `InService` status. Untuk memverifikasi statusnya, gunakan perintah berikut:

   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
   ```

6. Tambahkan hook kembali ke instans EC2:

   > **Important**
   >
   > Gunakan CodeDeploy perintah yang ditunjukkan di bawah ini untuk menambahkan hook. Jangan gunakan layanan Auto Scaling untuk menambahkan hook, karena penambahan tidak akan dikenali oleh. CodeDeploy

   ```
   aws deploy update-deployment-group --application-name APPLICATION_NAME --current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-groups ASG_NAME
   ```

   Setelah menjalankan perintah ini, berikut ini terjadi:

   a. CodeDeploy menginstal ulang hook siklus hidup Auto Scaling ke instans EC2

   b. CodeDeploy mendaftarkan ulang grup Auto Scaling dengan grup penerapan.

7. Buat penyebaran seluruh armada dengan Amazon S3 atau GitHub revisi yang Anda tahu sehat dan ingin digunakan.

   Misalnya, jika revisi adalah file.zip di bucket Amazon S3 yang `my-revision-bucket` dipanggil dengan kunci objek, masukkan `httpd_app.zip` perintah berikut:

   ```
   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}"
   ```

Karena sekarang ada satu `InService` instance dalam grup Auto Scaling, penerapan ini seharusnya berfungsi, dan Anda seharusnya tidak lagi melihat kesalahan Penerapan gagal karena tidak ada instance yang ditemukan untuk grup penerapan Anda.

8. Setelah penerapan berhasil, skala grup Auto Scaling Anda kembali ke kapasitas semula, jika sebelumnya Anda menskalakannya:

   ```
   aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME
   --desired-capacity ORIGINAL_CAPACITY
   ```

Untuk memperbaiki masalah kebuntuan penerapan (konsol)

1. (Opsional) Blokir CI/CD pipeline Anda yang menyebabkan CodeDeploy kesalahan sehingga penerapan yang tidak terduga tidak terjadi saat Anda memperbaiki masalah ini.
2. Buka konsol Amazon EC2, dan perhatikan pengaturan kapasitas Auto Scaling Desired Anda. Anda mungkin perlu mengurangi angka ini di akhir prosedur ini. Untuk informasi tentang menemukan setelan ini, lihat Menyetel batas kapasitas pada grup Auto Scaling Anda.
3. Setel jumlah instans EC2 yang diinginkan ke: 1

   Ini opsional jika kapasitas yang Anda inginkan lebih besar daripada 1 awalnya. Dengan menguranginya menjadi1, instance akan membutuhkan lebih sedikit waktu untuk menyediakan dan menerapkan nanti, yang mempercepat pemecahan masalah. Jika kapasitas Auto Scaling Desired Anda awalnya disetel ke`0`, Anda harus meningkatkannya. 1 Ini wajib.

   > **Note**
   >
   > Langkah-langkah yang tersisa dari prosedur ini mengasumsikan Anda telah mengatur kapasitas yang diinginkan Anda1.

   a. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/, dan pilih Grup Auto Scaling dari panel navigasi.

   b. Pilih wilayah yang sesuai.

   c. Pergi ke grup Auto Scaling yang bermasalah.

   d. Di Detail grup, pilih Edit.

   e. Atur kapasitas yang diinginkan ke**1**.

f. Pilih Perbarui.

4. De-register grup Auto Scaling dari grup penerapan:

> **Warning**
>
> Sub-langkah berikut akan meluncurkan instans EC2 baru tanpa perangkat lunak di dalamnya. Sebelum menjalankan perintah, pastikan bahwa `InService` instance Auto Scaling yang tidak menjalankan perangkat lunak dapat diterima. Misalnya, pastikan penyeimbang beban yang terkait dengan instance tidak mengirimkan lalu lintas ke host ini tanpa perangkat lunak.

a. Buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.

b. Pilih wilayah yang sesuai.

c. Pada panel navigasi, pilih Aplikasi.

d. Pilih nama CodeDeploy aplikasi Anda.

e. Pilih nama grup CodeDeploy penyebaran Anda.

f. Pilih Edit.

g. Dalam konfigurasi Lingkungan, batalkan pilihan grup Amazon EC2 Auto Scaling.

> **Note**
>
> Konsol tidak memungkinkan Anda untuk menyimpan konfigurasi jika tidak ada konfigurasi lingkungan yang ditentukan. Untuk melewati pemeriksaan, tambahkan sementara tag `EC2` atau `On-premises` yang Anda tahu tidak akan diselesaikan ke host mana pun. Untuk menambahkan tag, pilih instans atau On-premises instans Amazon EC2, dan tambahkan tag Kunci atau. **`EC2` `On-premises`** Anda dapat membiarkan tag Nilai kosong.

h. Pilih Simpan perubahan.

Setelah menyelesaikan sub-langkah ini, hal berikut terjadi:

i. CodeDeploy menghapus registrasi grup Auto Scaling dari grup penerapan.

ii. CodeDeploy menghapus hook siklus hidup Auto Scaling dari grup Auto Scaling.

iii. Karena hook yang menyebabkan penerapan gagal tidak lagi ada, Auto Scaling membatalkan instans EC2 yang ada dan segera meluncurkan yang baru untuk menskalakan ke kapasitas yang diinginkan. Contoh baru harus segera pindah ke `InService` negara bagian. Contoh baru tidak termasuk perangkat lunak.

5. Tunggu instans EC2 memasuki `InService` status. Untuk memverifikasi statusnya:

   a. Buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/
   b. Di panel navigasi, pilih Grup Auto Scaling.
   c. Pilih grup Auto Scaling Anda.
   d. Di panel konten, pilih tab Manajemen Instance.
   e. Di bawah Instance, pastikan bahwa kolom Lifecycle menunjukkan di InServicesebelah instance.

6. Re-register grup Auto Scaling dengan grup CodeDeploy penerapan menggunakan metode yang sama yang Anda gunakan untuk menghapusnya:

   a. Buka CodeDeploy konsol di https://console.aws.amazon.com/codedeploy/.
   b. Pilih wilayah yang sesuai.
   c. Pada panel navigasi, pilih Aplikasi.
   d. Pilih nama CodeDeploy aplikasi Anda.
   e. Pilih nama grup CodeDeploy penyebaran Anda.
   f. Pilih Edit.
   g. Dalam konfigurasi Lingkungan, pilih grup Amazon EC2 Auto Scaling dan pilih grup Auto Scaling Anda dari daftar.
   h. Di bawah instans atau On-premisesinstans Amazon EC2, temukan tag yang Anda tambahkan dan hapus.
   i. Hapus centang kotak di samping instans atau instans On-premises Amazon EC2.
   j. Pilih Simpan perubahan.

   Konfigurasi ini menginstal ulang hook siklus hidup ke grup Auto Scaling.

7. Buat penyebaran seluruh armada dengan Amazon S3 atau GitHub revisi yang Anda tahu sehat dan ingin digunakan.

   Misalnya, jika revisi adalah file.zip di bucket Amazon S3 yang `my-revision-bucket` dipanggil dengan kunci objek, lakukan hal `httpd_app.zip` berikut:

a. Di CodeDeploy konsol, di halaman Deployment Group, pilih Create deployment.

b. Untuk jenis Revisi, pilih Aplikasi saya disimpan di Amazon S3.

c. Untuk lokasi Revisi, pilih`s3://my-revision-bucket/httpd_app.zip`.

d. Untuk jenis file Revisi, pilih`.zip`.

e. Pilih Buat penerapan.

Karena sekarang ada satu `InService` instance dalam grup Auto Scaling, penerapan ini seharusnya berfungsi, dan Anda seharusnya tidak lagi melihat kesalahan Penerapan gagal karena tidak ada instance yang ditemukan untuk grup penerapan Anda.

8. Setelah penerapan berhasil, skala grup Auto Scaling Anda kembali ke kapasitas semula, jika sebelumnya Anda menskalakannya:

a. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/, dan pilih Grup Auto Scaling dari panel navigasi.

b. Pilih wilayah yang sesuai.

c. Buka grup Auto Scaling Anda.

d. Di Detail grup, pilih Edit.

e. Atur kapasitas yang diinginkan kembali ke nilai aslinya.

f. Pilih Perbarui.

Kode kesalahan untuk AWS CodeDeploy

Topik ini memberikan informasi referensi tentang CodeDeploy kesalahan.

Kode Kesalahan	Deskripsi
`AGENT_ISSUE`	Penerapan gagal karena masalah dengan CodeDeploy agen. Pastikan agen diinstal dan berjalan pada semua instance dalam grup penyebaran ini. Pelajari lebih lanjut: • Verifikasi CodeDeploy agen sedang berjalan

Kode Kesalahan	Deskripsi
	• Instal CodeDeploy agen • Bekerja dengan CodeDeploy agen
AUTO_SCALING_IAM_ROLE_PERMISSIONS	Peran layanan yang terkait dengan grup penyebaran Anda tidak memiliki izin yang diperlukan untuk melakukan operasi di AWS layanan berikut. Pelajari lebih lanjut: • Langkah 2: Buat peran layanan untuk CodeDeploy • Membuat Peran untuk Mendelegasikan Izin ke Layanan AWS
HEALTH_CONSTRAINTS	Penerapan keseluruhan gagal karena terlalu banyak instance individual gagal penerapan , terlalu sedikit instance sehat yang tersedia untuk penerapan, atau beberapa instance dalam grup penerapan Anda mengalami masalah. Pelajari lebih lanjut: • Instance Health • Memecahkan masalah contoh • Memecahkan masalah EC2/On-Premises penerapan

Kode Kesalahan	Deskripsi
HEALTH_CONSTRAINTS_INVALID	Penerapan tidak dapat dimulai karena jumlah minimum instance sehat, seperti yang ditentukan oleh konfigurasi penerapan Anda, tidak tersedia. Anda dapat mengurangi jumlah instans sehat yang diperlukan dengan memperbarui konfigurasi penerapan atau menambah jumlah instance dalam grup penerapan ini. Pelajari lebih lanjut: • Instance Health • Bekerja dengan instance untuk CodeDeploy
IAM_ROLE_MISSING	Penerapan gagal karena tidak ada peran layanan dengan nama peran layanan yang ditentukan untuk grup penerapan. Pastikan Anda menggunakan nama peran layanan yang benar. Pelajari lebih lanjut: • Langkah 2: Buat peran layanan untuk CodeDeploy • Ubah pengaturan grup penerapan dengan CodeDeploy

Kode Kesalahan	Deskripsi
`IAM_ROLE_PERMISSIONS`	CodeDeploy tidak memiliki izin yang diperlukan untuk mengambil peran, atau peran IAM yang Anda gunakan tidak memberi Anda izin untuk melakukan operasi di layanan. AWS Pelajari lebih lanjut: • Langkah 1: Menyiapkan • Langkah 2: Buat peran layanan untuk CodeDeploy • Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda

Kode Kesalahan	Deskripsi
NO_INSTANCES	Ini mungkin disebabkan oleh salah satu dari berikut ini. • Untuk EC2/On-Premises blue/green penerapan, jika Anda menggunakan tag Amazon EC2, tag tersebut mungkin tidak dikonfigurasi dengan benar. Di grup CodeDeploy penerapan Anda, pastikan mereka disertakan dalam instance biru dan instance hijau Anda. Anda dapat mengonfir masi instans Anda ditandai dengan benar menggunakan konsol Amazon EC2. • Jika Anda menggunakan grup Amazon EC2 Auto Scaling, grup Auto Scaling Anda mungkin tidak memiliki kapasitas yang cukup. Pastikan grup Auto Scaling Anda memiliki kapasitas yang cukup untuk penerapan Anda. Anda dapat melihat kapasitas grup Penskalaan Otomatis Amazon EC2 Anda dengan melihat jumlah instans sehatnya menggunakan konsol Amazon EC2. • Untuk EC2/On-Premises blue/green penyebaran, armada biru dan hijau mungkin tidak berukuran sama. Pastikan kedua armada memiliki ukuran yang sama. Pelajari lebih lanjut: • Tagging Instances for Deployments • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling • Buat aplikasi untuk blue/green penyebaran (konsol)

Kode Kesalahan	Deskripsi
`OVER_MAX_INSTANCES`	Penerapan gagal karena lebih banyak instance yang ditargetkan untuk penerapan daripada yang diizinkan untuk akun Anda. Untuk mengurangi jumlah instance yang ditargetk an untuk penerapan ini, perbarui setelan tag untuk grup penerapan ini atau hapus beberapa instance yang ditargetkan. Atau, Anda dapat menghubungi AWS Dukungan untuk meminta kenaikan batas. Pelajari lebih lanjut: • Ubah pengaturan grup penerapan dengan CodeDeploy • CodeDeploy kuota • Minta Peningkatan Batas
`THROTTLED`	Penerapan gagal karena lebih banyak permintaan dibuat daripada yang diizinkan AWS CodeDeploy oleh peran IAM. Coba kurangi jumlah permintaan. Pelajari lebih lanjut: • Tingkat Permintaan API Kueri
`UNABLE_TO_SEND_ASG`	Penerapan gagal karena grup penerapan tidak dikonfigurasi dengan benar dengan grup Amazon EC2 Auto Scaling. Di CodeDeploy konsol, hapus grup Amazon EC2 Auto Scaling dari grup penyebaran, lalu tambahkan lagi. Pelajari lebih lanjut: • Di Bawah Kap: CodeDeploy dan Integrasi Auto Scaling

Topik terkait

Pemecahan masalah CodeDeploy

CodeDeploy sumber daya

Sumber daya terkait berikut dapat membantu Anda saat Anda bekerja dengannya CodeDeploy.

Panduan referensi dan sumber daya pendukung

- AWS CodeDeploy Referensi API — Deskripsi, sintaks, dan contoh penggunaan tentang CodeDeploy tindakan dan tipe data, termasuk parameter umum dan kode kesalahan.
- CodeDeploy FAQ teknis - Pertanyaan teratas dari pelanggan tentang. CodeDeploy
- AWS pusat dukungan — Hub untuk membuat dan mengelola AWS Dukungan kasus Anda. Juga termasuk tautan ke sumber daya lain, seperti forum, FAQ teknis, status kesehatan layanan, dan. AWS Trusted Advisor
- AWS rencana dukungan — Halaman web utama untuk informasi tentang AWS Dukungan rencana.
- Hubungi Kami — Titik kontak pusat untuk pertanyaan tentang AWS penagihan, akun, acara, penyalahgunaan, dan masalah lainnya.
- AWS ketentuan situs — Informasi terperinci tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; dan topik lainnya.

Sampel

- CodeDeploy sampel pada GitHub - Sampel dan skenario template untuk CodeDeploy.
- CodeDeploy Jenkins Plugin - Jenkins plugin untuk. CodeDeploy
- CodeDeploy agen — Open-source versi CodeDeploy agen.

Blog

- AWS DevOps blog — Wawasan untuk pengembang, administrator sistem, dan arsitek.

AWS kit dan alat pengembangan perangkat lunak

AWS SDK dan alat berikut mendukung pengembangan solusi dengan CodeDeploy:

- AWS SDK untuk .NET

- AWS SDK untuk Java
- AWS SDK untuk JavaScript
- AWS SDK untuk PHP
- AWS SDK for Python (Boto)
- AWS SDK untuk Ruby
- AWS Toolkit for Eclipse — Bagian 1, 2, dan 3.
- AWS Tools for Windows PowerShell— Satu set PowerShell cmdlet Windows yang mengekspos fungsionalitas AWS SDK untuk .NET di lingkungan. PowerShell
- CodeDeploy cmdlet di Alat AWS untuk PowerShell - Satu set PowerShell cmdlet Windows yang mengekspos fungsionalitas di lingkungan. CodeDeploy PowerShell
- AWS Command Line Interface— Sintaks baris perintah seragam untuk mengakses layanan. AWS AWS CLI Menggunakan proses penyiapan tunggal untuk mengaktifkan akses untuk semua layanan yang didukung.
- AWS alat pengembang — Tautan ke alat pengembang dan sumber daya yang menyediakan dokumentasi, contoh kode, catatan rilis, dan informasi lainnya untuk membantu Anda membangun aplikasi inovatif dengan CodeDeploy dan AWS.

Riwayat dokumen

Tabel berikut menjelaskan perubahan besar yang dibuat pada panduan pengguna ini untuk mendukung fungsionalitas baru dan yang disempurnakan sejak rilis terakhir Panduan CodeDeploy Pengguna.

- Versi API: 2014-10-06

Perubahan	Deskripsi	Tanggal
CodeDeploy agent v2.1.2 rilis dengan dukungan macOS	AWS CodeDeploy Agen telah diperbarui ke versi 2.1.2, yang menambahkan dukungan untuk macOS pada instans EC2 Mac. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen dan Instal CodeDeploy agen untuk macOS.	Oktober 13, 2026
CodeDeploy agen v2.1.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 2.1.1. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	Oktober 2, 2026
CodeDeploy rilis agen v2.1.0	AWS CodeDeploy Agen telah diperbarui ke versi 2.1.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	September 7, 2026
CodeDeploy agen v2.0.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 2.0.1. Untuk informasi selengkapnya,	Agustus 24, 2026

	lihat Riwayat versi CodeDeploy agen.	
CodeDeploy rilis agen v2.0.0	AWS CodeDeploy Agen telah diperbarui ke versi 2.0.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	Juli 15, 2026
CodeDeploy agen v1.8.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.8.1. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	Februari 3, 2026
CodeDeploy agen v1.8.0 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.8.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	Juli 31, 2025
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	`AWSCodeDeployDeployerAccess` Itu diperbarui. Untuk informasi selengkapnya, lihat pembaruan kebijakan AWS terkelola.	Desember 16, 2024
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	`AWSCodeDeployReadOnlyAccess` Itu diperbarui. Untuk informasi selengkapnya, lihat pembaruan kebijakan AWS terkelola.	Desember 16, 2024
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	`AWSCodeDeployFullAccess` Itu diperbarui. Untuk informasi selengkapnya, lihat pembaruan kebijakan AWS terkelola.	Desember 16, 2024

CodeDeploy agen v1.7.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.7.1. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	November 14, 2024
Nama bucket Amazon S3 yang diperbarui	Memperbarui contoh bucket Amazon S3 dalam panduan ini untuk menggunakan nama yang dipesan oleh. AWS	Juni 17, 2024
Ditambahkan teks alternatif (alt text)	Memperbarui semua gambar dalam panduan ini untuk menyertakan teks alt. Teks Alt dibaca oleh pembaca layar dan membuat dokumentasi kami lebih mudah diakses oleh pengguna tunanetra.	22 Mei 2024
CodeDeploy agen v1.7.0 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.7.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	6 Maret 2024

Perintah yang diubah	`sudo service codedeploy-agent status|start|stop` Perintah tidak lagi direkomendasikan karena tidak digunakan `systemd` untuk manajemen proses CodeDeploy agen, yang merupakan praktik terbaik. Untuk memastikan `systemd` digunakan, gunakan `systemctl` perintah, seperti yang ditunjukkan pada contoh berikut:`sytemctl start codedeploy-agent` . Topik berikut diperbarui dengan `systemctl` perintah: Instal CodeDeploy agen untuk Amazon Linux atau RHEL, Instal CodeDeploy agen untuk Server Ubuntu, Pemecahan masalah semua peristiwa siklus hidup yang dilewati kesalahan, dan Buat file CodeDeploy log baru jika tidak sengaja dihapus.	Januari 12, 2024
Topik yang ditambahkan	Menambahkan Mengelola proses CodeDeploy agen dan Referensi file dalam topik skrip peristiwa siklus hidup Anda.	Januari 12, 2024
CodeDeploy sekarang mendukung konfigurasi zona	Memperbarui konfigurasi Create a deployment dengan CodeDeploy topik dengan informasi konfigurasi zonal.	Desember 7, 2023

CodeDeploy sekarang mendukung penerapan penghentian	Menambahkan penerapan Pengaktifan penghentian selama topik peristiwa penskalaan Auto Scaling untuk menjelaskan fitur penerapan penghentian. Juga memperbarui bagian AppSpec 'kait' untuk EC2/On-Premises penerapan, Buat grup penerapan untuk penerapan di tempat (konsol), dan Buat grup penerapan untuk topik EC2/On-Premises blue/green penerapan (konsol) untuk memperhitungkan fitur tersebut.	Desember 7, 2023
Pemformatan JSON tetap	Memperbaiki pemformatan sampel kode JSON di bagian AppSpec 'sumber daya' (hanya Amazon ECS dan AWS Lambda penerapan) topik.	Desember 3, 2023
Menambahkan topik pemecahan masalah	Menambahkan topik masalah penyebaran Amazon ECS Pemecahan Masalah.	24 Oktober 2023
Memperbarui nama AppSpec file	Memperbarui referensi CodeDeploy AppSpec file untuk menunjukkan bahwa AppSpec file harus diberi nama `appspec.yml` untuk EC2/On-Premises penerapan.	5 Oktober 2023

CodeDeploy sekarang mendukung beberapa penyeimbang beban	Memperbarui grup Buat penerapan untuk penyebaran (konsol) di tempat, Buat grup penerapan untuk penerapan (konsol), dan Siapkan penyeimbang beban di Elastic Load Balancing untuk topik EC2/On-Premises blue/green penerapan Amazon EC2 Amazon CodeDeploy EC2 untuk menunjukkan dukungan bagi beberapa penyeimbang beban.	26 September 2023
Memperbarui Wilayah dalam topik VPC	Memperbarui tabel dalam topik Gunakan CodeDeploy dengan Amazon Virtual Private Cloud untuk menampilkan dukungan Wilayah tambahan. Secara khusus, Asia Pasifik (Hyderabad), Asia Pasifik (Melbourne), Eropa (Milan), Eropa (Spanyol), dan Eropa (Zurich) diperbarui untuk menunjukkan dukungan untuk titik akhir agen.	September 22, 2023

Wilayah yang Diperbarui dalam topik kit sumber daya	Menambahkan AWS Wilayah berikut ke nama bucket kit Sumber Daya berdasarkan bagian Wilayah: Asia Pasifik (Osaka), Asia Pasifik (Hyderabad), Kanada (Tengah), Eropa (Spanyol), Eropa (Zurich), Timur Tengah (UEA). Juga memperbarui kebijakan IAM dengan Wilayah ini dan lainnya yang hilang.	September 22, 2023
Topik instalasi dan pembaruan agen yang dipersingkat	Memperpendek Instal CodeDeploy agen untuk Windows Server dan Perbarui CodeDeploy agen pada topik Windows Server. Menghapus URL bucket Amazon S3 yang berlebihan dan perintah salin Amazon S3.	21 September 2023
Menambahkan Wilayah Asia Pasifik (Jakarta)	Menambahkan Asia Pasifik (Jakarta) ke nama bucket Resource kit berdasarkan Region.	21 September 2023
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	Kebijakan AWSCodeDeployRole terkelola telah diperbarui. Untuk informasi selengkapnya, lihat AWS pembaruan kebijakan AWS terkelola.	16 Agustus 2023

Menambahkan batas	Menambahkan batas ke topik CodeDeploy batas. Batasnya adalah Jumlah maksimum alarm yang terkait dengan grup penyebaran.	15 Agustus 2023
Memperbaiki langkah yang terkait dengan penyeimbang beban	Memperbaiki instruksi di Buat grup penyebaran untuk EC2/On-Premises blue/green penyebaran (konsol). Langkah penyeimbang beban sekarang ditandai sebagai opsional.	3 Agustus 2023
Kata-kata yang diklarifikasi dalam topik Amazon ECS	Mengklarifikasi kata-kata dalam Tutorial: Menyebarkan aplikasi ke Amazon ECS. Kata-katanya sekarang menunjukkan Anda sedang menerapkan aplikasi. Sebelumnya, kata-kata tersebut menunjukkan bahwa Anda sedang menerapkan layanan Amazon ECS.	3 Agustus 2023
CodeDeploy sekarang tersedia di Israel (Tel Aviv) Region	CodeDeploy sekarang tersedia di Wilayah Israel (Tel Aviv) (il-central-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	31 Juli 2023

Pembaruan topik	Memperbarui topik masalah EC2/On-Premises penyebaran pemecahan masalah dengan tip tentang menggunakan runbook untuk melakukan tugas pemecahan masalah secara otomatis.	7 Juli 2023
Pembaruan topik	Memperbarui bagian AppSpec 'sumber daya' untuk topik penyebaran Amazon ECS dengan informasi selengkapnya tentang ARN definisi tugas.	7 Juli 2023
Pembaruan topik	Memperbarui Langkah 1: Instal dan konfigurasikan AWS CLI topik instans lokal dengan informasi pemecahan masalah.	7 Juli 2023
Pembaruan topik	Memperbarui topik pencegahan wakil yang Cross-service membingungkan dengan informasi tentang blue/green penyebaran Amazon ECS melalui. CloudFormation	6 Juli 2023
Pembaruan topik	Topik pencegahan wakil yang Cross-service membingungkan diperbarui dengan informasi tentang blue/green penyebaran Amazon ECS melalui. CloudFormation	6 Juli 2023

Pembaruan topik	Memperbarui konfigurasi penerapan yang telah ditentukan untuk topik platform EC2/On-Premises komputasi. Menambahkan catatan tentang perilaku konfigurasi penerapan yang `CodeDeployDefault.HalfAtATime` telah ditentukan sebelumnya dengan grup Auto Scaling.	29 Juni 2023
Pembaruan topik	Memperbarui keamanan Infrastruktur dalam AWS CodeDeploy topik untuk menunjukkan versi minimum dan direkomendasikan baru dari protokol Transport Layer Security (TLS).	28 Juni 2023
Batasi pembaruan	Batas berikut telah diubah: 'Jumlah jam maksimum penerapan EC2/On-Premises di tempat dapat dijalankan '. Untuk informasi selengkapnya, lihat Batas	Juni 27, 2023
Pembaruan topik	Langkah 3: Batasi topik izin CodeDeploy pengguna diperbarui dengan instruksi terperinci.	31 Mei 2023

CodeDeploy memperbarui kebijakan AWS terkelola yang ada	Kebijakan AWSCodeDeployFullAccess terkelola telah diperbarui. Untuk informasi selengkapnya, lihat AWS pembaruan kebijakan AWS terkelola.	16 Mei 2023
CodeDeploy agen v1.6.0 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.6.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	30 Maret 2023
CodeDeploy rilis agen v1.5.0	AWS CodeDeploy Agen telah diperbarui ke versi 1.5.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	3 Maret 2023
Pembaruan platform komputasi Amazon ECS	Penerapan pada platform komputasi Amazon ECS kini didukung di Wilayah Asia Pasifik (Jakarta).	Februari 8, 2023
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	Kebijakan AWSCodeDeployRole terkelola telah diperbarui. Untuk informasi selengkapnya, lihat AWS pembaruan kebijakan AWS terkelola.	3 Februari 2023
Pembaruan topik	Topik Penggunaan CodeDeploy dengan Amazon Virtual Private Cloud diperbarui dengan AWS Wilayah baru dan yang diubah.	2 Februari 2023

Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Melbourne) (ap-southeast-4). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeplo y agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	26 Januari 2023
Pembaruan praktik terbaik keamanan	CodeDeployBagian Memulai dengan dan beberapa bagian lainnya diperbarui agar sesuai dengan praktik terbaik AWS keamanan.	23 Januari 2023
CodeDeploy agen v1.4.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.4.1. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeplo y agen.	6 Desember 2022
Menambahkan topik pemecahan masalah	Menambahkan topik tentang cara memecahkan masalah kesalahan yang disebabkan oleh jalur file panjang yang digunakan dengan CodeDeplo y agen untuk Windows. Untuk informasi selengkapnya, lihat Jalur file yang panjang menyebabkan kesalahan “Tidak ada file atau direktori seperti itu”.	6 Desember 2022

Mengubah batas	Batas berikut telah diubah: 'Jumlah maksimum konfigurasi penerapan kustom yang terkait dengan AWS akun '. Batasnya sekarang 200. Untuk informasi selengkapnya tentang batas, lihat topik Batas.	7 September 2022
CodeDeploy agen v1.4.0 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.4.0. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	31 Agustus 2022
Memperbaiki beberapa batasan.	Batasan berikut telah diperbaiki: 'Jumlah maksimum penerapan bersamaan yang terkait dengan AWS akun 'sekarang 1000. 'Jumlah maksimum instance dalam satu penerapan 'sekarang 1000. 'Jumlah maksimum instance yang dapat digunakan oleh penerapan bersamaan yang sedang berlangsung dan terkait dengan satu akun 'sekarang 1000. 'Jumlah maksimum konfigurasi penerapan khusus yang terkait dengan AWS akun 'sekarang 100. Untuk informasi selengkapnya tentang batas, lihat topik Batas.	8 Agustus 2022

Menambahkan tabel yang menunjukkan CodeDeploy titik akhir yang didukung di setiap Wilayah.	Untuk informasi selengkapnya, lihat Menggunakan CodeDeploy dengan Amazon Virtual Private Cloud.	20 April 2022
Menambahkan batas baru untuk blue/green penerapan Amazon ECS.	Jumlah maksimum jam antara penyebaran revisi dan pergeseran lalu lintas ke lingkungan penggantian selama blue/green penyebaran Amazon ECS sekarang 120. Untuk informasi selengkapnya, lihat Penerapan dalam topik Batas.	12 April 2022
Menambahkan topik tentang cara mencegah masalah wakil yang bingung	Untuk informasi lebih lanjut, lihat AWS Identity and Access Management untuk AWS CodeDeploy.	Maret 14, 2022
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	`AmazonEC2RoleforAWSCodeDeployLimited` Peran itu diperbarui. Untuk informasi selengkapnya, lihat pembaruan kebijakan AWS terkelola.	22 November 2021
CodeDeploy memperbarui kebijakan AWS terkelola yang ada	`AWS CodeDeployRole` Itu diperbarui. Untuk informasi selengkapnya, lihat pembaruan kebijakan AWS terkelola.	18 Mei 2021

CodeDeploy agen v1.3.2 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.3.2. Untuk informasi selengkapnya, lihat Riwayat Versi CodeDeploy Agen.	6 Mei 2021
CodeDeploy mendukung pembaruan instans Amazon EC2 yang sudah ketinggalan zaman	CodeDeploy sekarang mendukung pembaruan instans Amazon EC2 yang sudah ketinggalan zaman secara otomatis. Untuk informasi selengkapnya, lihat Mengonfigurasi opsi lanjutan untuk grup penerapan.	23 Februari 2021
CodeDeploy agen v1.3.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.3.1. Untuk informasi selengkapnya, lihat Riwayat Versi CodeDeploy Agen.	22 Desember 2020
CodeDeploy rilis agen v1.3.0	AWS CodeDeploy Agen telah diperbarui ke versi 1.3.0. Untuk informasi selengkapnya, lihat Riwayat Versi CodeDeploy Agen.	10 November 2020
CodeDeploy agen v1.2.1 rilis	AWS CodeDeploy Agen telah diperbarui ke versi 1.2.1. Untuk informasi selengkapnya, lihat Riwayat Versi CodeDeploy Agen.	23 September 2020

CodeDeploy mendukung titik akhir Amazon VPC yang didukung oleh AWS PrivateLink	Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan VPC. CodeDeploy Anda dapat menggunakan koneksi ini untuk mengaktifkan CodeDeploy untuk berkomunikasi dengan sumber daya Anda di VPC Anda tanpa melalui internet publik. Untuk informasi selengkapnya, lihat Menggunakan CodeDeploy dengan Amazon Virtual Private Cloud.	11 Agustus 2020
Batas CodeDeploy layanan yang diperbarui	Memperbarui batas untuk jumlah aplikasi per akun dan jumlah grup penyebaran per aplikasi menjadi 1000. Untuk informasi selengkapnya tentang batas CodeDeploy layanan, lihat CodeDeploy batasan.	6 Agustus 2020
CodeDeploy rilis agen v1.1.2	AWS CodeDeploy Agen telah diperbarui ke versi 1.1.2. Untuk informasi selengkapnya, lihat Riwayat Versi CodeDeploy Agen.	4 Agustus 2020

CodeDeploy agent 1.1.0 rilis dan integrasi dengan Amazon EC2 Systems Manager	Versi 1.1.0 CodeDeploy agen sekarang tersedia, untuk informasi lebih lanjut, lihat Riwayat versi CodeDeploy agen. Sekarang Anda dapat menggunakan Amazon EC2 Systems Manager untuk CodeDeploy mengelola penginstalan dan pembaruan agen secara otomatis di Amazon EC2 atau instans lokal. Untuk informasi selengkapnya, lihat Menginstal CodeDeploy agen menggunakan Amazon EC2 Systems Manager.	30 Juni 2020
CodeDeploy mendukung pengelolaan blue/green penyebaran Amazon ECS dengan CloudFormation	Anda sekarang dapat menggunakan CloudFormation untuk mengelola blue/green penyebaran Amazon ECS melalui. CodeDeploy Anda menghasilkan penyebaran Anda dengan mendefinisikan sumber daya hijau dan biru Anda dan menentukan pengaturan perutean dan stabilisasi lalu lintas yang akan digunakan. CloudFormation Untuk informasi selengkapnya, lihat Membuat blue/green penyebaran Amazon ECS melalui. CloudFormation	19 Mei 2020

CodeDeploy mendukung pergeseran lalu lintas tertimbang untuk penyebaran Amazon ECS blue/green	CodeDeploy sekarang mendukung pergeseran lalu lintas tertimbang untuk penyebaran Amazon ECS blue/green. Anda memilih atau membuat konfigurasi penerapan untuk menentuka n jumlah interval perpindahan lalu lintas dalam penerapan dan persentase lalu lintas yang akan digeser di setiap interval. Topik berikut telah diperbarui untuk mencermin kan perubahan ini: Konfigura si penerapan pada platform komputasi Amazon ECS.	6 Februari 2020
Topik keamanan, otentikas i, dan kontrol akses yang diperbarui	Informasi keamanan, otentikas i, dan kontrol akses untuk CodeDeploy telah diatur ke dalam babak Keamanan baru. Untuk informasi selengkapnya, lihat Keamanan.	26 November 2019
CodeDeploy mendukung aturan pemberitahuan	Anda sekarang dapat menggunakan aturan notifikasi untuk memberi tahu pengguna tentang perubahan penting dalam penerapan. Untuk informasi selengkapnya, lihat Membuat aturan notifikasi.	5 November 2019

Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Hong Kong) (ap-east-1) Wilayah. Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini. Anda harus secara eksplisit mengaktifkan akses ke Wilayah ini. Untuk informasi selengkapnya, lihat Mengelola AWS Wilayah.	25 April 2019

Topik yang diperbarui	AWS CodeDeploy sekarang mendukung blue/green penerapan aplikasi kontainer dalam layanan Amazon ECS. CodeDeploy Aplikasi yang menggunakan platform komputasi Amazon ECS baru menyebarkan aplikasi kontainer ke tugas pengganti baru yang ditetapkan dalam layanan Amazon ECS yang sama. Beberapa topik telah ditambahkan dan diperbaru i untuk mencerminkan perubahan ini, termasuk Ikhtisar platform AWS CodeDeploy komputasi, Penerapan pada platform komputasi Amazon ECS, AppSpec struktur file untuk penerapan Amazon ECS, dan Membuat aplikasi untuk penyebaran layanan Amazon ECS (Konsol).	27 November 2018
CodeDeploy Agen yang diperbarui	AWS CodeDeploy Agen telah diperbarui ke versi 1.0.1.1597. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeplo y agen.	15 November 2018
Pembaruan konsol	Prosedur dalam panduan ini diperbarui agar sesuai dengan desain CodeDeploy konsol yang baru.	30 Oktober 2018

Versi minimum baru yang didukung dari CodeDeploy agen	Versi minimum yang didukung dari AWS CodeDeploy agen sekarang 1.8.x. Untuk informasi selengkapnya, lihat Riwayat versi CodeDeploy agen.	7 Agustus 2018

Pembaruan lebih awal

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna AWS CodeDeploy sebelum Juni 2018.

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Eropa (Paris) (eu-west-3) Wilayah. Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	19 Desember 2017
Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah Tiongkok (Ningxia). Untuk menggunakan layanan di Wilayah Tiongkok (Beijing) atau Wilayah Tiongkok (Ningxia), Anda harus memiliki akun dan kredensil untuk wilayah tersebut. Akun dan kredensil untuk AWS wilayah lain tidak berfungsi untuk Wilayah Beijing dan Ningxia, dan sebaliknya. Informasi tentang beberapa sumber daya untuk Wilayah Tiongkok, CodeDeploy seperti nama bucket Resource Kit CodeDeploy dan prosedur pemasangan agen, tidak termasuk dalam edisi Panduan Pengguna ini. CodeDeploy Untuk informasi selengkapnya:	11 Desember 2017

Perubahan	Deskripsi	Tanggal Diubah
	• CodeDeployMemulai dengan AWS di Wilayah Tiongkok (Beijing) • CodeDeploy Panduan Pengguna untuk Wilayah Tiongkok (versi bahasa Inggris | Versi bahasa Mandarin)	
Topik yang diperbarui	CodeDeploy sekarang mendukung penerapan fungsi Lambda. AWS Lambda Penerapan menggeser lalu lintas masuk dari fungsi Lambda yang ada ke versi fungsi Lambda yang diperbarui. Anda memilih atau membuat konfigurasi penerapan untuk menentukan jumlah interval perpindahan lalu lintas dalam penerapan dan persentase lalu lintas yang akan digeser di setiap interval. AWS Lambda penerapan didukung oleh Model Aplikasi AWS Tanpa Server (AWS SAM) sehingga Anda dapat menggunakan preferensi penyebaran AWS SAM untuk mengelola cara lalu lintas digeser selama penerapan. AWS Lambda Beberapa topik telah ditambahkan dan diperbaru i untuk mencerminkan perubahan ini, termasukIkhtisar CodeDeploy platform komputasi,Penerapan pada AWS Platform Komputasi Lambda,Buat sebuah AWS Penerapan Platform Komputasi Lambda (konsol),Buat aplikasi untuk AWS Lambda penyebaran fungsi (konsol), danMenambahk an AppSpec file untuk AWS Penyebaran Lambda.	28 November 2017
Topik baru	CodeDeploy sekarang mendukung penerapan langsung ke mesin lokal atau instance tempat CodeDeploy agen diinstal. Anda dapat menguji penerapan secara lokal dan, jika ada kesalahan, gunakan log kesalahan CodeDeplo y agen untuk men-debug mereka. Anda juga dapat menggunakan penyebaran lokal untuk menguji integritas revisi aplikasi, konten AppSpec file, dan banyak lagi. Untuk informasi selengkapnya, lihat Gunakan CodeDeploy agen untuk memvalidasi paket penerapan pada mesin lokal.	16 November 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	CodeDeploy dukungan untuk penyeimbang beban Elastic Load Balancing dalam kelompok penyebaran telah diperluas untuk menyertakan Network Load Balancer untuk penerapan dan penerapan di tempat. blue/green Sekarang Anda dapat memilih Application Load Balancer, Classic Load Balancer, atau Network Load Balancer untuk grup penyebaran Anda. Load balancer diperlukan untuk blue/green penerapan dan opsional untuk penerapan di tempat. Sejumlah topik telah diperbarui untuk mencerminkan dukungan ini, termasuk,Integrating CodeDeploy with Elastic Load Balancing, Buat aplikasi untuk penyebaran di tempat (konsol)Prasyarat penyebaran,Integrating CodeDeploy with Elastic Load Balancing, danBuat aplikasi untuk penyebaran di tempat (konsol).	12 September 2017
Topik yang diperbarui	CodeDeploy dukungan untuk penyeimbang beban Elastic Load Balancing dalam kelompok penyebaran telah diperluas untuk menyertakan Application Load Balancer untuk penerapan dan penerapan di tempat. blue/green Sekarang Anda dapat memilih antara Application Load Balancer dan Classic Load Balancer untuk grup penyebaran Anda. Load balancer diperlukan untuk blue/green penerapan, dan opsional untuk penerapan di tempat. Topik termasuk Integrating CodeDeploy with Elastic Load BalancingBuat aplikasi dengan CodeDeploy,, dan Buat grup penerapan dengan CodeDeploy telah diperbarui untuk mencerminkan dukungan tambahan ini.	10 Agustus 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik baru dan diperbarui	CodeDeploy sekarang mendukung penggunaan beberapa grup tag untuk mengidentifikasi serikat pekerja dan persimpangan instance untuk dimasukkan dalam grup penyebaran. Jika Anda menggunakan satu grup tag, instance apa pun yang diidentifikasi oleh setidaknya satu tag dalam grup akan disertakan dalam grup penyebara n. Jika Anda menggunakan beberapa grup tag, hanya instance yang diidentifikasi oleh setidaknya satu tag di setiap grup tag yang disertakan. Untuk informasi tentang metode baru untuk menambahkan instance ke grup penerapan, lihat. Tagging Instances for Deployments Topik lain yang diperbarui untuk mencerminkan dukungan ini termasuk Buat aplikasi untuk penyebaran di tempat (konsol)Buat aplikasi untuk blue/green penyebaran (konsol),Buat grup penerapan untuk penerapan di tempat (konsol),Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol),,Deploymen ts, dan Langkah 5: Buat grup aplikasi dan penyebaran dalamTutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub.	31 Juli 2017
Topik yang diperbarui	Dua metode tambahan untuk menginstal CodeDeploy agen pada instance Windows Server telah ditambahk an keInstal CodeDeploy agen untuk Windows Server. Selain PowerShell perintah Windows, instruksi sekarang tersedia untuk mengunduh file instalasi dengan menggunak an tautan HTTPS langsung dan dengan menggunakan perintah salin Amazon S3. Setelah file diunduh atau disalin ke sebuah instance, Anda dapat menjalankan instalasi secara manual.	12 Juli 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	CodeDeploy telah meningkatkan cara mengelola koneksi ke GitHub akun dan repositori. Anda sekarang dapat membuat dan menyimpan hingga 25 koneksi ke GitHub akun untuk mengaitkan CodeDeploy aplikasi dengan GitHub repositori. Setiap koneksi dapat mendukung beberapa repositori. Anda dapat membuat koneksi hingga 25 GitHub akun berbeda, atau membuat lebih dari satu koneksi ke satu akun. Setelah Anda menghubungkan aplikasi ke GitHub akun, CodeDeploy mengelola izin akses yang diperlukan tanpa memerlukan tindakan lebih lanjut dari Anda. Pembaruan untuk mencerminkan dukungan ini telah dilakukan untukTentukan informasi tentang revisi yang disimpan dalam repositori GitHub,Integrasi dengan CodeDeploy GitHub, danTutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub.	30 Mei 2017
Topik yang diperbarui	Di masa lalu, jika CodeDeploy agen mendeteksi file di lokasi target yang bukan bagian dari revisi aplikasi dari penerapan sukses terbaru, itu akan gagal penerapan saat ini secara default. CodeDeploy sekarang menyediakan opsi untuk bagaimana agen menangani file-file ini: gagal penyebaran, mempertahankan konten, atau menimpa konten. Buat penerapan dengan CodeDeploytelah diperbarui untuk mencerminkan dukungan ini, dan bagian baru Perilaku rollback dengan konten yang ada ditambahk an keMenerapkan ulang dan memutar kembali penerapan dengan CodeDeploy.	16 Mei 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	Classic Load Balancer di Elastic Load Balancing sekarang dapat ditetapkan ke grup penyebaran menggunakan konsol atau. CodeDeploy AWS CLI Selama penerapan di tempat, penyeimbang beban mencegah lalu lintas internet dirutekan ke instance saat sedang digunakan, dan kemudian membuat instance tersedia untuk lalu lintas lagi setelah penerapan ke instance itu selesai. Beberapa topik diperbarui untuk mencerminkan dukungan baru ini, termasukIntegrasi dengan yang lain AWS layanan,Integrating CodeDeploy with Elastic Load Balancing,Buat aplikasi untuk penyebaran di tempat (konsol),Buat grup penerapan untuk penerapan di tempat (konsol), danAppSpec Bagian 'kait'. Bagian baru telah ditambahkan ke panduan pemecahan masalah,. Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan	April 27, 2017
Topik yang diperbarui	Classic Load Balancer di Elastic Load Balancing sekarang dapat ditetapkan ke grup penyebaran menggunakan konsol atau. CodeDeploy AWS CLI Selama penerapan di tempat, penyeimbang beban mencegah lalu lintas internet dirutekan ke instance saat sedang digunakan, dan kemudian membuat instance tersedia untuk lalu lintas lagi setelah penerapan ke instance itu selesai. Beberapa topik diperbarui untuk mencerminkan dukungan baru ini, termasukIntegrasi dengan yang lain AWS layanan,Integrating CodeDeploy with Elastic Load Balancing,Buat aplikasi untuk penyebaran di tempat (konsol),Buat grup penerapan untuk penerapan di tempat (konsol), danAppSpec Bagian 'kait'. Bagian baru telah ditambahkan ke panduan pemecahan masalah,. Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan	1 Mei 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah Tiongkok (Beijing). Untuk menggunakan layanan di Wilayah Tiongkok (Beijing) atau Wilayah Tiongkok (Ningxia), Anda harus memiliki akun dan kredensil untuk wilayah tersebut. Akun dan kredensil untuk AWS wilayah lain tidak berfungsi untuk Wilayah Beijing dan Ningxia, dan sebaliknya. Informasi tentang beberapa sumber daya untuk Wilayah Tiongkok, CodeDeploy seperti nama bucket Resource Kit CodeDeploy dan prosedur pemasangan agen, tidak termasuk dalam edisi Panduan Pengguna ini. CodeDeploy Untuk informasi selengkapnya: • CodeDeployMemulai dengan AWS di Wilayah Tiongkok (Beijing) • CodeDeploy Panduan Pengguna untuk Wilayah Tiongkok (versi bahasa Inggris | Versi bahasa Mandarin)	Maret 29, 2017

Perubahan	Deskripsi	Tanggal Diubah
Topik baru dan diperbarui	Beberapa topik baru telah diperkenalkan untuk mencerminkan CodeDeploy dukungan baru untuk blue/green penerapan, metode penerapan di mana instance dalam grup penyebaran (lingkungan asli) digantikan oleh serangkaian instance yang berbeda (lingkungan pengganti). Ikhtisar blue/green penerapanmemberikan penjelasan tingkat tinggi tentang blue/green metodologi yang digunakan oleh. CodeDeploy Topik baru tambahan termasukBuat aplikasi untuk blue/green penyebaran (konsol). Membuat grup penyebaran untuk EC2/On-Premises blue/green penerapan (konsol), danSiapkan penyeimbang beban di Elastic Load Balancing untuk penerapan Amazon CodeDeploy EC2. Banyak topik telah diperbarui juga, termasuk,Buat penerapan dengan CodeDeploy, Bekerja dengan konfigurasi penerapan di CodeDeployBuat aplikasi dengan CodeDeploy,Bekerja dengan grup penyebaran di CodeDeploy,Bekerja dengan penerapan di CodeDeploy, danAppSpec Bagian 'kait'.	25 Januari 2017
Topik baru dan diperbarui	Topik baruGunakan perintah register-on-premises-instance (IAM Session ARN) untuk mendaftarkan instans lokal, menjelaskan cara mengautentikasi dan mendaftarkan instans lokal menggunakan kredensyal sementara yang diperbarui secara berkala yang dihasilkan. AWS Security Token Service Pendekatan ini memberikan dukungan yang lebih baik untuk mendukung armada besar instance lokal daripada hanya menggunakan kredensyal pengguna IAM statis pada setiap instance. Working with On-Premises Instancesjuga telah diperbarui untuk mencerminkan dukungan baru ini.	Desember 28, 2016

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah Eropa (London) (eu-west-2). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	13 Desember 2016
Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah Kanada (Tengah) (ca-central-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	8 Desember 2016
Topik yang diperbarui	CodeDeploy sekarang tersedia di Wilayah AS Timur (Ohio) (us-east-2). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	17 Oktober 2016
Topik baru	Bagian baru, Otentikasi dan Kontrol Akses, menyediakan informasi komprehensif tentang penggunaan AWS Identity and Access Management (IAM) dan CodeDeploy untuk membantu mengamankan akses ke sumber daya Anda melalui penggunaan kredensyal. Kredensyal ini memberikan izin yang diperlukan untuk mengakses AWS sumber daya, seperti mengambil revisi aplikasi dari bucket Amazon S3 dan membaca tag di instans Amazon EC2.	11 Oktober 2016
Topik yang diperbarui	Perbarui CodeDeploy agen di Windows Servertelah diperbarui untuk mencerminkan ketersediaan CodeDeploy agen baru updater untuk Windows Server. Ketika diinstal pada instance Windows Server, updater akan memeriksa secara berkala untuk versi baru. Ketika versi baru terdeteksi, updater akan menghapus versi agen saat ini, jika ada yang diinstal, sebelum menginstal versi terbaru.	Oktober 4, 2016

Perubahan	Deskripsi	Tanggal Diubah
Topik yang diperbarui	CodeDeploy sekarang terintegrasi dengan CloudWatch alarm Amazon, sehingga memungkinkan untuk menghenti kan penyebaran jika ada perubahan dalam keadaan alarm yang ditentukan untuk sejumlah periode berturut-turut, seperti yang ditentukan dalam ambang alarm. CodeDeploy juga sekarang mendukung secara otomatis memutar kembali penerapan jika kondisi tertentu terpenuhi, seperti kegagalan penerapan atau alarm yang diaktifkan. Sejumlah topik telah diperbarui untuk mencermin kan perubahan ini, termasuk,Buat aplikasi dengan CodeDeploy,Buat grup penerapan dengan CodeDeplo y,Ubah pengaturan grup penerapan dengan CodeDeplo y,Deployments,Menerapkan ulang dan memutar kembali penerapan dengan CodeDeploy, danIntegrasi produk dan layanan dengan CodeDeploy, bersama dengan topik baru,Memantau penyebaran dengan CloudWatch alarm di CodeDeploy.	15 September 2016
Topik baru dan diperbarui	CodeDeploy sekarang menyediakan integrasi dengan Amazon CloudWatch Events. Anda sekarang dapat menggunakan CloudWatch Peristiwa untuk memulai satu atau beberapa tindakan saat perubahan terdeteks i dalam status penerapan atau status instance yang termasuk dalam grup CodeDeploy penerapan. Anda dapat menggabungkan tindakan yang memanggil AWS Lambda fungsi; yang mempublikasikan ke aliran Kinesis atau topik Amazon SNS; yang mendorong pesan ke antrian Amazon SQS; atau yang, pada gilirannya, memicu tindakan alarm. CloudWatch Untuk informasi selengkap nya, lihat Memantau penerapan dengan Amazon Events CloudWatch.	9 September 2016

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	Topik Integrating CodeDeploy with Elastic Load Balancing dan Integrasi dengan yang lain AWS layanan telah diperbarui untuk mencerminkan opsi penyeimbangan beban tambahan. CodeDeploy sekarang mendukung Classic Load Balancer dan Application Load Balancer yang tersedia di Elastic Load Balancing.	11 Agustus, 2016
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Mumbai) (ap-south-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	27 Juni 2016
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Seoul) (ap-northeast-2). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini. Daftar isi telah direorganisasi untuk menyertakan bagian untuk instance, konfigurasi penerapan, aplikasi, grup penyebaran, revisi, dan penerapan. Bagian baru telah ditambahkan untuk CodeDeploy tutorial. Untuk kegunaan yang lebih baik, beberapa topik panjang, termasuk CodeDeploy AppSpec referensi file danPemecahan masalah CodeDeploy, telah dibagi menjadi topik yang lebih pendek. Informasi konfigurasi untuk CodeDeploy agen telah dipindahkan ke topik baru,CodeDeploy referensi konfigurasi agen.	Juni 15, 2016

Perubahan	Deskripsi	Tanggal Diubah
Topik baru dan diperbarui	Kode kesalahan untuk AWS CodeDeploymemberikan informasi tentang beberapa pesan kesalahan yang mungkin ditampilkan saat CodeDeploy penerapan gagal. Bagian berikut Pemecahan masalah CodeDeploy telah diperbarui untuk membantu menyelesaikan masalah penerapan dengan lebih baik: • Instans EC2 dalam grup Amazon EC2 Auto Scaling gagal diluncurkan dan menerima kesalahan “Heartbeat Timeout” • Hindari mengaitkan beberapa grup penerapan dengan satu grup Penskalaan Otomatis Amazon EC2	April 20, 2016
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Amerika Selatan (São Paulo) (sa-east-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini. Bekerja dengan CodeDeploy agentelah diperbarui untuk mencerminkan opsi konfigurasi new:max_revisions:, yang Anda gunakan untuk menentukan jumlah revisi aplikasi untuk grup penerapan yang ingin diarsipkan oleh agen. CodeDeploy	10 Maret 2016

Perubahan	Deskripsi	Tanggal Diubah
Topik baru dan diperbarui	CodeDeploy sekarang mendukung penambahan pemicu ke grup penerapan untuk menerima pemberitahuan tentang peristiwa yang terkait dengan penerapan atau instance dalam grup penerapan tersebut. Pemberitahuan ini dikirim ke penerima yang berlangganan topik Layanan Pemberitahuan Sederhana Amazon yang telah Anda jadikan bagian dari tindakan pemicu. Anda juga dapat menggunakan data JSON yang dibuat saat pemicu diaktifkan dalam alur kerja notifikasi khusus Anda sendiri. Untuk informasi selengkapnya, lihat Monitoring Deployments with Amazon SNS Event Notifications. Prosedur diperbarui untuk mencerminkan desain ulang halaman detail Aplikasi. Penerapan tidak gagal hingga satu jam saat instance dihentikan selama penerapanBagian di Pemecahan masalah CodeDeploy telah diperbarui. CodeDeploy kuotadiperbarui untuk mencerminkan batas yang direvisi untuk jumlah grup penerapan yang dapat dikaitkan dengan satu aplikasi; nilai yang diizinkan untuk pengaturan instance sehat minimum; dan versi yang diperlukan dari file. AWS SDK untuk Ruby	17 Februari 2016

Perubahan	Deskripsi	Tanggal Diubah
Topik baru dan diperbarui	CodeDeploy sekarang tersedia di Wilayah AS Barat (California Utara) (us-west-1). Beberapa topik, termasuk yang berisi instruksi untuk mendirikan CodeDeploy agen, diperbarui untuk mencerminkan penambahan Wilayah baru ini. Pilih jenis CodeDeploy repositoridaftar dan menjelaskan jenis repositori yang sekarang didukung oleh. CodeDeploy Topik baru ini akan diperbarui saat dukungan untuk jenis repositori lainnya diperkenalkan. Mengelola operasi CodeDeploy agendiperbarui dengan informasi tentang `.version` file baru yang ditambahkan ke instance untuk melaporkan versi CodeDeploy agen saat ini, serta informasi tentang versi agen yang didukung. Penyorotan sintaks untuk sampel kode, termasuk contoh JSON dan YAMAL, telah ditambahkan ke panduan pengguna. Tambahkan file spesifikasi aplikasi ke revisi untuk CodeDeploytelah direorganisasi sebagai petunjuk langkah demi langkah.	20 Januari 2016
Topik baru	Menyebarkan aplikasi di tempat yang berbeda AWS akunmenjelaskan persyaratan penyiapan dan proses untuk memulai penerapan yang dimiliki oleh akun organisasi Anda yang lain, tanpa memerlukan set lengkap kredensyal untuk akun lain tersebut. Ini paling berguna untuk organisasi yang menggunakan banyak akun untuk tujuan yang berbeda, seperti yang terkait dengan lingkungan pengembangan dan pengujian dan lainnya yang terkait dengan lingkungan produksi.	Desember 30, 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	Integrasi produk dan layanan dengan CodeDeployTopik telah didesain ulang. Sekarang termasuk bagian untuk contoh integrasi dari komunitas, dengan daftar posting blog dan contoh video yang terkait dengan CodeDeploy integrasi.	16 Desember 2015
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Singapura) (ap-southeast-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	9 Desember 2015
Pembaruan topik	Bekerja dengan CodeDeploy agentelah diperbarui untuk mencerminkan `:proxy_uri:` opsi baru dalam file konfigurasi CodeDeploy agen. CodeDeploy AppSpec referensi filediperbarui dengan informasi tentang penggunaan variabel lingkungan baru`DEPLOYMENT_GROUP_ID` , yang dapat diakses skrip hook selama peristiwa siklus hidup penerapan.	1 Desember 2015
Pembaruan topik	Langkah 2: Buat peran layanan untuk CodeDeploytelah diperbarui untuk mencerminkan prosedur baru untuk menciptakan peran layanan untuk CodeDeploy dan untuk menggabungkan perbaikan lainnya.	13 November 2015
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Eropa (Frankfurt) (eu-central-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini. Pemecahan masalah CodeDeployTopik diperbarui dengan informasi tentang memastikan bahwa pengaturan waktu pada instans akurat.	19 Oktober 2015

Perubahan	Deskripsi	Tanggal Diubah
Topik baru	CloudFormation template untuk CodeDeploy referensi diterbitkan untuk mencerminkan CloudFormation dukungan baru untuk CodeDeploy tindakan. Membuat Primary Components topik dan memperkenalkan definisi revisi target.	1 Oktober 2015
Pembaruan topik	Buat grup penerapan dengan CodeDeploytelah diperbaru i untuk mencerminkan kemampuan untuk menemukan instance untuk grup penerapan menggunakan pencarian wildcard. Instance Healthdiperbarui untuk memperjelas konsep contoh sehat minimum.	31 Agustus 2015
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Asia Pasifik (Tokyo) (ap-northeast-1). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	Agustus 19, 2015
Pembaruan topik	CodeDeploy sekarang mendukung penerapan ke instans lokal Red Hat Enterprise Linux (RHEL) dan instans Amazon EC2. Untuk informasi selengkapnya, lihat topik berikut: • Sistem operasi yang didukung oleh CodeDeploy agen • Bekerja dengan instance untuk CodeDeploy • Tutorial: Terapkan WordPress ke instans Amazon EC2 (Amazon Linux atau Red Hat Enterprise Linux dan Linux, macOS, atau Unix) • Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux)	23 Juni 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	CodeDeploy sekarang menyediakan satu set variabel lingkungan skrip penerapan Anda dapat digunakan selama penerapan. Variabel lingkungan ini mencakup informasi seperti nama CodeDeploy aplikasi saat ini, grup penyebaran, dan peristiwa siklus hidup penerapan, serta pengidentifikasi penerapan saat ini. CodeDeploy Untuk informasi lebih lanjut, lihat akhir AppSpec Bagian 'kait' bagian diCodeDeploy AppSpec referensi file.	29 Mei 2015
Pembaruan topik	CodeDeploy sekarang menyediakan serangkaian kebijakan AWS terkelola di IAM yang dapat Anda gunakan alih-alih membuat kebijakan setara secara manual sendiri. Ini termasuk: • Kebijakan untuk memungkinkan pengguna mendaftarkan revisi CodeDeploy hanya dengan lalu menerapkannya. CodeDeploy • Kebijakan untuk menyediakan pengguna dengan akses penuh ke CodeDeploy sumber daya. • Kebijakan untuk menyediakan pengguna dengan akses read-only ke CodeDeploy sumber daya. • Kebijakan untuk dilampirkan ke peran layanan sehingga CodeDeploy dapat mengidentifikasi instans Amazon EC2 dengan tag Amazon EC2, tag instans lokal, atau nama grup Amazon EC2 Auto Scaling dan menerapkan revisi aplikasi ke dalamnya. Untuk informasi selengkapnya, lihat Customer-managed contoh kebijakan bagian di Otentikasi dan Kontrol Akses.	29 Mei 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	CodeDeploy sekarang tersedia di Wilayah Eropa (Irlandia) (eu-west-1) dan Wilayah Asia Pasifik (Sydney) (ap-south east-2). Beberapa topik, termasuk yang berisi instruksi untuk menyiapkan CodeDeploy agen, diperbarui untuk mencerminkan ketersediaan Wilayah baru ini.	7 Mei 2015
Topik baru	CodeDeploy sekarang mendukung penerapan ke instans lokal dan instans Amazon EC2. Topik berikut ditambahkan untuk menggambarkan dukungan baru ini: • Working with On-Premises Instances • Tutorial: Menerapkan aplikasi ke instans lokal dengan CodeDeploy (Windows Server, Ubuntu Server, atau Red Hat Enterprise Linux) • Working with On-Premises Instances	2 April 2015
Topik baru	CodeDeploy sumber dayatelah ditambahkan.	2 April 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	Pemecahan masalah CodeDeploytelah diperbarui: • Long-running proses dapat menyebabkan penerapan gagalBagian baru menjelaskan langkah-langkah yang dapat Anda ambil untuk mengidentifikasi dan mengatasi kegagalan penerapan karena proses yang berjalan lama. • Pemecahan masalah Penskalaan Otomatis Amazon EC2 UmumBagian ini diperbarui untuk menunjukkan bahwa CodeDeploy telah meningkatkan logika batas waktu Amazon EC2 Auto Scaling CodeDeploy untuk agen dari lima menit menjadi satu jam. • Kait siklus hidup Amazon EC2 Auto Scaling yang tidak cocok dapat menyebabkan penerapan otomatis ke grup Amazon EC2 Auto Scaling berhenti atau gagalBagian baru menjelaskan langkah-langkah yang dapat Anda ambil untuk mengidentifikasi dan menangani penerapan otomatis yang gagal ke grup Penskalaan Otomatis Amazon EC2.	2 April 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	Topik berikut diperbarui untuk mencerminkan rekomenda si baru untuk membuat kebijakan kustom Anda sendiri dan kemudian melampirkannya ke pengguna dan peran di IAM: • Konfigurasikan instans Amazon EC2 agar berfungsi CodeDeploy • Langkah 4: Buat profil instans IAM untuk instans Amazon EC2 Anda • Langkah 2: Buat peran layanan untuk CodeDeploy Dua bagian ditambahkan kePemecahan masalah CodeDeploy: • Daftar periksa pemecahan masalah umum • PowerShell Skrip Windows gagal menggunakan versi Windows 64-bit PowerShell secara default AppSpec Bagian 'kait'Bagian dalam CodeDeploy AppSpec referensi file telah diperbarui untuk lebih akurat menggambarkan peristiwa siklus hidup penerapan yang tersedia.	Februari 12, 2015
Pembaruan topik	Bagian baru telah ditambahkan kePemecahan masalah CodeDeploy:Instans EC2 dalam grup Amazon EC2 Auto Scaling gagal diluncurkan dan menerima kesalahan “Heartbeat Timeout”. Sebuah CloudBees bagian ditambahkan keIntegrasi produk dan layanan dengan CodeDeploy.	28 Januari 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	Bagian berikut ditambahkan kePemecahan masalah CodeDeploy: • Menggunakan beberapa editor teks untuk membuat AppSpec file dan skrip shell dapat menyebabkan penerapan gagal • Menggunakan Finder di macOS untuk menggabungkan revisi aplikasi dapat menyebabkan penerapan gagal • Memecahkan masalah peristiwa siklus hidup yang gagal ApplicationStop BeforeBlockTraffic, atau AfterBlockTraffic penerapan • Memecahkan masalah peristiwa siklus hidup DownloadBundle penerapan yang gagal dengan UnknownError: tidak dibuka untuk dibaca • Pemecahan masalah Penskalaan Otomatis Amazon EC2 Umum	Januari 20, 2015
Topik baru	Integrasi produk dan layanan dengan CodeDeployBagian ini diperbarui untuk memasukkan topik-topik berikut: • Mengintegrasikan CodeDeploy dengan Amazon EC2 Auto Scaling • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi ke grup Auto Scaling • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Integrasi dengan CodeDeploy GitHub • Tutorial: Gunakan CodeDeploy untuk menyebarkan aplikasi dari GitHub	Januari 9, 2015

Perubahan	Deskripsi	Tanggal Diubah
Pembaruan topik	• Secara otomatis menyebarkan dari CodePipeline dengan CodeDeployBagian tersebut ditambahkan keIntegrasi dengan CodeDeploy GitHub. Anda sekarang dapat secara otomatis memicu penyebaran dari GitHub repositori setiap kali kode sumber di repositori itu diubah. • Memecahkan masalah Amazon EC2 Auto ScalingBagian tersebut ditambahkan kePemecahan masalah CodeDeploy. Bagian baru ini menjelaskan cara memecahkan masalah umum dengan penerapan ke grup Amazon EC2 Auto Scaling. • Subbagian baru “file Contoh” ditambahkan ke AppSpec Bagian 'file' (EC2/On-Premises hanya penerapan) bagian. CodeDeploy AppSpec referensi file Subbagian baru ini mencakup beberapa contoh cara menggunak an `files` bagian AppSpec file untuk menginstruksikan CodeDeploy menyalin file atau folder tertentu ke lokasi tertentu pada instans Amazon EC2 selama penerapan.	8 Januari 2015
Topik baru	Monitoring Deploymentstelah ditambahkan. CodeDeplo y terintegrasi dengan AWS CloudTrail, layanan yang menangkap panggilan API yang dilakukan oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. CodeDeploy	17 Desember 2014
Rilis publik awal	Ini adalah rilis publik awal dari Panduan CodeDeploy Pengguna.	12 November 2014

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.